INTEGRATING MULTI-LEVEL TRAFFIC SIMULATION FOR SAEV DRIVING PROFILE MODELING IN ADVANCED AUTOMOTIVE ENGINEERING

Zur Erlangung des akademischen Grades einer

DOKTORIN DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für

Bauingenieur-, Geo- und Umweltwissenschaften

des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Henriette Triebke, M.Sc.

Tag der mündlichen Prüfung: 06.02.2025

Referent: Prof. Dr.-Ing. Peter Vortisch
Korreferent: Prof. Dr. rer. nat. Peter Wagner

Karlsruhe (2025)

Integrating Multi-Level Traffic Simulation for SAEV Driving Profile Modeling in Advanced Automotive Engineering

A

DISSERTATION

by

M.Sc. Henriette Triebke

in partial fulfillment of the requirements for obtaining the degree of

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

submitted to

the Department of Civil Engineering, Geo and Environmental Science
of the Karlsruhe Insitute of Technology (KIT)

Day of oral examination: 2025-02-06

First referee: Prof. Dr.-Ing. Peter Vortisch
Second referee: Prof. Dr. rer. nat. Peter Wagner

Summary

The need to tackle climate change and reduce emissions, along with rapid urbanization resulting in increasingly congested cities, highlights the urgency for sustainable mobility solutions to make urban transport cleaner, more efficient, and accessible. Shared, autonomous electric vehicle (SAEV) fleets, when used to complement public transport, have the potential to alleviate urban traffic and reduce emissions. However, the adoption of such innovative mobility solutions raises critical technical questions for vehicle design, particularly from the perspective of automotive suppliers.

SAEVs are expected to have significantly different driving patterns and load profiles compared to privately owned, human-driven cars, necessitating a reevaluation of vehicle components such as the powertrain and thermal management system, which are highly sensitive to operating conditions. Consequently, accurate SAEV driving cycles are essential for automotive requirement engineering. In addition to capturing key driving dynamics and vehicle states (e.g., idling, charging, and occupied), these driving cycles must also consider the effects of different pricing schemes on travel behavior, electric vehicle (EV) range and charging constraints, as well as diverse fleet routing and dispatching strategies.

Given the limitations of real-world data for these novel concepts, we utilize multi-level traffic simulation to generate SAEV driving profiles tailored to automotive engineering needs. By integrating traffic modeling frameworks of varying granularity, we achieve a balance between computational efficiency and modeling precision – representing large-scale network flows with simplified models while applying more detailed models to specific areas. Our goal is to develop an automated toolchain and demonstrate its large-scale applicability, effectively addressing the persistent challenges associated with multi-level simulation in transport research.

The central contribution of our work is the successful integration of the mesoscopic transport planning framework MATSim with the microscopic

traffic simulation tool SUMO, overcoming substantial challenges arising from their fundamental differences in network representation and traffic dynamics. By leveraging specific characteristics of the vehicle-centric use case, we develop a novel sequential tool-coupling approach that enables a seamless transition between the two simulation frameworks. The key innovations include:

- Automated methods for building and populating microscopic SUMO models from mesoscopic MATSim simulations. This includes a robust graph-based network matching routine to translate mesoscopic network elements into their microscopic counterparts and a novel technique to disaggregate MATSim's sampled travel demand into the higher-resolution microscopic domain across dynamic meso-micro network borders. We apply stochastic methods to reduce computational load in SUMO while maintaining the representativeness of a vehicle fleet.
- An innovative, automotive-inspired calibration approach that employs a genetic algorithm to align SUMO's traffic states with those observed in a calibrated MATSim simulation, prioritizing the consistency of travel times over traffic counts as in traditional approaches. This distinctive choice of objective is grounded in our findings from numerous analytical and numerical experiments, which revealed significant discrepancies in traffic dynamics and network capacity between MATSim and SUMO. As a result, we focused on calibrating traffic performance by adjusting various demand, routing, and bottleneck mitigating measures.

The resulting toolchain is robust and efficient, even in scenarios with imperfect microscopic networks. Unlike many large-scale microscopic traffic simulations that require extensive network cleaning and demand synthesis, our method reduces these challenges by subjecting the microscopic modeling effort to the less stringent requirements of mesoscopic models. Additionally, we introduce a complementary data-driven approach to derive SAEV drive cycles. This method refines mesoscopic speed profiles with real-world driving data, providing a pragmatic (though less sensitive) alternative for obtaining detailed driving dynamics without requiring microscopic simulations. We demonstrate the toolchain's feasibility in a case study on power engine design, highlighting the distinct requirements of SAEV fleets compared to privately owned cars.

Overall, we offer both practical and advanced methodologies for generating SAEV driving profiles, advancing research on multi-level traffic simulation

and creating new opportunities for automotive suppliers to adapt to changing mobility landscapes. Our work not only supports the automotive industry in designing future vehicle systems but also paves the way for exploring new business opportunities in SAEV environments.

Zusammenfassung

Die Notwendigkeit, dem Klimawandel entgegenzuwirken und Emissionen zu reduzieren, sowie die zunehmende Urbanisierung, die zu immer stärker belasteten Städten führt, unterstreichen die Dringlichkeit nachhaltiger Mobilitätslösungen die helfen den städtischen Verkehr sauberer, effizienter und zugänglicher zu gestalten. Flotten aus gemeinsam genutzten, autonomen Elektrofahrzeugen (SAEVs), die als Ergänzung zum öffentlichen Verkehr eingesetzt werden, wird das Potenzial zugeschrieben, den städtischen Verkehr zu entlasten und Emissionen zu senken. Die Einführung solcher innovativer Mobilitätslösungen bringt jedoch entscheidende technische Herausforderungen für die Fahrzeugentwicklung mit sich, insbesondere aus der Sicht von Automobilzulieferern.

Es wird erwartet, dass SAEVs im Vergleich zu privat genutzten, menschlich gesteuerten Fahrzeugen signifikant abweichende Fahrmuster und Lastprofile aufweisen. Daher ist eine Neubewertung von Fahrzeugkomponenten wie dem Antriebsstrang und dem Thermomanagementsystem notwendig, da deren Auslegung stark von den Betriebsbedingungen abhängig ist. Aus diesem Grund wird die Entwicklung präziser SAEV-Fahrzyklen als essenziell für das Anforderungsmanagement in der Fahrzeugentwicklung angesehen. Solche Fahrzyklen müssen nicht nur wesentliche Fahrdynamiken und Fahrzeugzustände (wie Stillstand, Laden und Fahrgastbesetzung) erfassen, sondern auch die Auswirkungen unterschiedlicher Preismodelle auf das Mobilitätsverhalten, die Reichweite von Elektrofahrzeugen und Ladebeschränkungen sowie diverse Flottenstrategien im Bereich des Routings und der Einsatzplanung berücksichtigen.

Da reale Daten für diese neuartigen Konzepte nur eingeschränkt verfügbar sind, wird eine mehrstufige Verkehrssimulation eingesetzt, um SAEV-Fahrprofile zu generieren, die den Bedürfnissen der Automobilentwicklung entsprechen. Durch die Integration von Verkehrsmodellierungsansätzen unterschiedlicher Granularität wird eine Balance zwischen Rechenaufwand und Modellgenauigkeit erreicht – großflächige Netzwerke werden durch vereinfachte Modelle abgebildet, während detailliertere Modelle in spezifischen Bereichen

Anwendung finden. Ziel der vorliegenden Arbeit ist es dabei eine automatisierte Toolchain zu entwickeln, die die Herausforderungen mehrstufiger Verkehrssimulationen effektiv bewältigt und somit für großflächige Anwendungen geeignet ist.

Der zentrale Beitrag der Arbeit liegt in der erfolgreichen Integration des mesoskopischen Verkehrsplanungswerkzeugs MATSim mit dem mikroskopischen Verkehrssimulationswerkzeug SUMO, wobei erhebliche Herausforderungen aufgrund grundlegender Unterschiede in der Netzdarstellung und Verkehrsdynamik überwunden wurden. Durch die Berücksichtigung spezifischer Merkmale des fahrzeugzentrierten Anwendungsfalls wurde ein neuartiger, sequenzieller Kopplungsansatz entwickelt, der einen nahtlosen Übergang zwischen den beiden Simulationsframeworks ermöglicht. Die wesentlichen Innovationen bestehen aus:

- Automatisierten Methoden zur Erstellung mikroskopischer SUMO-Modelle auf Basis von mesoskopischen MATSim-Simulationen. Dies umfasst eine robuste, graphenbasierte Netzwerk-Matching-Routine zur Übersetzung mesoskopischer Netzwerkelemente in mikroskopische Gegenstücke sowie eine neuartige Technik zur Disaggregation der in MATSim gesampelten Verkehrsnachfrage in das höher aufgelöste mikroskopische Modell über dynamische Meso-Mikro-Netzwerkgrenzen hinweg. Stochastische Methoden werden eingesetzt, um den Rechenaufwand in SUMO zu reduzieren und gleichzeitig die Repräsentativität der Fahrzeugflotte zu gewährleisten.
- Einem innovativen, von der Fragestellunge der Automobilindustrie inspirierten Kalibrierungsansatz, der einen genetischen Algorithmus nutzt, um die Verkehrszustände von SUMO mit denen einer kalibrierten MATSim-Simulation abzugleichen. Dabei wird der Schwerpunkt auf die Konsistenz von Reisezeiten anstelle von Verkehrsaufkommen gelegt, wie es in herkömmlichen Ansätzen üblich ist. Diese Wahl des Kalibrierungsziels beruht auf Erkenntnissen aus zahlreichen analytischen und numerischen Experimenten, die signifikante Diskrepanzen in der Verkehrsdynamik und Netzkapazität zwischen MATSim und SUMO aufzeigten. Daher wird der Fokus auf die Kalibrierung der Verkehrsleistung durch Anpassung verschiedener Nachfrage-, Routing- und Engpassbewältigungsmaßnahmen gelegt.

Die resultierende Toolchain ist robust und effizient, selbst in Szenarien mit unvollständigen mikroskopischen Netzwerken. Im Gegensatz zu vielen großflächigen mikroskopischen Verkehrssimulationen, die umfangreiche Netzbereinigungen und Nachfragesynthesen erfordern, umgeht unser Ansatz diese Herausforderungen indem er für die mikroskopische Simulation die weniger strengen Anforderungen mesoskopischer Modelle zu Grunde legt. Darüber hinaus wird ein ergänzender, datengetriebener Ansatz zur Ableitung von SAEV-Fahrzyklen vorgestellt. Dieser Ansatz verfeinert mesoskopische Geschwindigkeitsprofile mit realen Fahrdaten und bietet eine pragmatische Alternative zur Gewinnung detaillierter Fahrdynamiken ohne den Einsatz mikroskopischer Simulationen. Die Umsetzbarkeit der Toolchain wird in einer Fallstudie zur Antriebsstrangauslegung demonstriert, wobei die spezifischen Anforderungen von SAEV-Flotten im Vergleich zu privat genutzten Fahrzeugen aufgezeigt werden.

Insgesamt werden praxisorientierte und fortschrittliche Methoden zur Generierung von SAEV-Fahrprofilen entwickelt, die die Forschung im Bereich mehrstufiger Verkehrssimulationen vorantreiben und neue Möglichkeiten für Automobilzulieferer schaffen, sich an die sich verändernden Mobilitätslandschaften anzupassen. Die Arbeit unterstützt die Automobilindustrie nicht nur bei der Entwicklung zukünftiger Fahrzeugsysteme, sondern eröffnet auch die Möglichkeit neue Geschäftsmodelle im SAEV-Umfeld zu explorieren.

Preface

This dissertation was written during my time as a research associate at the Corporate Research division of Robert Bosch GmbH.

Putting this thesis together has been a tough journey. Taking the abstract concept of new travel modes and distilling it into precise requirements for vehicle component design has been quite a stretch. It required moving from the meta-level of new mobility concepts to the mesoscopic realm of transport planning, the microscopic level of traffic simulation, and finally, to nanoscopic vehicle simulation. All of this to explore how novel mobility visions could be converted into technically and economically viable products for automotive suppliers.

I admit there were moments when I wished for a more focused research area that would allow me to dive deeper into specific issues. Instead, I often felt caught between the meta-meso-micro-nano spectrum, with little time for in-depth exploration. I had to constantly balance the corporate goals of my industry-sponsored thesis with the academic rigor required for a doctoral dissertation. To illustrate my struggles, I chose a chronological narrative for this thesis, including both the successes and setbacks of my research journey. I acknowledge, however, that this approach may have come at the expense of a more linear and comprehensive guiding thread. My apologies to the referees for any inconvenience caused.

Now, as I look back on nearly six years of work, I feel immense joy and satisfaction with how the thesis has finally come together. Most of all, I'm proud I persevered, despite the birth of my daughter Rebekka three years ago and the challenges of balancing motherhood with an academic career.

This work would not have been possible without the collective support of the incredible people around me. In the academic realm, I'd like to thank IFV and Prof. Dr.-Ing. Peter Vortisch for supervising my doctorate and for giving me the opportunity to pursue a PhD in traffic engineering as an electrical engineer.

I hope my unorthodox approaches did you justice. I'd also like to thank the DLR and SUMO community for their invaluable feedback during the 2020 and 2021 SUMO user conferences, as well as my anonymous TRB reviewers for relentlessly challenging my findings and pulling me out of my tunnel vision. A big thanks to TU Berlin and the MATSim community for the collaboration during the PAVE project, and especially to Amit Agarwal for his patience in introducing us to MATSim.

At Bosch, I am deeply grateful to Anna, Wael, Fary, Markus, Lakshay, Frank, Ronja, Leon and the rest of the "Bosch gang" for the brightest lunch breaks I've ever had – until Covid sadly separated us. I truly cherished our time together, and I especially enjoyed working with you all. A special thanks goes to Norbert, Mike and Hanse, for giving me the opportunity to pursue my PhD at Bosch, and to Mike, in particular, for your unwavering support throughout this journey. You were the best group leader I could have asked for.

I also want to thank my former teachers Herr Meyer and Frau Meyer-Oehring for nurturing my curiosity and instilling in me a passion for the study of languages, mathematics and technical sciences. Additionally, Prof. Dr.-Ing. Heide Brandstädter (HTW Berlin) and Dr. Sabine Wagner (formerly of Fraunhofer IAO) served as inspiring female role models in the technical field, demonstrating that success in these realms is possible even with young children.

Most importantly, I want to express my deepest gratitude to Markus. We started as colleagues at Bosch, with me admiring your kind, upright, and brilliant astrophysicist mind, and being grateful for your academic guidance, which encouraged me to venture into areas far beyond my comfort zone. Markus, who became my dearest friend during the tumultuous times of Covid. Markus, whom I married in 2021, just a few months before our beautiful daughter was born. You have been my steady rock, supporting me through the hardest times and never giving up on me. I love you. A heartfelt thank you to your parents, Ingeborg and Gerhard, who have generously spent nearly every weekend with Becky over the past year so I could devote extra hours to finishing this work. This PhD is as much a result of your support as mine.

Table of contents

Acronyms and Glossary

Acronyms

AC	alternating current
ACC	adaptive cruise control
AD	autonomous driving
ADR	Automotive Disruption Radar (issued by Roland Berger)
AI	artificial intelligence
ASI	Avoid-Shift-Improve
AV	autonomous vehicle
BEV	battery electric vehicle
CaaS	Car-as-a-Service
CACC	cooperative adaptive cruise control
CAV	connected autonomous vehicle
CCS	Combined Charging System (standard for EV charging)
C2X	car-to-everything (communication)
DC	direct current
DRT	demand-responsive transport
DVRP	dynamic vehicle routing problem
EBIT	earnings before interest and taxes
ED	electric drive
EMA	electric machine

EV	electric vehicle
FEM	finite element method
FIFO	first-in first-out (principle)
FKFS	Research Institute for Automotive Engineering and Vehicle Engines Stuttgart
GA	genetic algorithm
GK	Gauss-Krüger projection
GHG	greenhouse gas
HD	human driving
ICE	internal combustion engine
ID	identifier
IDE	Integrated Development Environment
INV	inverter
KL	Kullback-Leibler (divergence)
KPI	key performance indicator
KS	Kolmogorov-Smirnov (divergence)
KSD	kernel-smoothed distribution
LWR	Lighthill-Whitham-Richards (model)
MaaS	Mobility-as-a-Service
MSE	mean square error
NACS	North American Charging Standard
NEDC	New European Drive Cycle
OEM	original equipment manufacturer
OP	operating points (of an electric machine)
OSM	OpenStreetMap
PC	privately owned passenger cars

PM	permanent magnet
PMSM	permanent magnet synchronous motor
PT	public transport
QoS	quality of service
SAE	Society of Automotive Engineers
SAEV	shared autonomous electric vehicle
SAV	shared autonomous vehicle
SOC	state of charge
SQ	spatial queue (model)
Tra	transmission
TraCI	Traffic Control Interface
UTM	Universal Transverse Mercator projection
VRP	vehicle routing problem
V2I	vehicle-to-infrasstructure
V2V	vehicle-to-vehicle
V2X	vehicle-to-everything
WGS84	World Geodetic System 1984
WLTP	Worldwide harmonized Light vehicles Test Procedure
WLTC	Worldwide harmonized Light vehicles Test Cycle
XML	Extensible Markup Language
0D	zero-dimensional (model)
1D	one-dimensional (model)

Glossary of key MATSim and SUMO concepts

DUAROUTER A routing tool in SUMO that generates optimal vehicle paths based on current traffic conditions, network data, and dynamic factors, enabling realistic vehicle routing in both static and dynamic simulation scenarios.

flow capacity MATSim parameter defined for each network link, representing the maximum allowable vehicle outflow rate per hour.

flow capacity factor MATSim parameter that scales the `flow capacity` to maintain accurate traffic dynamics when using sampled populations. For example, in a 10% sample run, the `flow capacity factor` is typically set to 0.1.

ignore-junction-blocker SUMO option that teleports vehicles unable to move for an extended period to resolve deadlocks. If this is not desired, the option `-ignore-junction-blocker <TIME>` can be used to ignore vehicles blocking an intersection on an intersecting lane after the specified time. This models real-life behavior where drivers find ways around vehicles blocking the intersection.

max-depart-delay SUMO parameter specifying the time interval during which a vehicle ready for insertion must reattempt to enter a fully occupied edge before being permanently excluded from the simulation.

mobsim The first step of the MATSim cycle (see Figure 2.1), during which the activity chains of all agents are executed simultaneously within a traffic simulation.

NETCONVERT A tool in SUMO that converts various network formats into SUMO's network format, allowing users to create and modify road networks for simulation purposes. It facilitates the import of data from other traffic simulation tools and supports customization of network attributes.

remove-loops SUMO DUAROUTER option that suppresses U-turns at vehicles' starting and ending edges, aiming to reduce network disturbances.

rerouting-period SUMO option that sets the time intervals for route reevaluation.

rerouting-probability SUMO option specifying the probability of a vehicle re-routing during simulation.

sample run MATSim functionality that enables the simulation of a subset of agents. For example, a 10% sample run means that only 10% of all agents are simulated mesoscopically. To preserve traffic dynamics, `flow capacity` and `storage capacity` are adjusted accordingly.

short-link artifacts MATSim artifacts that describe the phenomenon where vehicle delays primarily occur on short links. In such cases, short links act as vehicle sinks, where queues fail to propagate correctly upstream, hindering MATSim's ability to model meaningful spatio-temporal congestion patterns. These artifacts result from MATSim's sample runs and are further aggravated by the `stuckTime` parameter, which often violates the link's `storage capacity` constraint.

storage capacity MATSim parameter, implicitly defined for each network link based on the link and `vehicle lengths`, representing the maximum allowable number of vehicles that can be accommodated on that link.

storage capacity factor MATSim parameter that scales the `storage capacity` to maintain accurate traffic dynamics when using a sampled population. For example, in a 10% sample run, the `storage capacity factor` is typically set to 0.1.

stuckTime MATSim parameter designed to prevent gridlocks by ensuring minimal vehicle flow, even under highly congested conditions. It forces the first vehicle in the queue to move to the next link (regardless of the `storage capacity` constraint) if it remains stuck for too long, typically after 30 s.

vehicle length In MATSim, the nominal vehicle length is set to 7.5 meters, applicable universally to all vehicles unless specified otherwise. During `sample runs`, the effective vehicle length is scaled based on the `storage capacity factor`. For example, in a simulation with a 10% sampling rate, the effective length of each vehicle is increased to 75 meters, reflecting the impact of capacity adjustments on vehicle representation in the simulation model.

time-to-teleport SUMO option defining the maximum time a leading vehicle can be stuck at an intersection before being teleported to the next available edge along its route, where it moves virtually at average edge speed during teleportation.

VSPAdjustments MATSim parameter set applied to improve the realism of travel times by accounting for additional delays caused by factors such as traffic lights, lane changes, and right-of-way rules that are not explicitly modeled in MATSim. When enabled, `flow capacities` are reduced by 33% for primary roads and 20% for secondary roads. However, for links shorter than 100 meters, the capacities are doubled to offset this reduction.

1 Introduction

The automotive industry is in great turmoil, facing considerable challenges at multiple fronts. Pressured by ecological concerns, a growing number of national authorities are ratifying the end of internal combustion engine (ICE) vehicles, thereby disrupting major revenue streams for original equipment manufacturers (OEMs) and suppliers. The rise of electric vehicles (EVs), the transition to software-defined automobiles, and the rapidly evolving field of autonomous driving (AD) are attracting new market entrants, threatening to outpace established players. The traditional automotive industry struggles to shed dependence on its ICE heritage, reluctantly relinquishing its former dominance in mechanical and electrical engineering to embrace software and AI (artificial intelligence)-driven solutions. Additionally, growing interest in shared mobility and changing consumer preferences, particularly in markets across Southeast Asia, pose further challenges to the automotive sector. Faced with strong competition from Chinese OEMs and Tesla, both emerging as global EV market leaders, established players find themselves in a state of shock, risking irrelevance. However, despite recent scandals and valid criticisms, the automotive sector remains crucial for economic prosperity in their respective home countries. Thus, the potential failure of traditional OEMs to adapt to recent socio-technological transformations poses a significant economic risk, with major consequences for automotive countries such as Germany.

Amidst all these challenges, a clear industry direction, or "true north", remains elusive. Western OEMs and suppliers must discern current trends and evolving consumer preferences to define their strategy and establish a focused development roadmap. This strategic approach is vital for wisely allocating limited R&D resources, avoiding hasty reactions to competitors' research and emerging trends in AD and EV technologies. Once the "true north" is identified, suppliers must surpass their reliance on traditional ICE OEMs and assert their own leadership and innovation by pioneering new business models and technological solutions. As competition and cost pressure intensify, suppliers need to adopt

modular solutions adaptable to various use cases and markets, moving away from custom component development based on OEM specifications. Instead novel methodologies and advanced tools are required to effectively translate evolving market trends into specific vehicle requirements.

It is within this context that this doctoral thesis aims to contribute. We aim to provide the necessary means to forecast the travel demand and usage pattern of future mobility concepts, such as shared autonomous electric vehicle (SAEV) fleets, in order to derive reliable driving cycles crucial for automotive prototyping. Through model-based exploration of SAEV fleets and their distinct driving characteristics, we seek to facilitate the derivation of component-specific load profiles for modular vehicle design. This enhances the adaptability and cost competitiveness of purpose-built vehicles while also enabling quantitative evaluation of new business models associated with SAEV ecosystems.

Before delving into the intricacies of SAEV drive cycle deduction, we first provide an overview of the techno-economic background motivating this area of study from the perspective of an automotive supplier. In Section 1.1, we explore in greater detail the challenges faced by the automotive sector, which compel suppliers to diversify their business area and innovate novel prototyping methods. Additionally, we examine the ongoing shift towards shared, electrified, and autonomous driving, driven by both national authorities and societal changes (Sec. 1.2). We then explore academic work (Sec. 1.3) on traditional automotive driving cycle deduction methods and their limitations in creating reliable driving patterns for emerging mobility concepts like SAEV fleets. Framed by this context and guided by existing research gaps, we outline our thesis objectives in Section 1.4.

1.1 Economic landscape of the automotive industry amidst technological transformation

Times are challenging for the global automotive industry with an unfavorable economic climate meeting the parallel emergence of disruptive technology changes. While supplier revenues peaked in 2018, the market has shrunk since 2017 due to various reasons [29]. Whilst the Dieselgate crisis led to huge image losses in the whole industry, Brexit-uncertainties and Trumpian US trade-wars

against China and Europe cooled down global markets and sales activities [25]. The Covid-19 pandemic intensified these issues in 2020, leading to widescale shutdown of assembly plants and supply chain disruptions, impacting sales and margins [26, 29]. Western markets, heavily dependent on exports and global supply chains, were particularly affected by the pandemic's aftermaths [29]. Nonetheless, by 2022/2023, the industry began to rebound, with OEMs reaching significant profit milestones [20, 21]. However, ongoing crises, including the Ukraine conflict, Russia's energy disputes with Europe, inflationary pressures, and upflaring Middle East conflicts, pose continued threats to global economic stability [154].

Amid economic instability, the automotive industry intensifies cost-cutting and restructuring efforts [26]. This comes at a time when substantial R&D investments are crucial to adapt to disruptive technological changes. The automotive industry, having followed a more or less linear technological development path over the last century, now faces the simultaneous emergence of mega-trends like the shared economy, autonomous driving, electrification, and digitization. These trends threaten to displace long-established OEMs within a decade by radically altering the performance metrics on which automotive companies compete [17, 118, 59]. Strained from the economic challenges, many OEMs and suppliers struggle to secure capital for transformation [26]. Their deep-rooted traditional business models hinder adaptation to disruptive technologies and evolving customer demands [118, 38], limiting their ability to develop essential competencies and partnerships for the future mobility landscape, including autonomous driving [26].

New market players and their market perception In 2017, Western OEMs faced increasing competition from Asian markets [22]. At the same time, tech giants and start-ups like Waymo are advancing swiftly in the autonomous vehicle (AV) sector, posing a mid-term threat to traditional OEM margins [24]. Despite challenges in software and mobility services, OEMs are showing more promising resilience in competition with Silicon Valley's tech firms as anticipated just a few years ago [20]. New EV players, including Rivian and Canoo, are forging strategic partnerships with firms like Amazon and Hyundai, positioning for future profitability [26].

These newcomers are gaining strong market reception, vital for financing innovation. While traditional OEMs sell more vehicles, new entrants achieve

comparable or higher market valuations, exemplified by Tesla's 2020 valuation exceeding the top five traditional OEMs combined [28]. Investors highly value these new players for their software and AI potential, viewing Tesla more as a tech company than merely an EV manufacturer [28, 18]. Zero-emission branding further enhances the market perception of new OEMs, positioning them as more future-oriented and environmentally responsible compared to traditional manufacturers [19]. Nonetheless, traditional OEMs are recovering market capitalization through EV commitments, such as Volkswagen's gigafactory plans [28] and its new strategic partnerships with EV OEMs like Rivian and XPENG [216, 217] to accelerate the development of their vehicle platforms.

Shifting consumer preferences and the rise of shared mobility The evolving automotive landscape reflects changing consumer preferences, especially among Millennials and Generation Z, who prioritize advanced technology, user experience, and safety, while increasingly relying on internet-based services for vehicle purchases and mobility solutions [118, 10, 178]. This shift has boosted online car transactions via platforms like Alibaba, eBay, and Carpal.com, challenging traditional OEMs and retailers [118]. In 2017, global online vehicle purchase consideration attained 11% in 2017 and grew from 25% to 30% in China by 2021 [17, 28, 16].

Environmental and resource efficiency concerns are reducing car ownership, especially in urban areas, where people opt for less maintenance-intensive options [118, 153]. Interest in fully autonomous mobility services grew from 52% in 2019 to 56% in 2020, yet the Covid-19 pandemic temporarily increased private vehicle reliance from 69% to 76% [27]. Asian countries like Singapore, China, and India are leading the adoption of new mobility solutions such as car sharing, ride-hailing, and taxi services, emphasizing transport efficiency over ownership [17]. In contrast, Western Europe and the US maintain a strong car ownership culture [24]. Despite a theoretical preference for shared mobility among Western youth [17], Germany has seen a rise in multi-vehicle households, suggesting continued personal vehicle preference among young Germans in 2022 [43, 74].

However, the continued rise of shared mobility is driving demand for Mobility-as-a-Service (MaaS), posing challenges to OEMs focused on vehicle sales [118]. By consequence, automotive manufacturers are transitioning to mobility service models with emerging sales channels like Car-as-a-Service (CaaS) that

offer all-inclusive monthly fees [23, 20]. Despite a decline during the pandemic, shared mobility demand has rebounded, with platforms like Uber and Lime recovering user bases [20].

In China, strong brand preference favors domestic brands like Nio, XPENG, and BYD, offering advanced technology and personalized features that resonate with the tech-savvy consumer base [19]. Traditional OEMs face significant risks if they fail to adapt to these preferences, prompting strategies tailored to regional markets [19]. Western manufacturers, including Mercedes-Benz, BMW, and Hyundai, are shifting towards personalized and made-to-order models to compete [19].

Relevance of the Chinese automotive market Although the Chinese industry did not establish a significant presence in the ICE-based vehicle market, they have now surpassed established automotive players in terms of EVs. Distinct among car manufacturing nations and unfettered by any ICE-related constraints, China foresaw the inevitable trajectory of electromobility and wholeheartedly embraced and championed its cause. In 2018, China emerged as the global leader in EV sales, accounting for over half of the market due to favorable regulations [23]. By mid-2021, the top-selling EV models in China were predominantly from local manufacturers [18]. In 2021, China sold nearly 3.4 million EVs, dwarfing Germany's 691,000, the second-largest market [19]. With 370 Chinese EV and PHEV models available in 2021, 85% of potential car buyers considered EVs [19].

By 2022, Chinese OEMs aimed to produce over half of China's vehicles by 2024, expanding their domestic market share [154]. This growth extends to the premium segment, traditionally dominated by European brands, potentially limiting Western companies' access to China's 30 million car production demand [154]. Failing to align with Chinese preferences could result in a 3.5% annual market decline through 2028 [19]. Supported by incentives and strong domestic demand, Chinese OEMs plan to increase their global market share, capitalizing on scale effects for cost efficiency [154].

Chinese OEMs leverage their EV leadership to drive global EV standards, as seen in Nio's strategic expansion of battery swap partnerships. Despite significant upfront investments, Nio has deployed over 2,300 battery swap stations and plans to add 1,000 more in 2024 [50], advancing its global battery

swap standards agenda. This forward-looking approach mirrors Tesla's historic investment in global fast-charging infrastructure despite initial profitability challenges. Today, Tesla's NACS charging connector standard is set to become the industry norm for EVs in North America, solidifying Tesla's position as the fast-charging network monopolist, while the CCS standard remains limited to Europe [192].

Pending breakthrough of autonomous driving In 2017, anticipation for autonomous driving breakthroughs was high, with leading CPU manufacturers targeting readiness for Level 4[1] autonomous electric vehicles by 2021 [17]. Commercialization efforts were evident, with the introduction of driverless "last mile" people movers (e.g., Auro, EasyMile or Navya in Singapore) and OEMs unveiling Level 4 vehicle roadmaps (e.g., large test fleets by GM in 2018, or first series models by BMW, Ford, Mercedes, and Audi) [17]. Growing interest in mobility services and autonomous driving drove a surge in employment, R&D investments, and patent filings within the industry [17, 22].

By 2018, however, initial predictions for widespread AV adoption have proven overly optimistic due to technological challenges [45, 24] and unsustainable business models, notably seen in ride-hailing firms like Uber and Lyft [25]. Their initial public offerings revealed significant losses and lack of profitability, suggesting reliance on investor cash to subsidize operations. This limited profitability led investors to demand tangible returns, resulting in a 32% decrease in mobility investments to USD 9.27 billion in 2019 from a peak of USD 21.4 billion in 2017 [26]. Updated estimates for full automation availability now vary widely, ranging from 2030 to 2050 [96, 150, 152].

Progress in the autonomous driving sector has further slowed in subsequent years due to still unresolved technical challenges, notably computing power requirements for Level 5 vehicles [20]. While Intel and NVIDIA have announced AD-specific semiconductors for 2025, their effectiveness is yet to be proven. Furthermore, the sustainability of earning revenue from AV services remains uncertain, with viable long-term business models still under development and no OEMs having established a sustainable model [18].

[1] Levels of vehicle automation, as defined by SAE (Society of Automotive Engineers) standards [106], range from automation level 0 (human driving) to 5 (fully automated driving).

The EV market finally comes of age In recent years, the EV market has witnessed remarkable growth and transformation, solidifying its position as a key segment in the automotive industry. Initially met with skepticism due to concerns about range, infrastructure, and cost, EVs have steadily gained traction, propelled by advancements in battery technology, government incentives, and growing environmental awareness. Powerful regulatory initiatives drive the adoption of EVs, cementing their importance. Governments, most notably China and the European Union plan to ban pure fossil-fueled vehicles by 2035 [176, 76], with additional nations following suit.

In 2017, global interest in EVs surged, particularly in Asia, with 37% of consumers considering an EV, a figure that skyrocketed to 60% in China [17]. While charging infrastructure constraints hindered EV adoption in emerging markets, high prices posed challenges in mature markets [17]. Established players began to allocate substantial resources to EV development, recognizing the inevitable transition from traditional combustion engines to electric drivetrains. This commitment has led to a significant increase in EV models. By 2018, the EV market featured over 250 electric or hybrid models, comprising 18% of the vehicle portfolio [24].

The momentum continued in 2020, with 50% of potential buyers considering an EV. Government stimulus and subsidies fueled this growth, tripling EV sales to 4.7% of total sales by the end of the year [25]. Tesla maintained its leadership in the EV market, but traditional OEMs and Chinese brands made significant strides by investing in EV technology [27]. Tesla's success can be attributed to its diversified approach extending beyond manufacturing to include CO2 trading, data services, recycling, and energy solutions. Its vertically integrated business model, spanning key technologies like batteries and e-powertrains, coupled with an online sales network, gave Tesla a competitive advantage, inspiring other OEMs to follow suit [27].

Despite the challenges posed by the Covid-19 pandemic in 2021, the EV market continued to grow robustly, with traditional automakers ramping up their EV investments and expanding their model offerings [28]. Funding in the sector soared, with 57% of top transport technology transactions in 2020 involving EV or battery companies, including industry players like Tesla, Rivian, Nikola, Waymo, and Gojek [28]. Charging infrastructure witnessed a threefold increase since 2017, and battery costs, which were over USD 300/kWh in 2015, have dropped to around USD 110/kWh and are projected to further decrease to

USD 60-70/kWh by 2030 [28]. The push to reduce emissions has accelerated electrification efforts, prompting OEMs like Volkswagen and BMW to target 50% battery-electric new cars by 2030. Volvo aims for full electrification by 2030, while Volkswagen is targeting carbon neutrality by 2050 [18].

However, the shift to electric vehicles presents challenges beyond production adjustments, raising concerns about job security due to reduced assembly needs [12, 93]. To mitigate these challenges, companies must diversify into new EV-related businesses and provide retraining opportunities for employees. For instance, the integration of electric mobility with smart homes and energy grids opens up new business sectors such as battery technology and energy market balancing. This expands the role of EVs beyond transportation, forming an environmentally conscious ecosystem.

Challenges facing Western automotive supplier Amidst all these geographical, economic upheavals, and socio-technological shifts, Western automotive suppliers are particularly hard hit. While China and South Asia are expected to fuel global production growth, Europe and North America may not reach peak volumes by the decade's end [154]. This trend especially affects European suppliers focused on domestic ICE markets, leading to their growing marginalization. As new players gain dominance, traditional suppliers risk missing out on future growth prospects.

Suppliers are struggling with thinner margins due to declining profitability from reduced volumes and scale efficiency. Supply chain disruptions and shifting OEM programs add market volatility, complicating planning. A skilled labor shortage, amplified by digital transformation and powertrain shifts, coupled with inflationary pressures, is driving up costs. OEMs reducing supplier contributions further erode profitability, while rising interest rates and a weakened sector perception in capital markets increase capital costs [154]. These challenges have permanently reduced EBIT (Earnings Before Interest and Taxes) margins by 3 percentage points, dropping from 7.5% in 2017 to below 5% in 2021/2022 [154].

The decline in profitability is further driven by the "normalization" of margins in China, stabilizing around 5.4% after aligning with Western standards [154]. EBIT margins vary widely, with larger suppliers showing more resilience to disruptions than smaller ones, which are often more vulnerable due to product

or market dependencies. Margin impacts also differ across sectors. Traditional automotive sectors see margin deterioration, while electronics and infotainment suppliers maintain margins through digitalization. ICE-focused suppliers face declining volumes, and those shifting to electrification grapple with rising R&D costs and scaling issues, impacting profitability. The transition from ICE further attracts new players to high-margin areas, with battery and non-automotive semiconductor and software suppliers achieving significantly higher margins than traditional suppliers and OEMs [154].

Despite slowing growth rates, the automotive supply industry is projected to grow at an average annual rate of approximately 4% from 2022 to 2030 [154]. Growth will primarily stem from software, electronics, and battery technology as ICE technologies wane and mechanical components become commoditized. Asian BEV OEMs, particularly Chinese companies like BYD, will be key drivers of demand. Except for Tesla, traditional North American and European powertrain suppliers are unlikely to substantially contribute to this growth with their existing portfolios [154].

As the industry shifts towards Asia, suppliers need to refocus on this region, considering Chinese OEMs as crucial customers and Chinese suppliers as strong competitors. Realigning regional activities, product portfolios, and supply chains allows suppliers to access Asian growth opportunities and reduce geopolitical risks [154].

Embracing digitalization and innovative strategies to diversify revenues
As the automotive industry transitions to software-driven value chains, OEMs and suppliers need to pivot from traditional mechanical and electrical engineering focus to embrace computing and AI expertise. Embracing technological advancements like software, battery technology, and digitalization is vital to stay competitive and expand beyond traditional areas [154]. Software is becoming the pivotal differentiator and revenue catalyst [21]. Given the capital-intensive nature of this transformation, forming strategic partnerships is crucial to pool resources, share risks, and navigate industry shifts effectively [154].

As consumers gravitate towards affordable and sustainable mobility solutions beyond traditional ICE car ownership, there is a rising demand for purpose-built vehicles with lower costs, reduced emissions, and enhanced safety [22]. Traditional OEMs need to adapt to these changing market demands while

catering to the appetite for innovative, personalized products in tech-focused markets like China. Subscription-based models, like Functions-on-Demand (FoD)[2], could play a pivotal role in the future by enabling faster adaptation to changing preferences and providing opportunities for continuous customer engagement and data collection [21].

OEMs and automotive suppliers need to reduce their reliance on ICE technologies and diversify revenue streams. Exploring alternative business models such as power management, ride-sharing platforms, assistance systems, and location-based content is crucial. Urban ride-sharing presents significant business opportunities across design, maintenance, cloud services, and payment management. Subscription-based models, such as MaaS and CaaS, stabilize OEM revenues and can boost customer retention compared to one-time sales. The anticipated shift towards demand-responsive ride-sharing fleets, like SAEVs, allows companies to position themselves as leading providers of mobile computational environments or secure long-term maintenance contracts with cloud-based transportation clients. However, a deep understanding of emerging mobility concepts is essential for developing innovative services and products in this evolving landscape.

The Western automotive industry needs to stay attuned to current trends and changing consumer preferences to establish a clear development path. This strategic approach is crucial for prudent allocation of limited R&D resources, avoiding impulsive responses to competitors' research, and emerging trends in AD and EV technologies. Making strategic investment decisions, prioritizing growth areas, and considering consolidation or exit strategies for non-core activities are imperative [154]. More specifically, suppliers must reduce their dependence on traditional ICE OEMs and drive their own leadership and innovation by pioneering new business models and technological solutions. As competition and cost pressure rise, suppliers should embrace modular solutions adaptable to diverse use cases and markets, moving away from custom component design based on OEM specifications. Innovative prototyping methodologies and advanced tools are essential for effectively translating evolving market trends into specific vehicle requirements.

[2] Functions on Demand refer to services or features offered on a subscription basis, allowing customers to access specific car functionalities as needed rather than purchasing them outright.

1.2 A new era of urban mobility

The prospect of improved employment, education, and healthcare opportunities continues to fuel urbanization, with over half of the world's population residing in cities as of 2018. The United Nations (UN) projects a global urbanization rate of 68% by 2050 [211]. In addition to rural migration, the world's growing population contributes to urban expansion. While the global population was expected to reach 8 billion in 2022, UN projections now estimate 8.5 billion by 2030 and 9.7 billion by 2050 [70]. This increasing number of city residents leads to a growing demand for both passenger and freight transport [152]. Given that current urban mobility is already plagued by traffic congestion, road accidents, air and noise pollution, these issues are likely to worsen without intervention. Fundamental shifts towards sustainable mobility systems are necessary to balance environmental and economic considerations while enhancing travel comfort and improving the quality of life for city residents.

In the EU, the transport sector accounts for a quarter of total greenhouse gas (GHG) emissions, with road transport responsible for the largest portion, representing 72% of these emissions in 2019 [5]. Despite climate and energy policies, the transport sector is the only major sector where GHG emissions have risen above 1990 levels [4]. Between 1990 and 2019, total transport-related GHG emissions increased by over 33%, and road transport emissions rose by nearly 28% [5]. Current efforts to reverse this trend are inadequate to meet the EU's 2050 climate neutrality target [5]. Under prevailing policies in 2016, transport-related GHG emissions were projected to rise slightly between 2030 and 2050, exceeding the targeted 60% reduction compared to 1990 levels [4]. The more recent initiatives by the European Parliament in 2022/2023 show more promise, with aspirations for a complete elimination of emissions from new car sales by 2035 [176]. Yet, the tangible impact on transport-related GHG emissions is still uncertain, as recent amendments to the law permit ICE models for sales and registration post-2035, with the stipulation that they operate exclusively on carbon-neutral fuels [215].

In addition to GHG emissions, poor air quality levels in urban areas pose significant public health challenges [4]. Although transport-related emissions of air pollutants, such as particulate matter (PM), nitrogen oxides (NOx), and sulfur oxides (SOx), have decreased since 1990, over 300,000 premature deaths occur annually due to chronic exposure to fine particulate matter (PM2.5)

alone [3]. Local traffic remains a primary contributor to urban air quality issues and is a major source of noise pollution. Chronic noise exposure can lead to annoyance, stress reactions, sleep disturbances, and increased risks of hypertension and cardiovascular disease [4].

So far, efforts to mitigate the negative impacts of traffic have seen limited success due to various reasons. Technological advancements in the transport sector often fail to counterbalance the growing demand for transport [4]. Additionally, several barriers hinder the transition to sustainable mobility. The automotive industry's interests, for instance, prioritize strengthening existing products over adopting new technologies [212]. Environmentally questionable subsidies, such as tax breaks for company cars, financial incentives for commuting expenses, and tax concessions for diesel fuels, further complicate the shift. The current costs of electric vehicles and the long lifespan of transport vehicles also pose challenges to adopting more sustainable mobility solutions [4].

While many of these barriers can be overcome with appropriate incentives, so-called "lock-ins" present more significant challenges. These lock-ins result from past decisions that have solidified the system into a specific state or technology [4]. For example, due to the prevailing dominance of roads in both passenger and freight transport, the majority of infrastructure investments are directed towards road development. This further strengthens the dominance of cars and roads, limiting investments in more sustainable modes of transport [4]. Other lock-ins include ongoing efforts to improve ICEs instead of transitioning to zero-emission vehicles and policies that continue to favor diesel use in Europe [4]. Additionally, cognitive lock-ins, stemming from resistance to change and familiarity with existing technologies, hinder the adoption of new, superior solutions [107, 161].

In addition to these barriers and lock-ins, there are other disincentives, including the absence of financial penalties, that hinder systematic changes in the transport system. For example, environmentally harmful transport modes are not adequately penalized. Since 1996, the cost of purchasing motor cars has decreased significantly compared to average consumer prices, while the cost of more sustainable shared passenger services and their operational costs has generally risen [4]. Another disincentive is the lack of inclusion of transport externalities in the consumer's final price. The global organization of the food system, combined with relatively low transport costs and a wide variety of

transport options, leads consumers to purchase food regardless of season and geographical origin, contributing to increased freight transport [4].

Until now, city planners and transport engineers have traditionally addressed traffic issues by predicting future traffic growth and constructing new road capacity to accommodate projected demand [227]. While evidence suggests that constructing new highways or expanding existing ones improves traffic flow, it also leads to an increase in travel demand, exacerbating congestion and air pollution issues [4, 227]. An alternative approach to traditional traffic planning is the "Avoid-Shift-Improve" (ASI) model. This model aims to achieve key performance goals for the transport system by balancing both supply and demand with low GHG emissions. The ASI model is based on three principles: (i) avoiding unnecessary travel activity, (ii) shifting travel demand from less efficient to more efficient modes, and (iii) improving the efficiency of traffic activity [227]. However, this approach has not yet been widely adopted.

As cities become denser, using limited road space for low-capacity, resource-intensive modes like privately owned passenger cars (PCs) is impractical [45]. PCs, which are parked 95% of the time, occupy valuable urban space, highlighting the inefficiency of a car-dominated transport system [156]. The future of urban mobility should prioritize public transport (PT) modes that efficiently pool individual travel needs in larger, yet flexible, vessels [45]. As numerous transfers between routes are a major cause of discomfort for today's PT users, it is vital to optimize the connectivity between PT routes to approach seamless urban mobility [45].

Recent technological advancements offer new mobility solutions in this regard. The concept of shared mobility stands out as a promising approach in urban areas, reducing the negative impacts of car-dominated transport while increasing user satisfaction [152]. Transitioning towards zero-emission modes, notably electric vehicles, further mitigates urban mobility's negative externalities such as noise and air pollution [152]. Additionally, advancements in autonomous driving promise more efficient land use, reduced energy consumption, improved traffic performance, and enhanced cost-effectiveness and safety [152].

However, the introduction of new mobility trends, particularly autonomous vehicles, has been met with both anticipation and skepticism. While AVs offer potential benefits like efficient driving, safer vehicles, and enhanced shared and on-demand mobility services [196], their introduction requires careful

management to avoid undesired effects [128]. For instance, the enhanced comfort of non-shared robotaxis could diminish the use of public transport, leading to greater capacity-space inefficiencies and a rise in individual car usage [97, 132, 110]. On the other hand, some studies suggest that AVs may primarily cater to irregular trips during evenings and weekends, rather than replacing the routine trips typically associated with public transport and private car usage [90, 73]. This suggests a potential increase in the overall number of vehicles on the road, rather than their substitution. Additionally, the expected traffic flow improvements from AVs may be limited by the presence of human-driven cars. These conventional vehicles are unable to fully leverage the benefits of real-time information sharing between vehicles (V2V) and infrastructure (V2X) to prevent accidents and ensure smooth traffic flow. Consequently, the actual benefits of AVs considerably depend on their penetration level, shared mobility adoption, and interaction with other transport modes [196]. Moreover, while AVs may reduce accidents, their effects on congestion and pollution are still uncertain [45]. Inefficient bundling of rides could result in extra empty mileage [162, 34, 202], which may counteract some of the environmental advantages, such as reduced energy consumption from smoother driving.

The integration of shared mobility, electromobility, and AD offers a promising solution to these challenges. However, the introduction of demand-responsive transport (DRT) modes such as SAEV fleets strongly depends on user acceptance, technological advancements, policy frameworks, and economic factors [152]. While shared mobility addresses significant challenges and may reduce private car ownership, it may also face social constraints by not adequately addressing comfort factors. People's natural resistance to change, coupled with uncertainty about AVs, affects their willingness to adopt new mobility solutions [89, 163]. Moreover, automation also introduces complexity across industries, leading to job losses, moral dilemmas, and challenges in human-machine interactions [45, 152]. As for EVs, concerns about global sustainability remain, particularly regarding the environmental impact of battery production and vehicle lifecycle [152]. Consequently, strong political leadership through regulatory interventions is crucial to ensure that new mobility concepts are both sustainable and economically viable [81, 57]. As AV technology evolves, collaboration between policymakers and manufacturers becomes essential to avoid policy misalignment with technological advancements [200].

1.3 Driving cycles in vehicle prototyping and methods for SAEV drive cycle deduction

The scope of the preceding sections was to elucidate the external forces currently shaping the automotive sector. While Section 1.1 addressed the economic landscape of automotive suppliers, emphasizing their need to explore new business areas and reduce dependence on OEMs through novel prototyping techniques, Section 1.2 outlined the ongoing transition towards more sustainable urban travel modes, such as SAEV fleets. Previously published in [209], this section now provides a concise literature review on the significance of driving cycles in automotive prototyping and the limitations of conventional methods in generating reliable driving patterns for new mobility concepts.

Standardized driving cycles are crucial in the automotive industry, particularly for state-of-the-art emission modeling, performance prediction and virtual prototyping [117, 65]. Hence, the deduction of realistic and representative driving cycles has been the focus of research for many decades [117, 7, 171]. Driving cycles are typically defined as time-velocity profiles at a second-by-second level and can be classified into two categories: modal and transient cycles. Modal cycles are simplified and consist of different idling, linear acceleration and steady speed phases. On the other hand, transient cycles reflect for real-life driving behavior under on-road conditions [65].

The New European Drive Cycle (NEDC) is a standardized modal cycle developed to assess emission levels of ICE vehicles, consisting of two subsets: the ECE-15 for urban driving and the EUDC for high-speed motorway driving [117]. Despite criticism for its outdatedness and lack of representativeness for real-world driving conditions, the NEDC remains widely used in the automotive sector. However, in 2017, the Worldwide harmonized Light vehicles Test Procedure (WLTP) was introduced by the United Nations Economic Commission for Europe to replace the NEDC as the European vehicle homologation procedure. In contrast to its predecessor, the Worldwide harmonized Light vehicles Test Cycle (WLTC) [210] is transient, and its data are collected through a chase-car method by instructed drivers [117]. Besides NEDC and WLTC, many other legislative driving cycles have been developed by governmental organizations for various regions and applications. For instance, in Japan, the modal 10-15 mode or more recently the JC08 cycle are employed, while the United States

rely on the transient Federal Test Procedure cycle (FTP-75) developed by the US Environmental Protection Agency.

As legislative driving cycles fail to capture typical driving behaviors across different nations, various techniques have been developed to generate new national or regional driving cycles [7, 117, 171, 82]. A summary of conventional methodologies can be found in [222]. The most basic approach involves synthesizing driving cycles by combining different idling, acceleration and steady speed phases. However, this modal approach leads to unrealistic dynamics in the transition zones, resulting in inaccurate emission estimates [7, 52, 112].

Another common technique used to generate new driving cycles involves four main steps: route selection, data collection, data clustering, and cycle generation [230, 7, 222]. Route selection refers to choosing the route for data collection, which can be gathered using on-board measurement, GPS tracking, and/or the chase-car method. On-road measurements generally provide the most accurate reflection of the selected route but are subject to a strong bias due to unusual congestion patterns, necessitating repetitive measurements [230]. In contrast, the chase-car method is less expensive and involves following target vehicles randomly while mimicking their driving behavior. However, this approach has the drawback of neglecting careful route selection. The collected profiles are often divided into micro trips[3], which are then clustered based on traffic conditions, vehicle type, or other key performance indicators (KPIs). Common techniques for clustering trips include k-means cluster algorithms [225, 82] or hybrid approaches that combine k-means and support vector machine (SVM) clustering [230]. Despite their validity, cluster methods often demand significant computational resources [7]. Subsequently, the final driving cycle is typically constructed from a pool of available micro trips [7, 222, 230]. The concept behind the micro-trip-based methodology is to identify micro trips that adequately represent real-world driving diversity but in a more condensed manner, which is practical and cost-effective for data collection [222, 7]. Typically, algorithms select micro trips based on predefined performance measures. Alternatively, Monte Carlo methods can generate multiple candidate cycles by randomly selecting several micro trips and evaluating their KPIs. The

[3] A micro trip denotes a trip between two idling phases.

most suitable cycle in terms of performance is then chosen as the final driving cycle.

Another statistical technique involves utilizing real-world driving data to generate synthetic driving cycles through Markov chain processes. As demonstrated in [171, 194, 86], this method constructs a transition probability matrix of a Markov chain using measured velocity profiles. Each element in the matrix represents a particular state determined by current velocity and acceleration, with transition probabilities stored within each state for moving from one state to another. Another data-driven approach to derive driving cycles is referred to as route information mapping, introduced in [65]. This method emphasizes the importance of incorporating external conditions such as weather, traffic, and terrain data. A similar approach is taken in [83], where data on slope, road curvature, and speed limit are combined with traffic information and driver models to form a control problem that is numerically solved to generate velocity profiles. However, such data-driven approaches require large databases of GPS-tracked driving cycles, detailed maps, and/or open access to traffic information.

All previous methods for deducing driving cycles suffer from a significant drawback: they rely exclusively on measured data, which makes them unsuitable for capturing the driving behavior of autonomous vehicles that differ from human-driven profiles. Therefore, autonomous driving behavior is often approached by flattening human driving profiles [131, 8, 95]. To this end, researchers have proposed various approaches, including filter or smoothing techniques such as moving average, local polynomial regression, kernel density estimation, and smoothing splines. These methods have also been applied in motion planning or trajectory optimization for AVs [149, 223]. The smoothing approach is justified by kinetosis prevention[4] and by the vehicle's improved perception of traffic conditions, thanks to advanced sensors and car-to-everything (C2X) communication. Nevertheless, smoothing techniques may erase idling times (e.g., in stop-and-go patterns or at red signals) and fail to capture platooning effects or connected driving in a methodologically sound manner.

Conventional approaches to deduce representative driving cycles also fail to capture the unique driving patterns of emerging mobility concepts, such as

[4] To ensure the passenger's well being, the lateral and longitudinal acceleration is limited.

Table 1.1: Typical driving characteristics of privately owned vehicles (PCs) compared to shared, autonomous electric vehicle (SAEV) fleets.

travel mode	PC	SAV	SAV	SAEV
source	[170]	Tab. 2.5	[49]	[49]
data acquisition technique	survey	transport simulation (MATSim)		
region	Germany	Berlin	Austin (Texas)	
∅ no. of daily trips	1.9	24.4	27.4	15.2
∅ mileage per trip [km]	-	6.7	8.2	8.3
∅ daily operating time [h:mm/car]	0:46	7:24	-	-
∅ daily mileage [km/car]	30	92.3	253	137

SAEV fleets. Unlike conventional private cars, SAEVs used for urban passenger transport have distinct driving characteristics (cf. Table 1.1), including higher daily mileages, increased uptime, and shorter driving ranges limited by battery capacity. As vehicle development becomes more intricate, obtaining detailed SAEV driving profiles becomes essential for virtual prototyping to guarantee optimal on-road performance in future fleet deployments.

As traditional methods fall short in addressing these types of questions, traffic simulation, particularly microscopic simulation, has gained increasing popularity. Macro, meso, and microscopic traffic simulations vary primarily in the scale and level of detail with which they represent traffic flow and vehicle interactions, as illustrated in Figure 1.1. Microscopic frameworks have been employed for cost-optimized generation of driving cycles [7] and to assess the effects of automated driving on fuel consumption [102, 185, 199]. The research in [102] assesses VISSIM's[5] ability to replicate real-world driving cycles, highlighting that while the simulated profiles satisfyingly emulate aerodynamic speed, they inadequately capture acceleration patterns. This discrepancy arises from human drivers' tendency to accelerate more aggressively at lower speeds, a nuance overlooked by the simulation, which also neglects stochastic variations around the target velocity. Similar findings are reported in [7], which integrates microscopic traffic simulation and micro-trip-based methods to derive representative driving cycles. The authors highlight that default parameters in microscopic

[5] PTV GROUP, https://www.ptvgroup.com/de/loesungen/produkte/ptv-vissim/. Accessed: Sep. 29, 2024.

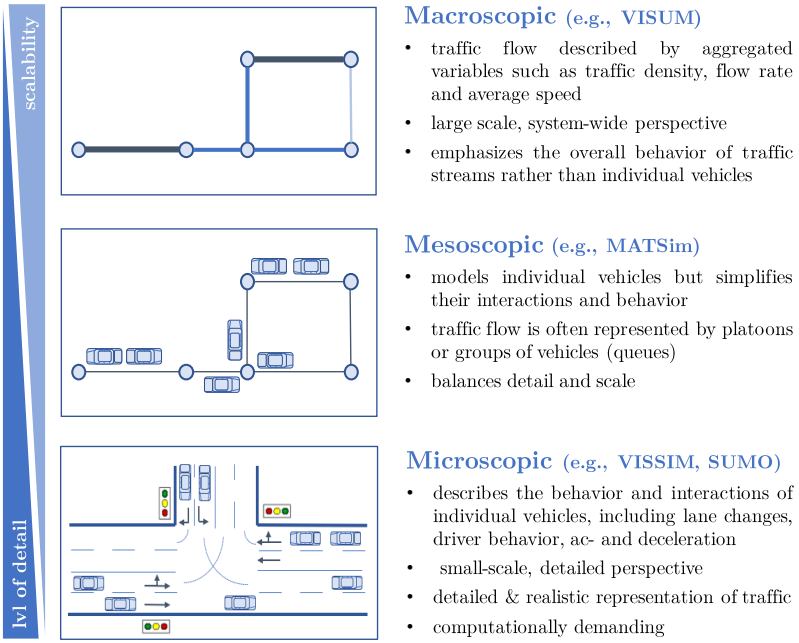

Figure 1.1: Schematic representation illustrating the differences between macro, meso, and microscopic traffic simulations, highlighting their varying scales and levels of detail in representing traffic flow and vehicle interactions.

simulation lead to unrealistic driving behavior, with simulated velocity profiles being overly aggressive due to gradients often set to the vehicle's maximum acceleration capability. This aspect is also addressed in [199], underscoring the importance of a well-calibrated traffic model for generating quality driving cycles. Similarly, the relevance of microscopic traffic models for assessing the impact of traffic strategies on fuel consumption is scrutinized in [185]. The authors underscore that microscopic traffic simulation models commonly face validation challenges when it comes to capturing driving dynamics. Despite generating detailed velocity profiles, these models typically prioritize macroscopic objectives relevant to transportation planning. As a result, they are often calibrated based on aggregated parameters such as traffic flow, density, speed, or queue length, rather than considering instantaneous vehicle speed

and acceleration [199, 185]. Therefore, the speed profiles generated by these models are often oversimplified and may not be suitable for environmental studies or automotive requirement engineering. Despite these limitations in capturing human driving behavior, microscopic traffic simulation tools may be well-suited for fully automated driving scenarios, as they involve fewer stochastic variables.

In regard to SAEV fleet driving characteristics, extensive literature exists exploring the acceptance, simulation, and impact of (autonomous) vehicle fleets. AV fleets are anticipated to augment network capacity through connected driving and enhanced safety [155]. Conversely, AVs may also amplify traffic volumes due to increased travel demand resulting from improved travel comfort, additional empty rides, and smaller vessel sizes compared to public transportation modes [97]. AV fleet simulations have been scrutinized from various angles due to their disruptive nature. In this regard, the mesoscopic multi-agent transport simulation framework MATSim [98] stands out as a well-established simulation tool. For instance, studies such as [34] and [72] simulate the citywide replacement of private cars with shared autonomous vehicle (SAV) fleets in Berlin and Austin, respectively. Both investigations indicate that each SAV has the capacity to replace ten privately owned cars. In [155], the impact of SAVs on general trip statistics, such as average trip duration and vehicle-km traveled, is analyzed using a case study for Munich. Other studies delve into the influence of various SAV pricing schemes and service areas on mode choice, as outlined in [132, 110]. The results suggest that when AV prices are low, their market share increases, leading to a decrease in the use of public transportation. Moreover, if SAVs operate within small service areas only, this may inadvertently discourage walking and cycling in favor of autonomous fleets. Several studies have investigated the electrification of SAV fleets and its implications for charging infrastructure planning [229, 35, 134, 49]. While SAEVs are more cost-effective in fleet operations compared to their combustion counterparts, the high initial investment costs for batteries and charging infrastructure pose challenges to their electrification. However, these costs can be significantly mitigated through optimal routing and relocation strategies [229]. The impact of routing and dispatching algorithms on taxi services is extensively discussed in the literature, as seen in [138, 140, 48]. However, while MATSim excels in large-scale fleet simulation, mesoscopic traffic simulation tools lack the necessary level of detail to accurately simulate the dynamics of individual vehicles [203].

In conclusion, existing publications often focus on generating driving cycles, modeling autonomous driving behavior, and simulating SA(E)V fleets but frequently overlook detailed vehicle dynamics. These individual approaches, while valuable, prove inadequate when applied to broader SAEV contexts and lack the accuracy required for vehicle prototyping. To our knowledge, there is currently no comprehensive approach that integrates all these aspects.

1.4 Thesis objective and structure

Building upon the context presented in Sections 1.1 and 1.2, and addressing the research gap identified in Section 1.3, we now elaborate on our research endeavor: the development of a model-driven approach to generate comprehensive 24-hour SAEV driving profiles tailored for automotive prototyping requirements. Through model-based exploration of SAEV fleets and their inherent driving characteristics, we seek to facilitate the derivation of component-specific load profiles essential for modular vehicle design. This not only enhances the adaptability and cost competitiveness of purpose-built vehicles but also enables quantitative evaluation of new business models within SAEV ecosystems.

Developing these driving cycles with virtual vehicle prototyping in mind leads us to delineate six essential key requirements (KRs) for SAEV driving profiles, as previously outlined in [209]. These KRs also serve as performance benchmarks for assessing the achievements of this thesis.

In contrast to traditional references that focus exclusively on speed profiles, our definition of a "driving profile" includes a variety of interconnected time-series that capture a vehicle's driving pattern over the course of the day. KR0/1/5 address essential requirements for our modeling approach, while KR2-4 represent interlinked time-series of physical and categorical quantities intended for virtual vehicle prototyping.

- *KR0 – DRT travel demand:* The widespread adoption of DRT modes is accompanied by significant uncertainties, making future travel demand difficult to predict. Thus, we need a method to assess future DRT demand and its interaction with existing transport modes, enabling responsiveness to scenarios like city-wide PC bans and traffic-calmed zones. Social acceptance of DRT greatly impacts SAEV fleet size and mileage.

- *KR1 – SAEV fleet driving patterns:* For automotive prototyping, we need driving profiles that reflect vehicle movement across urban regions over 24 hours, accounting for EV range and charging constraints, as well as for various DRT fleet routing and dispatching strategies. In technical component design, understanding both average and worst-case loads is crucial. Thus, a modeling approach is needed that not only generates average SAEV driving profiles but also accommodates a range of fleet vehicles.
- *KR2 – Vehicle states:* Furthermore, the driving profiles should include vehicle states like *idling*, *relocating*, *charging*, or *occupied* to facilitate optimal climate control and battery preconditioning.
- *KR3 – Detailed velocity profiles:* Crucially, we need second-by-second speed profiles that account for diverse driving styles[6], road congestion, and transport infrastructures. As we plan to explore additional AV-related topics in the future, such as vehicle platooning and connected driving, we aim to integrate relevant interfaces into our toolchain.
- *KR4 – Environmental data:* Given the road network's topography and the driving cycle's impact on component design, additional time-series like altitude or occupancy profiles are important. While accurate EV charging levels (SOC) are secondary, as they will be determined in vehicle simulation, estimates should still be considered in transport simulation to account for realistic driving ranges and charging times.
- *KR5 – Feasibility, scalability and transferability:* The developed approach must be practical regarding data availability, automation capacity, and applicability to different urban areas of varying sizes.

We structure the thesis into three parts. Part I examines the potential of single-level traffic simulation methods in creating authentic SAEV driving profiles. Single-level traffic simulation approaches focus on modeling transportation systems at a specific level of detail or abstraction. Unlike multi-level simulation approaches, which incorporate different levels of complexity to represent various aspects of the transportation system, single-level simulations typically

[6] Spanning from automation level 0 (human driving) to 5 (fully automated driving) as per SAE standards [106].

operate at either a macro, meso, or microscopic level (cf. Figure 1.1). In Chapter 2, we employ an existing MATSim model to incorporate demand-responsive travel modes, evaluating the suitability and constraints of mesoscopic transport simulation for depicting SAEV fleets in multi-modal transport systems. Moreover, Chapter 3 enriches mesoscopic driving cycles derived from MATSim simulation with real-world driving data to enhance the accuracy of driving dynamics. This chapter also explores microscopic traffic simulation with SUMO, providing a model-based solution to this challenge while concurrently reducing the need for resource-intensive on-road measurements.

In Part II, the core of the thesis, we present a multi-level traffic simulation approach aimed at integrating the benefits of large-scale mesoscopic transport planning with the detailed insights provided by microscopic traffic simulation. We introduce automated procedures in order to seamlessly bridge both domains, enabling the construction and population of microscopic SUMO models based on mesoscopic MATSim simulations. In Chapter 4, we conceive a robust network matching algorithm aimed at translating mesoscopic network elements into their microscopic equivalents and vice versa. This chapter serves as a foundational step for subsequent work. Additionally, Chapter 5 presents a method to refine MATSim's sampled travel demand for more detailed microscopic simulations while also constraining computation requirements. Chapter 6 underscores the importance of maintaining consistency between micro- and mesoscopic traffic simulations. Here, we evaluate the coherence of both MATSim and SUMO in terms of traffic dynamics and network capacity through various analytical and numerical experiments, providing invaluable insights for the final toolchain design. Furthermore, in Chapter 7, we introduce a novel calibration approach aimed at synchronizing spatio-temporal network states in multi-level traffic simulation.

As an example application in the automotive context, we showcase the toolchain's feasibility through a case study on EV powertrains in Part III. In Chapter 8, we illustrate how distinct driving patterns and behaviors of SAEV fleets, compared to privately owned cars, result in altered power engine requirements through vehicle simulation. This study marks the culmination of our research efforts. Finally, Chapter 9 presents the conclusions of the thesis, along with a brief outlook.

Part I

Capabilities of Single-Level Simulation Approaches

2 Driving patterns of demand-responsive ride-sharing fleets – a mesoscopic simulation study with MATSim

Adopting a top-down approach, we embark on our investigation by exploring suitable single-level methodologies for modeling city-wide demand-responsive transport in Section 2.1. Employing a comprehensive MATSim simulation study focused on the city of Berlin (Section 2.2), we forecast future DRT travel demand using a scenario-based approach and assess the impact of the DRT travel mode on the global urban transport system (Section 2.3). Shifting to a fleet-centered perspective, we analyze typical fleet driving characteristics (Section 2.4) and explore the limits of mesoscopic transport simulation by deriving detailed driving cycles for individual fleet vehicles (Section 2.5). Additionally, we examine the uncertainties inherent in mesoscopic DRT simulation (Section 2.6) and relate them to our specific use case. Parts of the work presented in Sections 2.2, 2.3, and 2.4 have been included in [120, Sec. 4.7].

2.1 The multi-agent transport simulation framework MATSim

2.1.1 Fundamentals of the MATSim cycle

MATSim[1] [98] is a multi-agent transport simulation framework widely embraced in the scientific community for large-scale fleet simulations. It facilitates

[1] MATSim, https://www.matsim.org/. Accessed: Sep. 29, 2024.

activity-based transport modeling, where each user is depicted as an individual agent adhering to a daily plan. These plans comprise activities (e.g., home, work, leisure) with specific spatio-temporal properties, linked by trips of various transport modes (e.g., car, bicycle, public transportation). The traffic assignment process employs a co-evolutionary algorithm, with agents iteratively optimizing their daily schedules, competing for space-time resources until quasi-equilibrium is attained (Fig. 2.1).

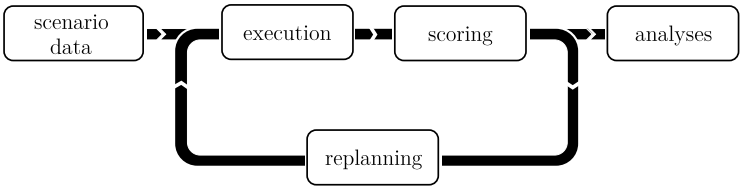

Figure 2.1: Schematic illustration of the MATSim cycle, adapted from [98]

In the initial stage (referred to as *execution* or *mobsim*), all agents' activity chains are concurrently executed within a traffic simulation. This enables the modeling of interactions among traffic users, including effects such as congestion. To facilitate the modeling of large-scale scenarios, MATSim employs a spatial queue model to approximate traffic dynamics [85, 197, 46]. This model offers sufficient detail for addressing most transportation-related questions while sidestepping the need for computationally intensive and intricate car-following models. A more detailed discussion of MATSim's queue model is presented in Section 2.5.

During the second step (*scoring*), the plan executed by each agent undergoes evaluation based on mode-specific utility functions and predetermined behavioral parameters unique to each agent. The total utility of a plan comprises three components: (i) the utility derived from activity engagement, (ii) the generalized travel cost or trip-related disutility (e.g., travel time or monetary payments), and, if relevant, (iii) the daily mode-specific disutility (e.g., costs of car-ownership) [47, 101]

$$S_P = \sum_{q=1}^{N} S_{\text{act},q} + \sum_{q=1}^{N} S_{\text{trav},q} + \sum_{m \in M} S_{\text{day},m}. \tag{2.1}$$

Here, S_p represents the total utility of plan p, N denotes the number of activities within plan p; $S_{\text{act},q}$ signifies the positive utility gained during activity q; $S_{\text{trav},q}$ indicates the negative utility associated with trip q to reach activity q; $S_{\text{day},m}$ represents the daily disutility attributed to the utilization of transport mode m, and M represents the set of available transport modes.

In MATSim, activities are presumed to take place within a 24-hour timeframe, with the initial and final activities of the day being merged [47]. The agent's score steadily rises during activity engagement. The travel disutility for a single-leg trip q is expressed as [101]

$$\begin{aligned} S_{\text{trav},q} &= \chi_{\text{trip}}^{m(q)} + \beta_{\text{m}} \cdot \Delta f_q + \beta_{\text{transfer}} \cdot x_{\text{transfer}}^{q} \\ &+ \beta_{\text{trav}}^{m(q)} \cdot t_{\text{trav}}^{q} + \beta_{\text{wait}}^{m(q)} \cdot t_{\text{wait}}^{q} \\ &+ (\beta_{d}^{m(q)} + \beta_{\text{m}} \gamma_{\text{d}}^{m(q)}) \cdot d_{\text{trav}}^{q} \end{aligned} \tag{2.2}$$

where $\chi_{\text{trip}}^{m(q)}$ is a mode-specific constant per trip; β_{m} is the marginal utility of money; Δf_q is the change in monetary budget induced by fares or tolls during trip q; β_{transfer} is the transfer penalty if a public transport (PT) trip involves a transfer from one transit line to another; x_{transfer}^{q} takes a value of 0 for PT trips without line transfer and 1 for trips involving a transfer; $\beta_{\text{trav}}^{m(q)}$ is the marginal utility of time while traveling with mode $m(q)$; t_{trav}^{q} is the travel time of trip q; $\beta_{\text{wait}}^{m}(q)$ is the marginal utility of time spent waiting for transport mode $m(q)$, and t_{wait}^{q} the corresponding waiting time; $\beta_{\text{d}}^{m(q)}$ is the marginal utility of distance traveled by mode $m(q)$; $\gamma_{\text{d}}^{m(q)}$ is the monetary distance rate for mode $m(q)$, and d_{trav}^{q} denotes the distance traveled on trip q.

In the third step of the MATSim cycle, known as *replanning*, each agent opts for a new plan to execute in the subsequent iteration. Most agents employ a multinomial logit model to select the optimal plan from their previously executed ones. Additionally, a portion of agents, typically around 10%, generate a new travel plan by adjusting route choice, mode choice, and departure time choice in an existing plan, while discarding the plan with the poorest performance.

Through iterative execution of all three steps, the agents select plans with higher utilities, thus improving their behavior by learning. This process leads

the system towards stochastic user equilibrium[2], where each agent optimizes his plan egoistically. The evolutionary approach of the MATSim method is particularly suitable for studying the impact of transport-related policy measures or the integration of new transport modes such as SAEV fleets.

2.1.2 Simulation of demand-responsive transport (DRT)

In MATSim, (autonomous) taxi fleets are categorized as demand-responsive transport modes. Modeling such systems entails addressing the dynamic vehicle routing problem (DVRP, [179]), a time-dependent version of the well-known vehicle routing problem (VRP, [60]). The DVRP focuses on determining optimal (usually minimal-cost) routes for a fleet of vehicles to serve a designated set of customers.

Recently, modules for DRT schemes have been incorporated into the MATSim framework. The core functionalities are provided by the `dvrp` extension [141, 137], which implements the concept of dynamic (ride) requests. These requests are fulfilled by a fleet of vehicles, which are managed by the `VrpOptimizer` that responds to all DVRP-related events, such as the submission of ride requests or the boarding and alighting of passengers from fleet vehicles.

The `dvrp` extension offers the essential interfaces for configuring DRT schemes, enabling extensive customization of the framework. However, the actual functionality of the supply-and-demand matching algorithm executed in the optimizer must be implemented by additional MATSim extensions such as `taxi` [139], `sav` [142], or `drt` [32]. In this thesis, we employ the `drt` extension, which we will briefly describe next.

The `drt` extension allows for both station-based and door-to-door DRT services, and unlike the `taxi` extension, it supports ride pooling where multiple passengers with different pickup and drop-off locations can share the same vehicle. When a new ride request is received, the extension incorporates it into the existing DRT routes in a way that minimizes the vehicle's additional operation time.

[2] Nash-equilibrium is a state where no individual can improve its satisfaction by unilaterally changing its behavior (≠ system optimum) [165, 164, 98].

The ride pooling implementation in the `drt` extension is governed by two constraints [32]. Firstly, the total ride time t_r for any additional passenger sharing the same vehicle must not exceed a specified threshold, defined as

$$t_r <= \alpha t_r^{direct} + \beta. \tag{2.3}$$

Here, t_r^{direct} represents the direct ride time without detours, while α and β are parameters that capture the maximum allowable time loss due to waiting, boarding, and detours arising from the pickup and drop-off of other passengers. It is crucial to emphasize that this ride time constraint does not extend to the requesting user. Secondly, both the already scheduled customers and the new passenger must have expected boarding times that fall within a specified time range t_{wait}.

The fare for a DRT ride of distance d and duration t is calculated as follows

$$f_{drt}(d,t) = \max(f_{min}, f_{base} + f_{dist}d + f_{time}t), \tag{2.4}$$

where f_{min} is the minimum fare, f_{base} the base fare per ride, f_{dist} the fare rate per distance, and f_{time} the fare rate per time. The total fare f_{drt} of each ride is added as a negative term to the MATSim utility function of the agent performing the corresponding ride.

2.1.3 Modeling of electric vehicles (EVs)

MATSim enables the simulation of electric vehicle fleets and their charging infrastructure through the utilization of the `ev` extension[3]. This updated version of the `TransEnergySim` contribution [218, 219, 33] allows for the definition of EV-specific attributes, such as battery capacity and initial state of charge (SOC), as well as charging infrastructure-related details, such as location, charging power, and the number of charging plugs. The `dvrp` extension now includes the former `evrp` contribution, addressing EV routing problems by accounting for limited availability of charging vehicles and the increased range constraints of low-energy vehicles.

[3] Source code for EV contribution in MATSim 15.x https://github.com/matsim-org/matsim-libs/tree/master/contribs/ev. Accessed: May 04, 2023.

The charging module within the ev extension provides several charging strategies, including fixed-speed charging for traditional AC[4] charging, fast-then-slow charging to mimic DC[4] fast-charging, and customizable variable-speed charging strategies. Various charging logics consider the availability of different charging stations, occasionally leading to vehicle queuing at busy stations. Upon reaching a charging station, the charging model determines the energy needed for the electric vehicle to recharge to its desired state of charge and the time required for this process. A simplified battery model, assuming a linear relationship between the battery's SOC and its available energy, is employed.

The discharge module in MATSim differentiates between driving and idling energy consumption. Idling energy refers to the constant energy drain of auxiliary devices at given temperatures. Driving energy consumption can be simulated using (i) user-specified fixed energy consumption, (ii) a look-up-table approach based on a physical EV model [67], or (iii) a macroscopic average-speed energy consumption model [172]. The look-up-table approach reads a csv file containing precalculated energy consumption for given road slopes and vehicle speeds, inherently accounting for aerodynamic drag, rolling resistance, drive-train efficiency, and regenerative braking. In contrast, the average-speed energy consumption model relies on statistical evidence regarding the relationship between average speed and other statistics describing speed profiles at the level of individual road sections.

2.2 Scenario definition

With the fundamentals of MATSim introduced, we now aim to leverage its existing DRT and EV-related contributions to simulate SAEV fleets in a realistic study.

[4] AC and DC denote two distinct types of electric current. AC, or alternating current, reverses its direction periodically, commonly used in residential and commercial settings. DC, or direct current, flows steadily in one direction and is frequently employed in batteries and electronic devices.

2.2.1 Base scenario: The MATSim Open Berlin Scenario

As the base case for this simulation study, we utilize the MATSim Open Berlin Scenario [234] in release version 5.3.[5] This scenario comprises a MATSim transport simulation for the Berlin metropolitan area, encompassing the principal transport modes of the present-day Berlin transport system: private car as driver (*car*), private car as passenger (*ride*), bicycle (*bike*), walking (*walk*), and public transport (*pt*).

The road network is derived from OpenStreetMap[6] (OSM), encompassing all roads within the city borders of Berlin down to the level of residential streets. In the surrounding areas of Brandenburg, only major roads are included (see Fig. 2.2). Public transport is simulated based on real-life schedules. Bike and walk modes are approximated using beeline distances and modeled with a constant speed.

The Open Berlin Scenario incorporates travel demand by considering the daily activity chains[7] of the adult population of the German federal states of Berlin and Brandenburg (approximately 4.7 million individuals). These activity chains are generated from a fully synthetic, person-specific, econometric activity scheduling model detailed by [235]. For instance, Fig. 2.3 illustrates the spatial distribution of home and work locations for all agents in Berlin and its immediate vicinity.

The network and activity chains serve as input for the agent-based transport simulation with MATSim. In the Open Berlin Scenario, the MATSim utility parameters were calibrated to ensure that the simulation accurately replicates the observed traffic patterns (mode and distance distributions, as well as traffic counts) of present-day Berlin [234, 77]. The resulting scenario is sensitive to measures regarding route, mode, and departure-time choices. The corresponding utility parameters are summarized in Table 2.1.

[5] The data and code required to execute the scenario are freely available on GitHub at https://github.com/matsim-scenarios/matsim-berlin/tree/5.3.x. Accessed: Mar. 23, 2020.

[6] OpenStreetMap, https://www.openstreetmap.com. Accessed: Sep. 29,2024.

[7] The model represents the following activity categories: home, work, leisure, shopping, and other. These are combined into daily activity chains, e.g., home-work-home or home-work-home-leisure-shopping-home.

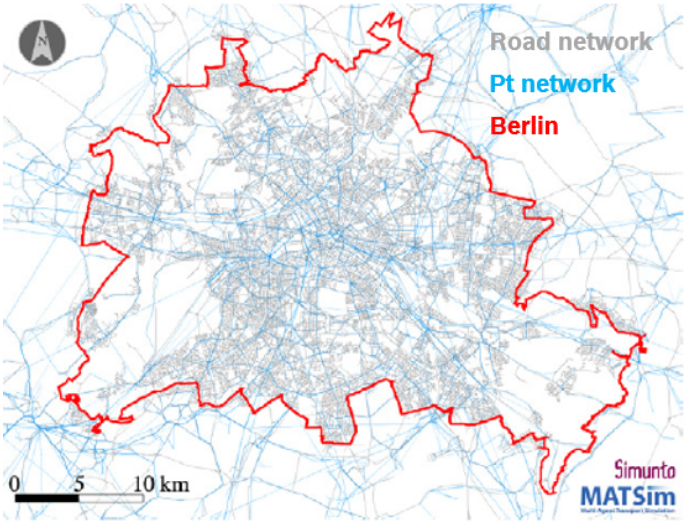

Figure 2.2: Road and public transport network in the central area of the Open Berlin Scenario. The red line indicates the border of Berlin state (figure adapted from [234]).

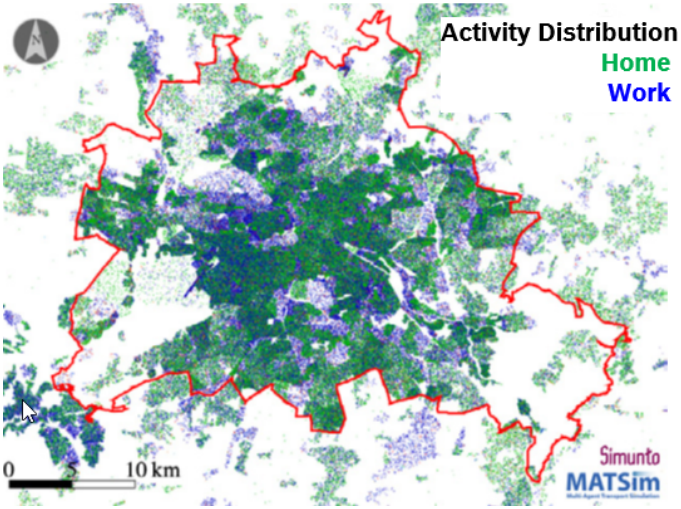

Figure 2.3: Distribution of home and work locations of agents in the central area of the Open Berlin Scenario. The red line indicates the border of Berlin state (figure adapted from [234]).

Table 2.1: Utility parameters of the Open Berlin Scenario

	bicycle	car	pt	walk
α_{day} (EUR)	0	-5.3	-2.1	0
α_{trip} (EUR)	-1.9	-0.525	0	0
α_{dist} (EUR/km)	0	0.20	0	0

For reasons of computational efficiency, we conduct the following DRT simulations only for a subsample of the population, following a common practice in the MATSim community [e.g., 98]. Typically, 10% samples are used, where each simulated agent represents 10 individuals in reality. However, due to the substantial CPU time required by the current DRT implementation in MATSim—where a 10% sample simulation takes about a month to run—we have chosen to use a 1% sample (approximately 470,000 agents) for this study. Consequently, when analyzing the simulation data, we scale the number of trips (and related quantities) by a factor of 100 to estimate the actual transport performance. In Section 2.6.4, we discuss how small sample sizes affect the statistical significance of our findings.

2.2.2 Future DRT fleet scenarios

We defined two future scenarios where the Berlin transport system is expanded with a large fleet of autonomous taxis offering DRT services. For simplicity, EV range and charging constraints are not simulated at this stage, as they increase runtime without being critical for this thesis, as these functionalities are already integrated into the MATSim framework. The first scenario targets the broad market with an affordable ride-pooling DRT service, accommodating up to 5 passengers in a single taxi vehicle (P5-10af). The second scenario caters to a more premium market segment with an individual DRT service where customers do not share rides in the taxi vehicles (I-10af). Therefore, the two scenarios have significantly different fares (see Table 2.2). Further considerations, such as sensitivity to different fleet sizes, fare levels, congestion pricing, and competition between fleet operators, are addressed in Section 2.6.2.

Table 2.2: DRT fares and key cost parameters of selected DRT fleet scenarios

scenario	base	P5-10af	I-10af
f_{min} (EUR)	-	1.0	2.0
f_{base} (EUR)	-	0.0	0.0
f_{dist} (EUR/km)	-	0.18	0.30
f_{time} (EUR/min)	-	0.0	0.0
$\chi^{\text{car}}_{\text{daily}}$ (EUR)	5.3	10.6	10.6
$\chi^{\text{car}}_{\text{dist}}$ (EUR/km)	0.2	0.2	0.2

In both of our future scenarios, we utilize the Open Berlin Model as the base case. We introduce the DRT service as a new travel mode and employ MATSim to simulate how this new service influences the mode choice of agents and, consequently, the traffic patterns. To achieve this, we conduct 500 *mobsim* iterations in MATSim until the system reaches a new user equilibrium. During the first 400 iterations, 5% of the agents modify their plans to explore new modes, departure times, and routes, respectively.[8] In the last 100 iterations, plan innovation is disabled, and agents can only choose between previously known plans.

We implement the DRT service on a city-wide scale, within the political borders of the state of Berlin, as depicted by the red line in Fig. 2.2. All agents with at least one trip within the service area are considered potential DRT customers, allowing them to explore the new DRT mode in the simulation. DRT passengers are served in a door-to-door mode, meaning they are picked up directly at their origin location and dropped off at their destination. For the ride time constraint described in Eq. 2.3, we set $\alpha = 1.5$ and $\beta = 1200\,\text{s}$ [cf. 32]. The limit on the expected boarding times is set to $t_{\text{wait}} = 1200\,\text{s}$. Furthermore, we assume that the usage of private cars will be discouraged by regulatory measures such as road tolls or increased taxes. We approximate such policies by doubling the daily car costs $\chi^{\text{car}}_{\text{daily}}$ in our future scenarios compared to the Open Berlin base scenario (see Table 2.2). Costs play a significant role in the attractiveness of

[8] For agents using private cars as non-driving passengers (ride mode), the transport mode is fixed.

transport modes. Thus, increasing the daily car costs discourages the use of private cars, making other modes more appealing.

Initially, our fleet vehicles are randomly distributed within the service area. Starting from iteration 2, the vehicles retain their last position from the previous iteration. Idle taxis remain stationed at their last known position, without any rebalancing[9]. In this pilot study, we assume that there is an "unlimited" supply of taxi vehicles[10], thereby disregarding the effects of capacity bottlenecks. The combination of no rebalancing and an unlimited number of fleet vehicles is anticipated to result in a significantly lower proportion of empty rides compared to a profit-optimized fleet setup from the perspective of a mobility service provider. Profit optimization is outside the scope of this study.

2.3 Transport system analysis

2.3.1 Impacts of DRT on global mode choice and shift

In Fig. 2.4, we display the mode share distribution by trip count (left panel) and travel distance (right panel) for our future scenarios P5-10af and I-10af, compared to the Open Berlin reference data (base). This analysis considers only trips originating within the Berlin city borders, which align with the operation area of our DRT service.[11] By increasing the daily costs of private cars and offering an efficient autonomous taxi service, the contribution of cars to cumulated travel distances decreases notably from 22.6% (base) to 11.0% for the pooling scenario (P5-10af) and 8.7% for the individual scenario (I-10af). Concurrently, the new DRT service captures a substantial fraction of the cumulated travel distance, accounting for 11.5% and 27.4% for P5-10af and I-10af, respectively (cf. Table 2.3).

[9] In fleet management, rebalancing refers to the process of strategically relocating vehicles within a fleet to optimize their distribution across a service area. This is typically done to ensure that vehicles are positioned where they are most likely to be needed, thereby maximizing service efficiency and minimizing customer wait times.

[10] Technically, the fleet size in the simulation is limited, but the limit is set to a value well above the maximum number of cars requested.

[11] The mode distribution for the complete Open Berlin scenario, which includes all of Brandenburg and thus rural areas, exhibits a significantly higher usage of private cars than in the city of Berlin.

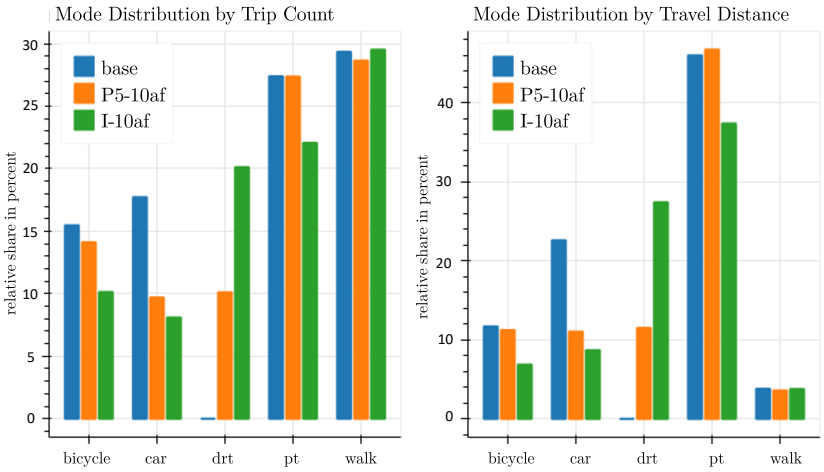

Figure 2.4: Distribution of mode share by trip count (left panel) and travel distance (right panel). Car users as passengers (ride mode) are excluded, as their fraction remains constant.

In the DRT pooling scenario P5-10af, the share of bike and walk modes decreases slightly, while the usage of public transport mode increases marginally. Conversely, in the DRT individual scenario I-10af, bike (6.9%) and public transport (37.4%) travel distances decrease significantly compared to the base case (11.7% and 46% for bike and public transport, respectively), as the new DRT mode offers much shorter travel times (cf. Table 2.3).

The DRT service similarly affects the mode distribution by trip count (cf. Table 2.4). However, the relative shares of the standard modes differ notably from the distribution weighted by trip distance, primarily due to the prevalence of short-distance walking trips.

We also note that the total distance covered by private cars and autonomous taxis behaves differently from the cumulated travel distance: while the vehicle mileage of 82,800,000 km in the pooling scenario P5-10af is over 10% below the vehicle mileage of the base scenario (91,700,000 km), the individual scenario I-10af results in a slight increase in vehicle mileage to 92,800,000 km.

The Sankey plots in Figs. 2.5 and 2.6 provide a detailed visualization of the mode-shifts observed in the ride-pooling and individual DRT scenarios, respectively.

Table 2.3: Mode share by trip distance in percent

	base	P5-10af	I-10af
bicycle	11.7	11.2	6.9
car	22.6	11.0	8.7
drt	0.0	11.5	27.4
pt	46.0	46.8	37.4
ride	15.9	15.9	15.9
walking	3.8	3.6	3.8

Table 2.4: Mode share by trip number in percent

	base	P5-10af	I-10af
bicycle	15.5	14.1	10.1
car	17.7	9.7	8.1
drt	0.0	10.1	20.1
pt	27.4	27.4	22.1
ride	10.1	10.1	10.1
walking	29.3	28.6	29.5

In the ride-pooling DRT scenario P5-10af, the DRT mode primarily attracts previous car and bicycle users, with only a small number of pedestrians and public transport users switching to the DRT mode. Additionally, due to the increased daily car costs relative to the base scenario, a significant portion of former car users also transition to bicycle and public transport modes in P5-10af. Conversely, in the individual DRT scenario I-10af, the DRT mode is considerably more appealing and attracts substantial proportions of former car, bicycle, and public transport users.

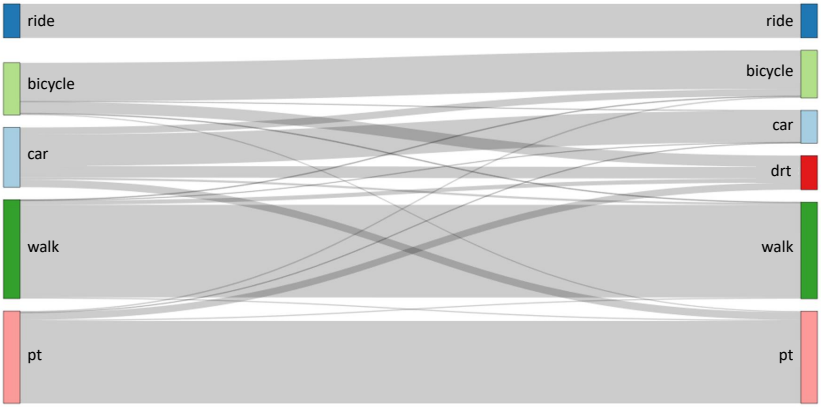

Figure 2.5: Mode shift from the base scenario (left) to the P5-10af model with ride-pooling DRT service (right).

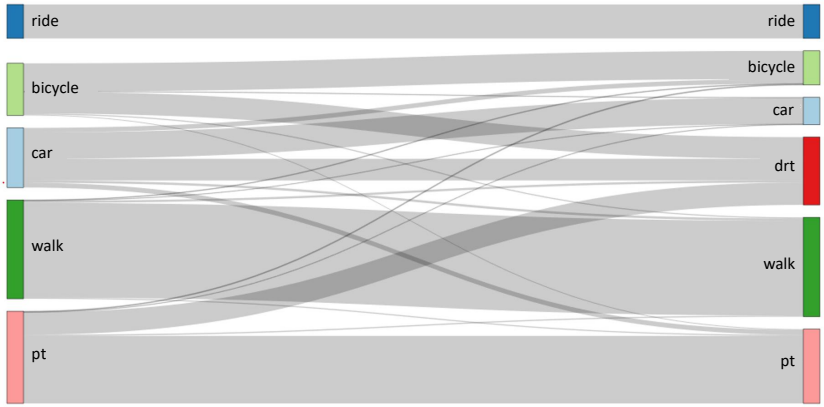

Figure 2.6: Mode shift from the base scenario (left) to the I-10af model with individual DRT service (right).

2.3.2 Performance evaluation of the DRT service

Pooling scenario P5-10af In our pooling scenario, the simulation forecasts 1,050,200 DRT trips within a 24-hour period. These trip requests are handled by 76,400 taxi vehicles, with a maximum of approximately 24,000 being active simultaneously. The temporal distribution of ride requests, displayed in the top left panel of Fig. 2.7, exhibits a sharp increase around 6 am, peaking at about 40,000 requests per 30-minute interval around 10 am, followed by a gradual decline in the evening hours. In the top right panel of Fig. 2.7, vehicle occupancies throughout the simulation are depicted using different colors. Despite being a pooling service, the majority of vehicles predominantly carry a single passenger, with some accommodating two or three passengers. Instances of higher occupancy are rare. Moreover, there are always some fleet vehicles without passengers, such as when en route to a pickup. The envelope of the occupancy profile represents the total number of active fleet vehicles at each time interval, peaking at approximately 24,000 vehicles.

Waiting times for DRT passengers are depicted in the middle panels of Fig. 2.7. The overall distribution, shown as a histogram in the left panel, exhibits a prominent peak for waiting times shorter than 120 s, followed by a gradual decline. This distribution results in an average waiting time of approximately 450 s. Throughout the day, average and median waiting times remain relatively stable, with slightly lower values observed during nighttime, as illustrated in the right panel. However, the maximum waiting time exhibits significant variation over time, with occasional outliers surpassing one hour, likely attributed to local congestion near the passenger's location.

The distribution of travel distances for all DRT trips, illustrated in the bottom left panel of Fig. 2.7, exhibits a prominent peak at approximately 3 km, followed by a rapid decline and an extended tail extending up to about 50 km. This distribution results in an average trip distance of approximately 8 km. The daily mileage distribution of all fleet vehicles displays a broader profile, with a peak around 90 km and a total mileage of 7,054,500 km, as shown in the bottom right panel of Fig. 2.7. The daily empty mileage per vehicle is predominantly centered at zero, with a total empty mileage of 566,900 km (empty factor[12] 0.08). None of the taxi vehicles travel more than 30 km without passengers.

[12] empty mileage divided by vehicle mileage

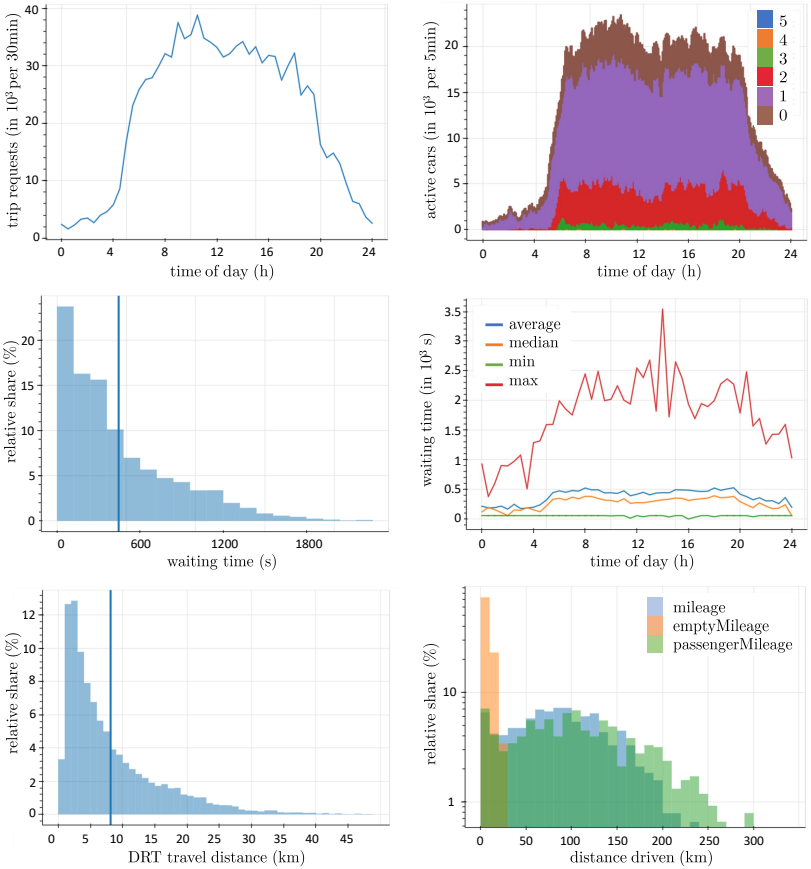

Figure 2.7: Selected KPIs for DRT model P5-10af. *Top left:* Number of ride requests over time. *Top right:* Occupancy profile of fleet vehicles over time (passenger count is color coded). *Middle left:* Distribution of waiting times for all ride requests. The vertical line represents the average waiting time (448.8 s). *Middle right:* Waiting profile over time, showing average, median, minimum, and maximum waiting times per 30-minute interval. *Bottom left:* Distribution of DRT ride distances, with the average ride distance (8.1 km) indicated by the vertical line. *Bottom right:* Histograms depicting vehicle driven-mileage distribution (blue), vehicle empty-mileage distribution (orange), and vehicle passenger-mileage distribution (green).

Due to pooling effects, the passenger mileage distribution per vehicle extends to larger values compared to the vehicle mileage, resulting in a total passenger mileage of 85,346 person-kilometers and a revenue factor (passenger mileage divided by vehicle mileage) of 1.21.

Individual taxi scenario I-10af For our individual taxi scenario, the simulation predicts 2,089,500 DRT trips over a 24-hour period, served by a total of 142,500 fleet vehicles. This trip volume is significantly higher than the 1,050,200 trips of the pooling scenario, indicating greater user preference for the taxi service. This preference is evident in shorter waiting times for the individual scenario (averaging around 350 s, as shown in the middle panels of Fig. 2.8) compared to the pooling scenario (around 450 s). Additionally, ride times are shorter in the individual scenario since fleet vehicles do not need to deviate for other passengers' pickups. As the negative impact on agents' utility from travel episodes (Eq. 2.2) is directly tied to travel time (sum of waiting and ride time), individual DRT rides are more attractive than pooled rides (assuming comparable fares). In our simulation, this attractiveness persists even though the individual taxi service is more expensive than the pooled offering.

The temporal distribution of ride requests (top left panel of Fig. 2.8) exhibits a broad peak between 9 am and 4 pm, with steep flanks on both sides. Vehicle occupancy (top right panel of Fig. 2.8) follows a similar profile, reaching a maximum of approximately 70,000 simultaneously used fleet vehicles around 4 pm. In an individual taxi service, the maximum occupancy is limited to one passenger per vehicle. Similar to the pooling scenario, a non-negligible fraction of taxis operate without passengers, particularly while en route to a pickup.

The distribution of travel distances for all DRT trips (bottom left panel of Fig. 2.8) peaks at approximately 5 km, slightly higher than in the pooling scenario (3 km), reflecting the higher minimum fare. DRT travel distances extend up to approximately 50 km, resulting in an average trip length of about 8 km, comparable to the pooling scenario. The daily mileage distribution of all fleet vehicles exhibits a broad peak around 130 km, with a total mileage of 18,624,000 km (bottom right panel of Fig. 2.7). The daily empty mileage per vehicle is sharply peaked at zero, totaling 1,739,900 km (empty factor 0.09). No vehicle travels more than 40 km without passengers. Without pooling effects, the total passenger mileage (1,684,000 person-km) falls short compared to the vehicle mileage, resulting in a revenue factor less than 1 (0.91). This

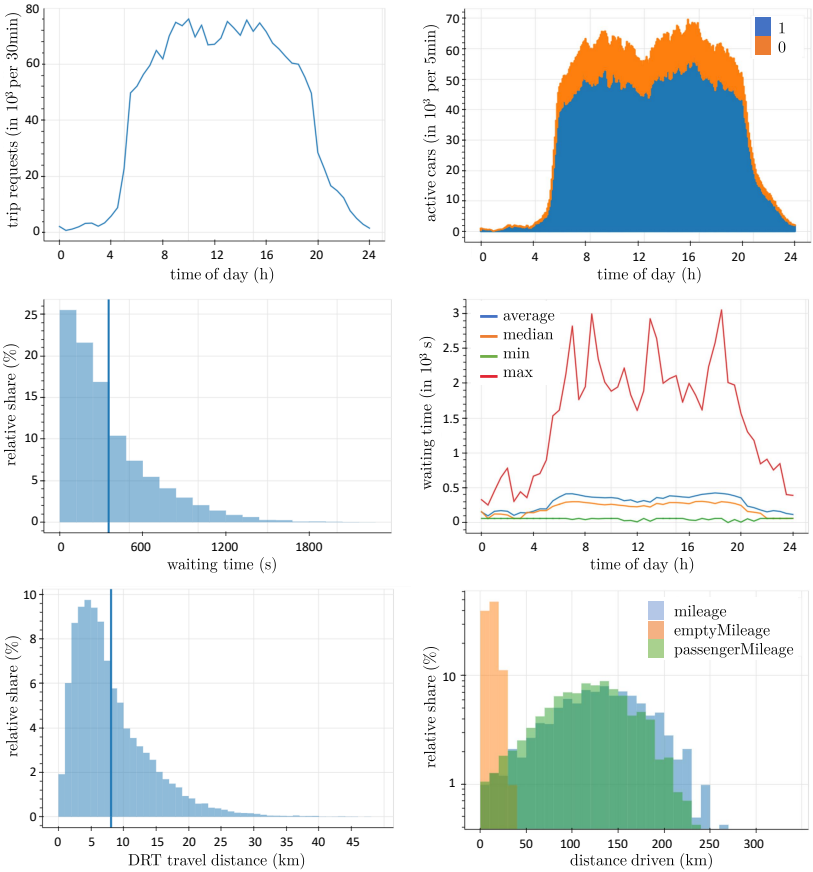

Figure 2.8: Selected KPIs for scenario I-10af of the DRT model. *Top left:* Number of ride requests over time. *Top right:* Occupancy profile of fleet vehicles over time, with the number of passengers color coded. *Middle left:* Distribution of waiting times for all ride requests, with the average waiting time (353.2 s) indicated. *Middle right:* Waiting profile over time, displaying average, median, minimum, and maximum waiting times per 30-minute time bin. *Bottom left:* Distribution of DRT ride distances, with the average ride distance (8.1 km) marked. *Bottom right:* Histograms illustrating the distribution of vehicle driven mileage (blue), vehicle empty mileage (orange), and vehicle passenger mileage (green).

discrepancy is also evident in the passenger mileage per vehicle distribution, which covers a narrower range than the vehicle mileage distribution.

2.4 Average driving patterns of ride-sharing fleets

2.4.1 Aggregated performance metrics of DRT fleets

For easy comparison, we compile key metrics that characterize the DRT system performance in Table 2.5. One aspect to note regarding fleet utilization is that, as discussed in Section 2.2, the fleet size for this pilot study has been intentionally set to a notably large number to avoid capacity bottlenecks within the DRT system. Consequently, in the P5-10af scenario, only about a third of the utilized DRT vehicles are simultaneously active (or approximately 50% in the I-10af scenario). To ensure economically viable fleet operation, a mobility service provider will likely seek to improve this ratio by reducing fleet size. However, this reduction may lead to decreased vehicle availability, thereby

Table 2.5: Selected KPIs of the DRT system. Simulation data with fleet context (*) have been scaled by a factor of 100 to estimate the actual transport performance.

	P5-10af	I-10af
# DRT trips*	1,050,200	2,089,500
# DRT vehicles in use (24h)*	76,400	142,500
max # simultaneously active DRT vehicles*	24,000	70,000
DRT mileage (km)*	7,054,500	18,624,000
DRT emtpy mileage (km)*	566,900	1,739,900
DRT passenger mileage (pkm)*	8,534,600	16,884,000
empty factor	0.08	0.09
revenue factor	1.21	0.91
∅ mileage per taxi and day (km)	92.3	130.7
∅ empty mileage per taxi and day (km)	7.4	12.2
∅ passenger mileage per taxi and day (km)	111.4	118.5
∅ waiting time (s)	444.8	353.2

lowering the Quality of Service (QoS). Additionally, the service area plays a pivotal role, substantially influencing both QoS and fleet profitability. Future research could optimize scenarios for fleet size and QoS, leveraging demand prediction and optimized DRT dispatching and routing.

2.4.2 Identification of representative fleet vehicles

Based on suitable ensemble estimates $\bar{\xi}_j$ (where j may be of type mileage, operation time, number of rides ...), it is possible to select representative vehicles for automotive requirement engineering. To this end, for each vehicle i, we calculate the L2 distance of this vehicle's attributes, $\xi_{i,j}$, to a set J of ensemble estimates

$$d_i = \sqrt{\sum_{j \in J} (\xi_{i,j} - \bar{\xi}_j)^2} \tag{2.5}$$

and minimize the distances d_i to find the representative vehicle.

Since different attributes have quite different value ranges and units, Eq. 2.5 may lead to distances that are dominated by deviations in only a subset of the selected attributes. To avoid this problem, we use a slightly modified formula to calculate the distance

$$\tilde{d}_i = \sqrt{\sum_{j \in J} \left(\frac{\xi_{i,j} - \bar{\xi}_j}{M_j} \right)^2}, \tag{2.6}$$

where $M_j = |\max(\xi_{i,j} - \bar{\xi}_j)|$ ensures a normalization of the different distance contributions.

Table 2.6 provides a summary of vehicle attributes corresponding to the ensemble median (P50) and 95th percentile (P95) in simulations P5-10af and I-10af, using the attribute set $J = \{\text{mileage, operation time, number of rides}\}$ defined in Eq. 2.6. Contextual fleet statistics for both scenarios are also included. Although mean and median values are similar, P95 values show greater variability from the ensemble centroids, also highlighted by relatively large standard deviations (σ) and attribute outliers (max).

Figure 2.9 showcases essential ensemble metrics from the P5-10af simulation using a plot matrix. In this visualization, the distributions of labeled attributes

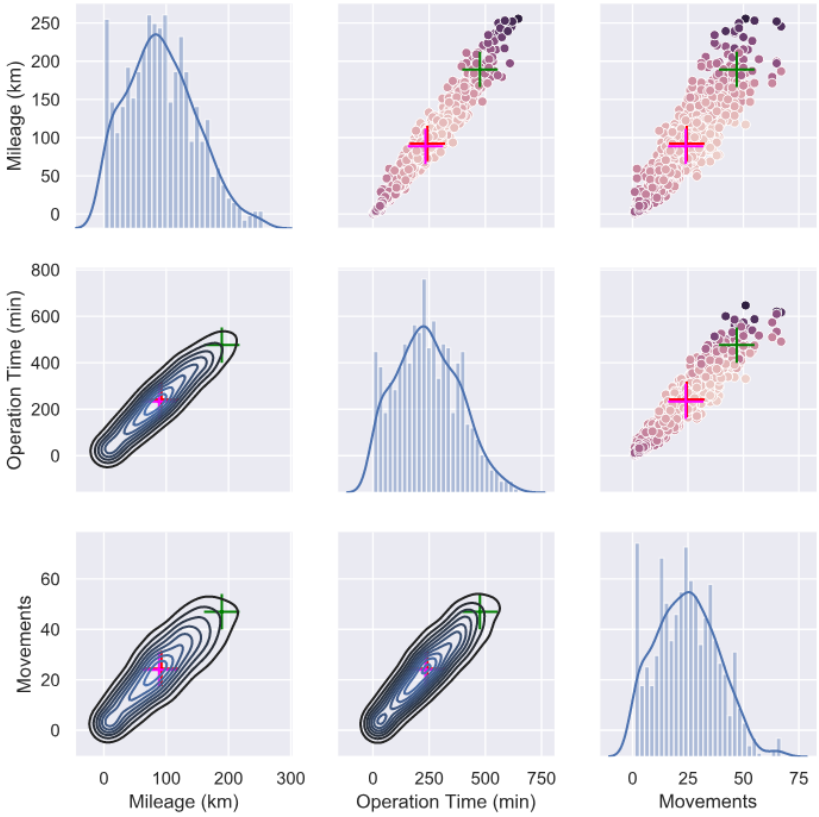

Figure 2.9: Plot matrix for scenario P5-10af, illustrating DRT simulation data. The descending diagonal displays distributions for the labeled attributes $J = \{\text{mileage, operation time, number of rides}\}$, while bivariate scatterplots above the diagonal show color-coded L2 distances from mean values. Below the diagonal, contour plots provide kernel density estimates, with some decile lines omitted for clarity. Mean and median values are indicated by red and magenta crosses, respectively, and the P95 value is represented by a green cross.

Table 2.6: Performance metrics for the ensemble median (P50) and 95th percentile (P95) in DRT simulations P5-10af and I-10af, alongside contextual fleet statistics.

		mileage (km)	operation time (min)	number of movements	max. velocity (km/h)	∅ velocity (km/h)
P5-10af	σ	55.4	138.9	13.7	25.3	4.4
	min	0.4	1.9	1	14.9	7.6
	∅	92.3	242.2	24.4	82	22.5
	P50	88.5	233.5	24	80	22.1
	P95	189.2	476	47	119	29.4
	max	255.7	646.6	67	120	58.8
I-10af	σ	53.2	141.9	10.4	19.5	2.8
	min	2.0	6.4	1.0	23.9	13.0
	∅	130.7	365.9	26	90.3	21.4
	P50	131.4	374.3	27	80	21.2
	P95	216.9	580.0	42	119.2	26.2
	max	317.9	846.6	58	120.0	45.4

are displayed along the descending diagonal. Above the diagonal, bivariate scatterplots feature color-coded L2 distances to the mean values. Below, contour plots depict kernel density estimates, with each contour line representing a decile. Mean and median values are marked with red and magenta crosses, respectively, while the P95 value is denoted by the green cross.

2.5 MATSim's limits with respect to detailed driving dynamics

With MATSim demonstrating its suitability for modeling the daily driving patterns of SAEV fleets in city-wide scenarios, our focus now shifts to evaluating MATSim's ability to replicate realistic vehicle driving dynamics, crucial for automotive prototyping. Portions of this section have been previously published in [209, 207].

2.5.1 MATSim's spatial queue model

MATSim is *microscopic* in terms of demand modeling but *mesoscopic* with respect to traffic dynamics. Despite its spatial queue (SQ) model, MATSim ignores the agents' exact position on a link to save computational resources. To produce more realistic flow dynamics, the queue model [85, 197, 46] was extended with elements of the *link transmission model* [224] by introducing backward traveling holes [1]. As a result, the SQ model now resembles Newell's *simplified kinematic wave model* [167] and is also consistent with the *LWR-model-family* [129, 182][13]. Under stationary conditions, the extended queue model features an idealized fundamental diagram which is defined by the minimum of the density-dependent *sending function* $S(\rho)$ and *receiving function* $R(\rho)$. As depicted in Figure 2.10, the FD results in an overall triangular shape parametrized by the free-flow speed v, the backward wave speed w, the ultimate (flow) capacity $\hat{q}$, and the maximum density $\hat{\rho}$.

[13] Refer to [78, 1] for more on how MATSim relates to existing traffic flow theory.

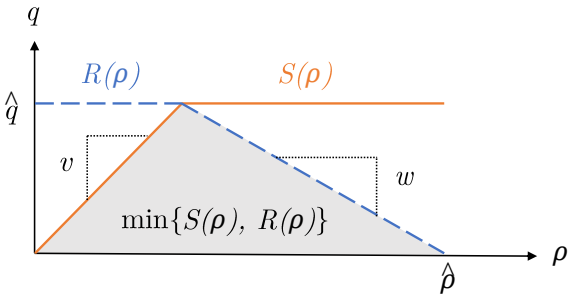

Figure 2.10: Idealized fundamental diagram defined by the minimum of the density-dependent sending function $S(\rho)$ and receiving function $R(\rho)$ (adapted from [78]). In a stationary case on a homogeneous road segment, the average flow q through any cross-section is described by this diagram based on the road's average vehicle density ρ. Flow initially increases linearly with density at low densities, with the gradient representing the free-flow velocity v. Upon reaching its maximum value $\hat{q} = vw\hat{\rho}/(v+w)$ (flow capacity), it then decreases linearly at higher densities. Here, the gradient corresponds to the backward wave speed w, the negative ratio of vehicle length to the safety gap in congested conditions.

In non-stationary state, the receiving and sending function are expressed as [78]

$$R(t) = \min\{\hat{\rho}L - UQ(t), \hat{q}\delta\} \tag{2.7}$$

$$S(t) = \min\{DQ(t), \hat{q}\delta\} \tag{2.8}$$

where L represents the link length, δ denotes the discrete time step, and UQ and DQ are the respective upstream and downstream queues. The vehicle queues are computed for each time step δ as follows

$$UQ(t) = UQ(t-\delta) + \delta\left[q^{\text{in}}(t-\delta) - q^{\text{out}}(t - L/|w|)\right] \tag{2.9}$$

$$DQ(t) = DQ(t-\delta) + \delta\left[q^{\text{in}}(t - L/v) - q^{\text{out}}(t-\delta)\right]. \tag{2.10}$$

In intuitive terms, all vehicles δq^{in} that have recently entered the link are placed in the upstream queue UQ, while those that left the link $L/|w|$ time units ago are now removed from UQ [78]. A similar approach is used for the downstream queue.

The queue model is typically subject to three constraints [100]: (i) Vehicles may enter a link i only when sufficient space is available, determined by the *storage capacity* parameter N_i. This parameter represents the maximum number of vehicles allowed to queue up on a link, calculated as the ratio of link length L_i and vehicle length l_{veh} multiplied by the number of lanes $n_{i,\text{lanes}}$ on that link. Additionally, vehicles may leave a link only (ii) once their free-speed travel time T_0 has passed. This travel time is calculated as the ratio of link length L_i and speed limit $v_{i,0}$, disregarding the vehicle's maximum speed capability at this point. Lastly, vehicles may leave a link (iii) only if the link's outflow capacity, represented by the *flow capacity* parameter Q_i, has not been exceeded. This parameter simulates a throttling mechanism, allowing vehicles to pass only at intervals of $\frac{3600}{Q_i} \times n_{i,\text{lanes}}$ seconds.

As discussed in Sec. 2.2.1, MATSim facilitates *sample runs* to enhance computational efficiency by simulating only a subset of agents. For example, in a 10% run, each simulated vehicle is weighted as ten cars, occupying a net space of 75 m on the network (instead of 7.5, m in a 100% sample scenario). To maintain traffic dynamics, the *flow* and *storage capacities* Q and N are adjusted accordingly and multiplied by a factor $f_{f,s} = 0.1$ [209, 98][14]. Additionally, to

[14] Exceptions to this rule and their necessity are discussed in Sec. 2.6.4.

enable heterogeneous traffic modeling, MATSim's queue sorting order can be changed from the *first-in-first-out principle (FIFO)* to a *passing mode*, where all vehicles are sorted by their earliest link exit time [2].

2.5.2 Implications for the deduction of realistic driving profiles

MATSim inherently allows for the deduction of vehicle trajectories and status profiles. Every agent's action, such as entering or leaving a road segment (*link*), is recorded. This data enables the easy derivation of daily status and speed profiles, as illustrated in Figure 2.11. However, since MATSim employs a simplified queue model to approximate traffic dynamics, it does not provide precise information on a vehicle's position within a link. Only average link speeds can be extracted, which are insufficient for automotive requirement engineering. More realistic speed profiles can be obtained through post-processing and filtering techniques (e.g., as proposed in [205]). However, the accuracy of these speed profiles depends heavily on MATSim's ability to model congestion patterns accurately, such as spillback effects from queuing vehicles in upstream links.

Unfortunately, the queue model has limitations in congestion modeling because MATSim prioritizes handling large scenarios efficiently. As a result, traffic and driving dynamics are simplified [9]. This may lead to inaccurate congestion patterns and distorted vehicle dynamics, particularly on short links or in sample runs.

The flow capacity functions like a batch system: a capacity of 600 cars/h means a vehicle can leave a link only every six seconds, causing newly arriving vehicles to queue. This can result in unrealistically long passing times. For example, two subsequent vehicles on a 15 m link would require at least 6 seconds for the rear car to pass, even with a free flow velocity of 50 km/h, as the exit is blocked by the first vehicle.

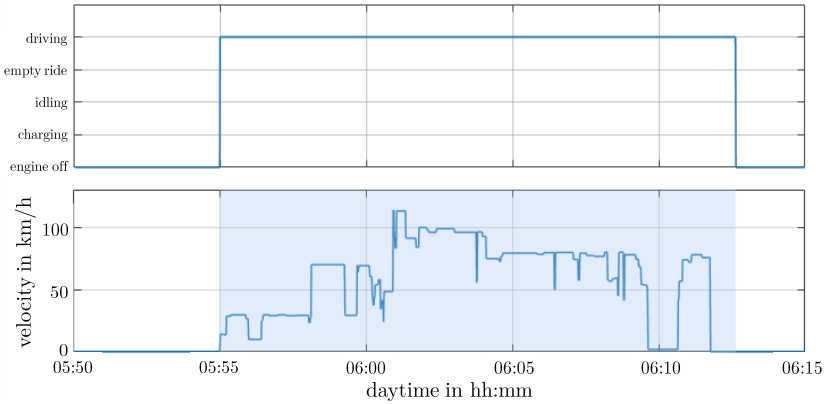

Figure 2.11: Status (top) and average link-speed profile (bottom) for a selected MATSim car, derived from the MATSim event file. Relevant vehicle events (e.g., `LinkEnterEvent`, `LinkLeaveEvent`, and other DRT-specific events) are captured using EventHandler interfaces [239]. Post-processing of this information generates status and speed profiles for the chosen agents.

The *stucktime parameter*[15] exacerbates this issue by temporarily violating the downstream link's storage constraint, allowing a surplus of vehicles on a link. In 10% sample runs, vehicle queues can extend to 300 m on a 10 m link, with the second vehicle taking at least 1 minute to pass, the third 2 minutes, and the fourth 3 minutes[16]. Technically, in 1% sample runs, it is possible to enforce four vehicles, each with a weight of 100, to occupy a short link without triggering an error. This results in a total queue length of 3 km, calculated as 4 cars x 100 weight x 7.5 m vehicle length. It is important to note that under these conditions, queues do *not* extend back onto upstream links, preventing MATSim from accurately modeling spatial congestion patterns, even if the flow capacity method is theoretically sound[17].

[15] To counteract gridlocks, the *stucktime parameter* bypasses the storage capacity constraint if the first vehicle in the queue is stuck for too long, ensuring minimal flow even in very congested conditions [98].

[16] Given a nominal flow capacity of 600 cars/h, corresponding to 60 cars/h in a 10% sample run.

[17] We emphasize at this point that MATSim's core functionalities remain unaffected by these artifacts, as its primary aim is to capture realistic trip-wise travel times, not the precise spatio-temporal positions of vehicles.

As a consequence, short links act as temporary vehicle sinks, storing excess vehicles that would otherwise cause spillback on upstream links. Consequently, average link-speed profiles become inaccurate under congested conditions, often showing near-zero speeds on short links and nearly free-flow speeds on the preceding links. Potential efforts to enhance the realism of these average link speed profiles through post-processing must address this issue. For example, applying filtering techniques with larger 'window sizes' can smooth out these artificial congestions, both spatially and temporally, thereby mitigating their impact. However, caution is warranted, as the objective is not to eliminate all extreme driving events, such as stop-and-go patterns.

Before delving into techniques for obtaining more detailed driving dynamics in Chapter 3, we first take a necessary detour to assess the uncertainties in MATSim DRT simulation with respect to SAEV mode acceptance and fleet driving patterns. This evaluation will help us determine whether a MATSim-based approach is suitable for our overall endeavor of SAEV drive cycle deduction. Relying on MATSim in this context offers the advantage that our key requirements KR0, KR1, KR2, and KR4 (see Section 1.4) are inherently met by its core functionalities.

2.6 Assessing uncertainties in MATSim DRT simulation

The MATSim Open Berlin Scenario [233, 234] provides a robust foundation for deriving reliable SAEV driving cycles. Its agent-based framework is highly suitable for exploring new travel modes within scenario-based contexts and deriving the corresponding travel demand. Following thorough calibration, as delineated by [234], the Open Berlin Scenario effectively models the complexities of Berlin's current transport system, making it an optimal starting point for implementing the DRT travel mode.

However, it is essential to recognize the inherent limitations of any model in fully capturing the complexity of reality, despite efforts to achieve realism. Additionally, the lack of real-world data for SAEV fleets hinders the validation of DRT models against actual fleet performance, increasing modeling uncertainties.

Acknowledging these factors, this section evaluates the inherent uncertainties in MATSim DRT simulation related to SAEV mode acceptance and average driving patterns. Our investigation focuses on three key aspects: (i) identifying global uncertainties in SAEV travel demand modeling, (ii) analyzing output sensitivity to variations in DRT-specific input parameters, and (iii) assessing the robustness of average SAEV driving patterns in response to increased population rates simulated in MATSim (cf. sample shares in Sec. 2.2.1) and variations in random seed in the steady state (cf. MATSim iterations in Sec. 2.2.2).

2.6.1 Global uncertainties related to SAEV modeling

As previously discussed in Chapter 1.2, significant socio-technological and regulatory uncertainties are associated with the introduction of SAEV fleets. While future fleet operators are prepared to assume certain risks, such as high acquisition costs, range anxiety, and charging inconvenience—factors that might have deterred individuals from purchasing EVs—the widespread market entry of SAEV fleets is also constrained by several AV-related issues that MATSim cannot fully address.

Notably, the absence of psychological models in the MATSim Open Berlin Scenario prevents the modeling of technology aversion concerning AVs. While MATSim theoretically permits the incorporation of sub-populations to address SAEV mode acceptance based on demographic factors (e.g., social background, education, gender, age, technical affinity), this capability is usually not applied due to parameterization challenges. Instead, MATSim implicitly accounts for user reservations and decision-making processes by adjusting the agents' overall utility, such as by imposing higher "costs" associated with SAEVs[18].

Unfortunately, this utility-based approach lacks consistency in mirroring real-world population mobility behavior, as MATSim operates on the assumption of rational agent behavior and grants agents omniscient knowledge of the mobility system. This enables them to anticipate all route-based travel times and frequently evade congestion by selecting more advantageous travel modes or routes [9]. To achieve a more realistic portrayal of flawed agent behavior, one could consider simulation runs that have not yet reached steady user equilibrium

[18] Compare DRT fares and costs in Tab. 2.2 with utilities of other modes in Tab. 2.1

or diminish agents' memory of beneficial plans. However, exploring these approaches is beyond the scope of this thesis.

Furthermore, while MATSim simulations can effectively model dedicated AV/EV lanes or city-wide bans on private cars to encourage SAEV mode acceptance, MATSim lacks the capability to model the enhanced traffic flow achieved through the utilization of advanced sensors and V2X communication by AVs. Increased AV velocities could approximate this scenario; however, in urban areas, a vehicle's speed is constrained by the road's legal speed limit rather than its maximum speed capability.

Another aspect not accounted for is the temporal variability in SAEV mode acceptance. This variability may arise from temporal fluctuations in the choice dimension (e.g., weather, seasonal effects, or global economic conditions) and/or the internal state of the decision-maker (depicting inherent motivations such as personal preference and experience) [99]. For instance, the inclination to choose biking or walking modes notably declines during cold and rainy conditions. Additionally, the usage of public transport reached historic lows during the COVID-19 pandemic due to increased health concerns, leading people to shift towards more individual transport modes like private cars [27]. Moreover, a person's mode choice may evolve over time based on factors such as household size or ecological beliefs. Although we did not address these effects in our study, we recommend consulting Horni et al. [99] for a deeper understanding of temporal choice variability in MATSim. For our automotive requirement engineering purposes, we considered this issue less relevant.

2.6.2 Sensitivity analysis with respect to fleet KPIs

To emphasize how drastically fleet KPIs can change depending on the underlying DRT scenario, we present Table 2.7, which exemplifies the MATSim simulation outputs for seven different scenarios (A-G). Each scenario is classified into one or more of four focal groups to assess the impact of (i) ride-pooling, (ii) DRT service area, (iii) private car penalties, and (iv) varying DRT fares on SAEV mode acceptance and fleet characteristics. Unlike the DRT scenarios P5-10af

and I-10af detailed in Sec. 2.2.2, we modify only one input parameter at a time to individually evaluate the impact of each variable.[19]

In automotive requirements engineering, it is crucial to acknowledge the wide variation in daily SAEV driving patterns. Table 2.7 illustrates that the average daily mileage (R7) and operation time (R8) can vary significantly, depending on the DRT mode's appeal to different user groups. This ranges from 20 kilometers in 3.7 hours per day to 120 kilometers over 9.4 hours per day. As demonstrated in Chapter 8 on power engines, these differences significantly impact the operational lifetime of the EV system and its components.

Simulation scenarios A and B in the first focal group assess the impact of ride-pooling without changes to the pricing scheme, unlike scenarios P5-10af and I-10af shown in Table 2.2. In scenario A, each DRT vehicle accommodates up to five passengers, while scenario B offers a non-pooled taxi service, transporting only individuals or groups from the same household. The DRT mode share significantly increases with this premium service due to reduced waiting times from the absence of detours, making it highly attractive as both services have similar fees. However, economically, the non-pooled service is less profitable. Despite similar empty mileage ratios, the revenue factor for the premium service is considerably lower, necessitating more vehicles to meet customer demand due to the lack of pooling efficiency.

The second focus is on the impact of the DRT service area, as demonstrated by scenarios C and D. Scenario C encompasses the entire ("full") city of Berlin, defined by the red state border shown in Fig. 2.3, while scenario D is confined to the "inner" city, delineated by the S-Bahn ring enclosed by lines S41/S42. Scenario C covers more suburban, less densely populated areas, while scenario D focuses on highly profitable areas with high trip density. Contrary to expectations, the high trip density in scenario D does not result in higher pooling efficiency, larger revenue factor, or lower empty factor. Despite significantly lower waiting times and in-vehicle travel times, scenario D is less profitable than scenario C.

This counterintuitive result arises from second-order effects: even with high customer service, the DRT travel mode is less attractive in scenario D, as

[19] For a comprehensive discussion on the impact of EV charging strategies on DRT performance, please refer to [134, 49, 229].

Table 2.7: Key performance indicators for DRT fleet scenarios (A-G) with varying DRT-specific inputs. Each scenario is categorized into one or more of four groups to assess the impact of (i) ride-pooling, (ii) DRT service area, (iii) private car penalty, and (iv) DRT fare. Performance metrics are divided into fleet-related (F1-6), vehicle-based (R7-10), and service-related (S11-12) categories. "Full" city refers to all of Berlin within the state border, while "inner" city denotes the central area within the S-Bahn ring lines S41/S42. Revenue and empty factor denote the ratio of passenger mileage to total vehicle mileage and the ratio of empty mileage to total vehicle mileage, respectively. All simulations use a 1% MATSim sample population, and, for clarity, fleet-related results (F2-4) are not extrapolated to 100%. All simulations are based on the MATSim Open Berlin Scenario [233, 234], supplemented with a DRT travel mode.

		– DRT fleet scenario –						
		A	B	C	D	E	F	G
focal group	(i) impact ride-pooling factor	x	x					
	(ii) impact DRT service area			x	x			
	(iii) impact private car penalty			x		x	x	
	(iv) impact DRT fares	x		x				x
DRT-specific inputs	service area	full	full	full	inner	full	full	full
	pooling capacity	5	1	5	5	5	5	5
	price (EUR/km)	0.18	0.18	0.18	0.18	0.18	0.18	0.10
	base price (EUR)	0	0	1	1	1	1	0
	min price (EUR)	1	1	0	0	0	0	1
	constant daily car costs (EUR)	-10.6	-10.6	-10.6	-10.6	-5.3	-26.5	-10.6
	car costs (EUR/km)	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
DRT performance metrics	F1 DRT mode share (%)	6.1	17	3.7	0.8	3.1	4.2	5.6
	F2 # DRT vehicles	764	1988	579	209	521	676	695
	F3 # DRT trips	10502	30362	6186	1196	4911	6972	9446
	F4 daily DRT mileage (km)	70545	245332	45099	4238	31993	51463	82723
	F5 empty factor	0.08	0.08	0.09	0.12	0.12	0.08	0.07
	F6 revenue factor	1.21	0.92	1.13	0.98	1.01	1.16	1.3
	R7 ∅ mileage (km/car)	92.3	123.4	77.9	20.3	61.4	76.1	119
	R8 ∅ operation time (h/car)	7.4	4.5	6.5	3.7	6	6.5	9.4
	R9 ∅ # movements (1/car)	24.4	26.4	18.7	9.7	16.3	17.9	24.6
	R10 ∅ velocity per taxi (km/h)	22.5	20.7	23.9	19	22.4	24	23.3
	S11 ∅ customer waiting time (min)	7.4	4.5	6.5	3.7	6	6.5	9.4
	S12 ∅ in-vehicle travel time (min)	21.6	21.1	19.9	11.3	17	20.7	28.4

evidenced by smaller mode shares. We infer that the average trip lengths in the downtown area are too short, making the base price relatively expensive and thus rendering the DRT mode less attractive. To analyze the impact of the base fare independently of the DRT service area, scenarios A and C—both covering the entire city but differing in base and minimum fares—are examined. The base price entails a fixed fee before the first kilometer is driven, while the minimum price is charged at the end of the trip if the kilometer-based fee is below this threshold (otherwise, the kilometer-based fare applies). Results from both scenarios indicate that the minimum fee imposes less financial strain on customers, making the travel mode more attractive.

The third focal group examines the impact of private car penalization on DRT mode acceptance and usage (simulations C,E,F). As expected, when private car usage is more expensive (scenario F), the DRT mode share increases significantly. Conversely, when private car usage is not strongly penalized (scenario E), the DRT service becomes less attractive in comparison. This low mode adoption leads to fewer pooling opportunities, resulting in high empty factors and substantially lower revenue values.

The fourth focus evaluates the impact of DRT pricing models (simulations A, C, and G) on fleet use pattern. Generally, high prices per kilometer combined with a minimal base price (scenario A) favor medium trip lengths. Low prices per kilometer make long-distance trips more appealing (scenario G); however, these lead to longer operation times and consequently longer customer waiting times. Interestingly, despite the seemingly inconvenient scenario for customers, this model yields the most profitable DRT fleet configuration. Due to the relatively cheap ride service, customers seem willing to accept longer ride detours resulting from pooling. The high pooling efficiency on longer trips leads to larger revenue factors. In contrast, a high price per kilometer combined with a base fee (simulation C) favors short-distance trips.

2.6.3 Output variability due to different random seeds

Transport simulation tools are essential for infrastructure decision-making and investments. For these tools to be reliable, their outputs must be robust to stochastic effects; otherwise, the variability of results can overshadow the measures being considered [177]. MATSim, like other agent-based models

that rely heavily on discrete choice models, is inherently non-deterministic.[20] It frequently uses random numbers, for example, to determine which agent undergoes *innovation* by selecting, cloning, and mutating their daily plans [99]. The sequence of random numbers is determined by a seed, so initializing a pseudo-random number generator with the same seed will produce the same sequence.[21] Due to this stochastic nature, running a MATSim scenario N times with identical parameters but varying random seeds will produce N independent simulation outputs [78].

Stochastic effects are intentionally implemented in many simulation frameworks to represent model uncertainties. Many phenomena in mobility and transportation systems are not fully understood [78]. Thus, stochastic model outputs and their variability offer a more accurate representation of real-world processes and their inherent uncertainties. To make sense of these outputs, multiple seed trials should be evaluated using statistical techniques such as the arithmetic mean or empirical variance [78]. However, large-scale micro-simulations are computationally expensive, so many studies rely on a single run. This is often due to the fact that decisions based on a single run are still better than those made without any simulation information [99]. Some studies have investigated the random seed variability of MATSim simulations based on different metropolitan scenarios [177, 31, 98, 195].

In [177], random seed variability was analyzed at the link level for Santiago de Chile using a 10% sample. The authors performed 100 seed trials with 100 MATSim iterations each[22]. A key finding was that the relative accuracy of estimated link volumes generally increases with link load, although deviations of up to 10% were observed even for high-traffic links. The share of links with high variance in daily load remained constant across trials. Some busy links showed high error-proneness, which is troublesome as busy links are often the focus of infrastructural measures, highlighting the need for further geographical

[20] While [147] claims that MATSim is deterministic because its outputs can be reproduced using a fixed random seed, we argue that reproducibility does not necessarily imply determinism. A model can be stochastic yet still produce reproducible outcomes when randomness is controlled.

[21] However, this does not render the model deterministic; it simply ensures that the random processes within the model produce consistent results across runs. By our definition, deterministic models are characterized by the complete absence of randomness, even when controlled.

[22] The authors suggest that 100 iterations might not suffice to achieve a stable user-equilibrium, possibly contributing to output variability.

analysis. Additionally, seed-related variation in link volumes exceeded the variability observed in subsequent user-equilibrium iterations within the same seed. While mode share variability was not an issue, travel speed variability was similar to individual link load variability.

In [31], the uncertainty of travel times was investigated in a case study of Hanover, Germany (10% sample). They recommended a technique for estimating the number of simulation runs needed for stable results. Sixteen different seed trials with 750 MATSim iterations each were performed. They found minimal variance in global average travel time but significant variability in individual travel times due to the agents' large decision space involving time, route, mode, and destination choice [99]. District-level uncertainties were moderate as agent-level variabilities compensated at aggregated levels, showing that aggregation reduces variability, as empirically shown in [195] and formally proven in [98].

[98] further demonstrated, using a Zurich 10% test case, that average variability in agents' executed plan utilities was approximately 3% at the person level, while population-level variability was minimal (0.087%). Nonetheless, [99] cautions against relying solely on aggregation to control sampling error, noting that spatial aggregation of urban and rural areas may not be meaningful. Both [31] and [99] agree that higher aggregation levels require fewer runs.

Interestingly, [31] found that 6 out of 16 seed runs had identical travel time distributions, suggesting that simulation output may settle into discrete states for each agent. For example, nearly 22% of private cars exhibited a single state where travel time was constant across runs. This finding is intriguing given that an agent's travel time is influenced by the travel behavior of other agents, and the reasons for these discrete states are not yet fully understood.

Unfortunately, there is limited literature on the seed variability of ride-hailing services. To address this, we conducted two seed trials with 1% and 10% sample populations. The results, shown in Figures 2.12 and 2.13, indicate that lower sample shares lead to higher seed-related output variability. For the 1% seed trials (Figure 2.12), the second run shows an 8.2% higher DRT mode acceptance, resulting in a 29.3% increase in fleet vehicles. This higher travel demand improves pooling efficiency, reducing empty factors and waiting times. However, the significant mode share difference suggests the simulation may not have reached a stable user-equilibrium despite 500 iterations. In contrast, the 10% seed trials (Figure 2.13) show that all fleet-related, and thus

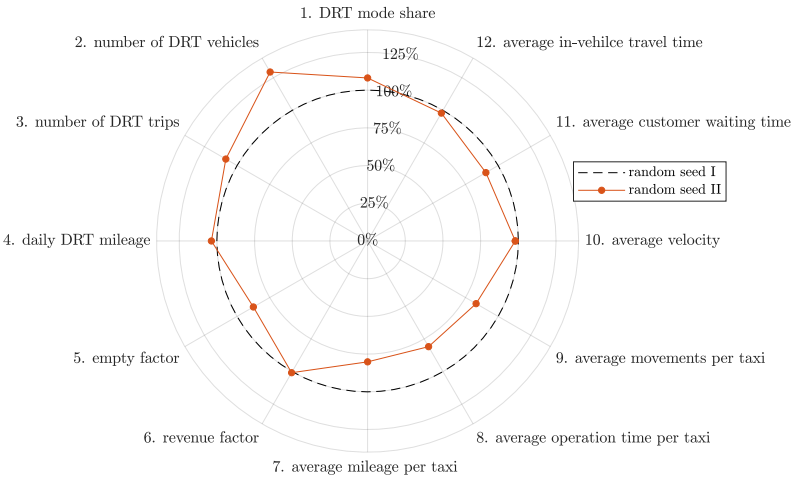

Figure 2.12: Relative difference between two seed trials with a 1% sample run. Fleet KPIs from the first run are normalized to 100% (black dashed), while KPIs from the second run are shown in relative terms (red solid).

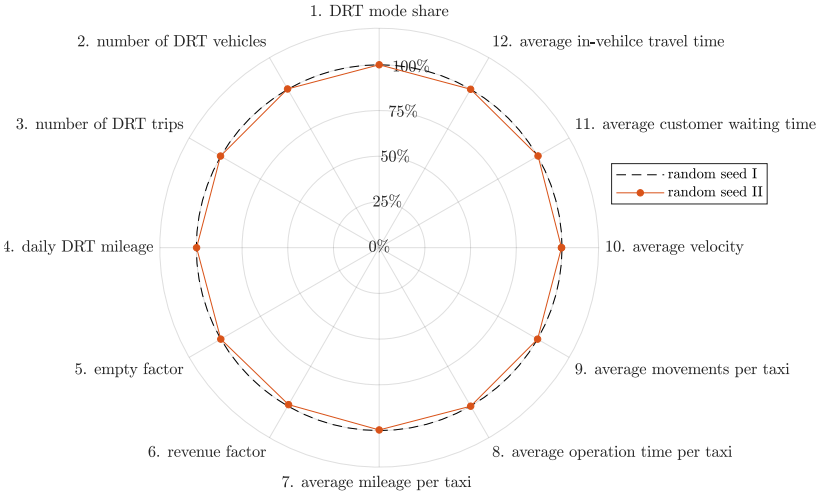

Figure 2.13: Relative difference between two seed trials with a 10% sample run. Fleet KPIs from the first run are normalized to 100% (black dashed), while KPIs from the second run are shown in relative terms (red solid).

aggregated, KPIs vary by less than 1%, indicating that seed-related uncertainties are negligible for sample populations of 10% or larger.

However, vehicle-specific travel times exhibit some variability across different seed trials with 10% sample populations. While our number of seed simulations is insufficient to draw statistically significant conclusions, we estimate deviations of up to ±10%, consistent with findings in [177]. This variability in individual travel time is acceptable, as it reflects the variations typically observed in real-world driving. To support this, we refer to the bottom panel of Figure 7.14, which shows that individual MATSim travel times generally fall within the range of Google Navigation estimates. In fact, one might even argue that MATSim's travel time variability is too conservative[23], as Google's estimates vary by up to ±40% at night and ±50% during the day.

2.6.4 Distorting effects of MATSim sample shares

Agent-based transport simulation is particularly demanding in terms of computing time and memory consumption. To perform large-scale simulations efficiently, it is common practice to simulate only a randomly selected subset of agents. For example, in a MATSim 10% sample run, each agent represents ten agents, and network capacities are proportionally reduced to preserve traffic dynamics.

However, proportional downscaling in small sample runs (<10%) leads to inconsistencies in travel time and distance for private cars [133], affects the pooling efficiency and fleet size of ride-hailing services [111], and creates other artifacts that distort simulation results and driving dynamics [66, 207]. To minimize these effects, MATSim network capacities are occasionally adjusted non-linearly to achieve more realistic representations of traveled distance and time distributions [169, 238]. For instance, a heuristic factor γ is introduced in [169] to describe the ratio of flow and storage capacity for a given sample share $x \in]0, 1]$

$$\gamma = \sqrt[4]{x} = \frac{f_\mathrm{f}}{f_\mathrm{s}}. \tag{2.11}$$

[23] or that agents are overly efficient at avoiding congestion

With a flow capacity factor of $f_f = x$, the storage capacity factor f_s can be expressed as

$$f_s = x^{\alpha} \tag{2.12}$$

where the exponent α equals 0.75 [133]. Thus, for a 10% sample run, instead of reducing both flow and storage capacity factors to 0.1, the storage capacity factor is set to $f_s = 0.59$. The efficacy of the exponent α is questionable, however. In [133], the exponent had little impact on average travel time across various sample runs $x = \{0.01/0.05/0.1/0.2/0.5/1\}$. Average travel time for private cars depends heavily on the sample share x, forming a U-shaped curve with minimum travel time at around $x = 0.1 - 0.2$. The authors question the reliability of small sample runs, as their results deviate from the general trend of reduced average travel times with higher MATSim iterations across all sample shares.

The insufficient validity of small sample runs is also confirmed by [66], who note that population sampling is challenging when agents belong to the same social community (e.g., households). Instead of randomly drawing agents, samples should be drawn at the community level to ensure all or none of its members are included. Small sample shares are acceptable only for highly aggregated data, but detailed traffic flow analysis requires higher sample shares [133], especially for infrastructural decision-making. Additionally, small sample runs are susceptible to outliers, potentially skewing overall results due to the small sample size, leading to errors in aggregated KPIs [31].

Similar to the household issue, representing ride-pooling services is challenging for small sample runs due to discretization errors [98]. For instance, in a 10% sample run with a vehicle capacity of 4, a taxi picking up a second customer effectively carries 20 people. This requires identical departure and destination locations, making such events less frequent than in a 100% sample where each agent represents one person. Consequently, ride-pooling is less effective in small sample runs, as fewer ride-pooling requests lead to smaller, less flexible taxi fleets, resulting in longer waiting times and less appealing services. This is confirmed by [111], who studied the impact of sample shares on fleet size and pooling rate of ride-hailing services in two German regions: (a) the eastern inner-city of Berlin with sample shares of $x = \{0.05, 0.1, 0.15, 0.2, 0.25\}$ and (b) the rural district of the Vulkanic Eifel with $x = \{0.1, 0.2, ..., 0.9, 1\}$.

They found a non-linear relationship between trip density (thus, sample share x) and ride-hailing parameters. Contrary to naive assumptions, fleet size and other KPIs cannot be proportionally up-scaled by a factor of x^{-1} to obtain 100% sample results, as this leads to overestimations of DRT fleet size, operating hours, and vehicle kilometers. The authors provide corrected up-scaling factors based on their experiments, as shown in Table 2.8. For instance, to normalize the fleet size of a 10% sample to 100%, the correct up-scaling factor is not 10 but 4.34, as a non-linear factor of $x^{-0.637}$ is applied instead.

Table 2.8: Adjusted scaling factors for DRT fleet size, operation time, and vehicle kilometers in Berlin's eastern inner-city, derived from [111] as functions of sample size $x \in]0, 1]$ using non-linear regression.

	corrected scaling factor	coefficient of determination	factors 1%	factors 10%
fleet size	$f_{sz} = x^{-0.637}$	$R^2 = 0.9954$	18.79	4.34
operation hours	$f_{op} = x^{-0.662}$	$R^2 = 0.9975$	21.08	4.59
vehicle-kilometers	$f_{km} = x^{-0.928}$	$R^2 = 0.9999$	71.80	8.47

The authors also observed that larger sample shares enhance the pooling effect and increase the revenue factor (ratio of revenue distance to total distance). A revenue factor of 1 means pooling compensates for additional empty mileage from pick-up trips. For the eastern Berlin test case, the revenue factors for $x = \{0.05, 0.1, 0.15, 0.2, 0.25\}$ samples are approximately 1.16, 1.22, 1.25, 1.28, and 1.31, respectively.

The findings from [111] are highly relevant to our work, but the up-scaling factors provided cannot be directly applied to our scenario. Using the corrected fleet size scaling factor $x^{-0.637}$ from Table 2.8 for our 1% and 10% sample runs of the Open Berlin Scenario did not yield identical results as would be expected if the factor accurately enabled the calculation of 100% sample values.[24] This discrepancy is likely due to two factors. First, the test case setups are too different to assume identical up-scaling factors, even though both focus on Berlin, as shown in Table 2.9.

[24] With well-fitting regression models, typically evaluated by a high coefficient of determination ($R^2 \approx 1$), the 100% fleet size ($F_1 = F_x \cdot x^{-0.637}$) should remain relatively stable across all sample shares ($x \in]0, 1]$).

Table 2.9: Model differences between our Berlin test case and the one analyzed in [111].

	Open Berlin Scenario (own DRT configuration)	Berlin Scenario with DRT by [111]
Demand modeling	land-use & survey data	survey & mobile phone data
Service area	Berlin full city	eastern Berlin
Pooling capacity	5	4
Service quality	max. waiting time of 20 min	90% of all waiting times below 10 min
DRT fares & utilities	cf. scenario A in Table 2.7	not provided
Fleet size limit	pre-set	dynamically

Second, the uncertainties in 1% sample runs, as evidenced in Figure 2.12 (with deviations in fleet size reaching up to 30%), are too large for effective use in regression modeling. [111] also deliberately excluded small sample sizes from regression analysis. Curve fitting yields more accurate results with larger sample sizes ($x \geq 0.05$), as the associated up-scaling factors (x^k) converge towards one when approaching 100% samples

$$\lim_{x \to 1} f(x) = \lim_{x \to 1} x^k = 1. \tag{2.13}$$

Based on the specific behavior described in Equation 2.13, scenario-specific upscaling factors (x^k) can be approximated for selected ride-hailing attributes $F = \{\text{fleet size, operation time, vehicle kilometers}\}$ using two arbitrary sample runs x_1 and x_2 :

$$k = \frac{\log\left(\frac{F_{x_1}}{F_{x_2}}\right)}{\log\left(\frac{x_2}{x_1}\right)}. \tag{2.14}$$

The corresponding 100% sample results F_1 can then be determined using:

$$F_1 = F_{x_1} \cdot x_1^k. \tag{2.15}$$

However, our exclusive reliance on 1% and 10% sample runs precludes us from applying this approach effectively, as 1% runs lack sufficient reliability. Unfortunately, we lack the resources to generate higher sample shares and derive corrected rescaling factors independently. Unlike [111], where travel demand was synthesized from anonymized mobile phone data [166], our demand generation relies solely on land-use and survey data. This method is time-consuming and typically results in smaller sample shares (we only have 1% and 10% sample demands) compared to the more automated mobile data approach, which offers greater sample coverage.

In conclusion, sample shares significantly impact DRT key performance indicators. Proportional up-scaling (x^{-1}) leads to substantial overestimation of KPIs such as fleet size, vehicle operation time, and mileage. Without accurate up-scaling factors, we must rely on those suggested by [111] or accept these uncertainties. Further research is warranted.

2.6.5 Uncertainties put into perspective

Modeling the driving behavior of SAEV fleets is inherently subject to many uncertainties. While MATSim effectively captures the impacts of different pricing schemes, dispatching strategies, and interactions with other travel modes on average driving patterns, the wide range of relevant modeling levers means driving characteristics can vary significantly. This variability complicates DRT predictions and allows for considerable flexibility in simulation outcomes. Given the broad range of possible future DRT scenarios[25], it is essential to acknowledge this variability, especially when using data for prototyping. Balancing cost-saving potentials through optimized component sizing with the need for safety margins in fatigue analyses requires careful consideration.

In addition to the variability in daily driving behavior shown in Table 2.7, scenario uncertainties can reach 10-30% due to the stochastic nature of 1% sample runs influenced by different random seeds. Consequently, simulation results should be seen as guiding principles rather than fixed outcomes. To

[25] e.g., DRT as a transit feeder connecting passengers to main transit hubs, or a city-wide private car ban where a DRT fleet replaces all private vehicles

enhance output stability, future MATSim simulations should use sample populations of at least 10

Caution is also needed when extrapolating sampled scenarios to 100% populations. As shown in Table 2.8, the up-scaling of fleet-related KPI does not follow the reciprocal of the sample size x. Although we lack precise scaling factors for our simulations, we can infer that the average daily mileage per vehicle (ratio of total vehicle-km to fleet size) is also influenced by this new insight. This is because fleet size and total vehicle kilometers have distinct scaling factors (f_{sz} and f_{km})[26]. According to [109], the average km per car should be multiplied by $\frac{f_{km}}{f_{sz}} = \frac{8.47}{4.34} = 1.9$ for a 10% sample scenario to represent a 100% population sample. This significantly increases component strain and must be carefully considered during prototyping. By contrast, the average operation time per vehicle remains nearly constant, as the adjusted re-scaling factors are similar $\frac{f_{op}}{f_{sz}} \approx 1$.

[26] Previously, both f_{sz} and f_{km} were identical, making the up-scaling of vehicle-centered metrics unnecessary, as their ratio amounted to one.

3 Advanced methods for detailed driving dynamics

In this chapter, we develop two distinct approaches to overcome MATSim's limitations in modeling detailed driving dynamics, aiming to derive more realistic velocity profiles (KR3). First, we seek to enhance MATSim's average link-speed profiles with real-world data (Sec. 3.1). Second, we explore microscopic traffic simulation with SUMO (Sec. 3.2), offering a model-based solution to this challenge, thereby reducing the need for resource-intensive on-road measurements. Both approaches are compared in a test case study (Sec. 3.3), with their implications for the thesis scope also discussed.

3.1 Integrating real-world measurements to enhance MATSim drive cycles

As a first attempt to obtain more realistic driving dynamics, we develop a post-processing procedure that replaces MATSim's simplified velocity profiles with real-world driving data in a piece-wise manner. This method represents an updated and expanded version of our enrichment approach published in [209]. Before detailing the method, we first examine the underlying real-world driving data.

3.1.1 Driving cycle data set description

Real-world measurement campaign The quality of enriched driving cycles is directly tied to the range of underlying on-road measurements [209]. Therefore, we require a substantial number of real-world measurements encompassing various driving behaviors, road types, and congestion levels for

genuine representation. However, driving cycles with severely congested traffic are particularly scarce, as they are less useful for conventional dynamometer testing and are seldom the focus of measurement campaigns.

The real-world driving cycles used in this thesis originate from a joint research project[1] between the *Research Institute of Automotive Engineering and Vehicle Engines Stuttgart* (FKFS) and *Robert Bosch GmbH* in the Stuttgart region. In this project, a Citroën C-Zero was equipped with multiple sensors to quantify the energetic impact of various car components under varying ambient conditions. Several metrics were redundantly recorded to limit the risk of signal loss and sensor failure. For example, the vehicle motion was captured using two independent GPS trackers, a contact-free optical sensor, and a longitudinal acceleration sensor, all at a rate of 50 Hz. Differences in traveled distance between all four measurement devices were found to be marginal (<3%). Due to frequent GPS signal dropouts in the tunnel-rich Stuttgart region, speed profiles were mainly derived from the acceleration sensor[2]. In total, this source provided us with 41 hours of raw data, covering a total driving distance of 1643 km over various terrains. For more details on the vehicle routes, please refer to Figure 3.1.

Data cleansing and preprocessing techniques Measured data typically contains systematic and random errors, so we implement multiple data-cleansing measures on the recorded velocity profiles. Once these profiles are sufficiently conditioned, we derive related distance-, acceleration-, and jerk-time series through integration and differentiation, respectively. This ensures that cleaning measures are consistently reflected in all metrics.

First, we omit sampling points where the car idles for prolonged periods, typically at the start and end of measurement runs, without affecting actual waiting periods (e.g., at traffic lights). Second, we scan the velocity-time series and its derivatives for duplicate entries and missing data that cause unrealistic driving dynamics, such as excessively high acceleration rates (Figure 3.2, upper panel). When the acceleration sensor loses track of the current motion (e.g.,

[1] "Bordnetzmessungen am Elektrofahrzeug" (C-Zero), 2013

[2] Although these sensors primarily measure acceleration using a mass-spring system, they can also provide velocity metrics by integrating the acceleration data over time.

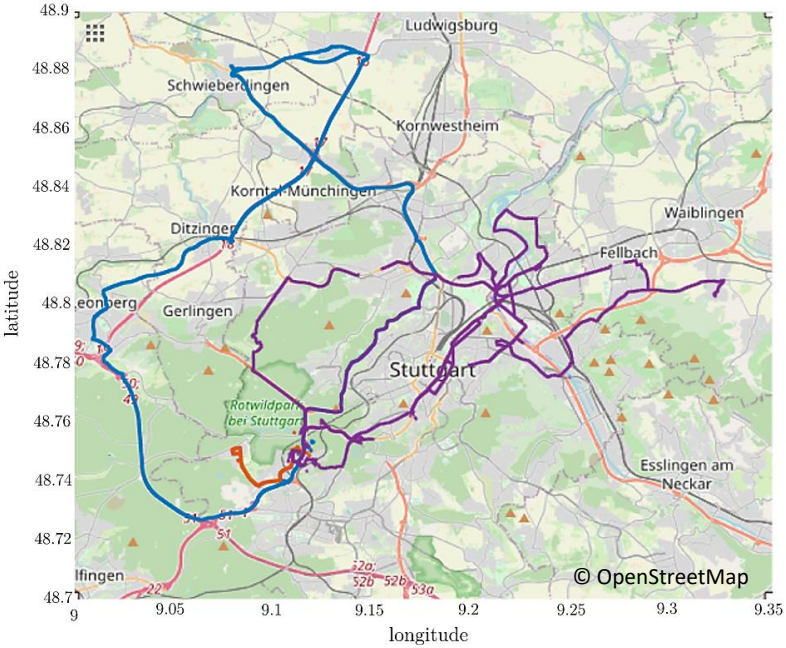

Figure 3.1: Routes of measured driving cycles from an aerial perspective. The *bluish line* represents the "FKFS-Rundkurs" data set with 23 independent cycles, featuring urban, suburban, and motorway segments over a 60 km track, typically taking 65 to 95 minutes. This circuit was designed by the FKFS Institute of the University of Stuttgart to serve as a representative driving cycle for the Stuttgart region. The *reddish line* shows a minor urban cycle of 9.1 km and 17 to 21 minutes duration, driven 32 times for a total of 286 km. The *purple data set* includes mixed road character cycles with no predetermined route, covering 228 km over 12 hours from 9 different cycles.

due to stark shocks like potholes), the speed values typically default to zero (Figure 3.2, lower panel). These false zero-speed segments are problematic as they imply misleading EV charging opportunities and cool-down times for power electronics and thermal components [69]. To mitigate this issue, we identify unrealistic acceleration spikes and employ cubic spline interpolation to enhance signal continuity. A significant challenge here is to replace erroneous data while minimizing the total amount of data altered. Therefore, to avoid

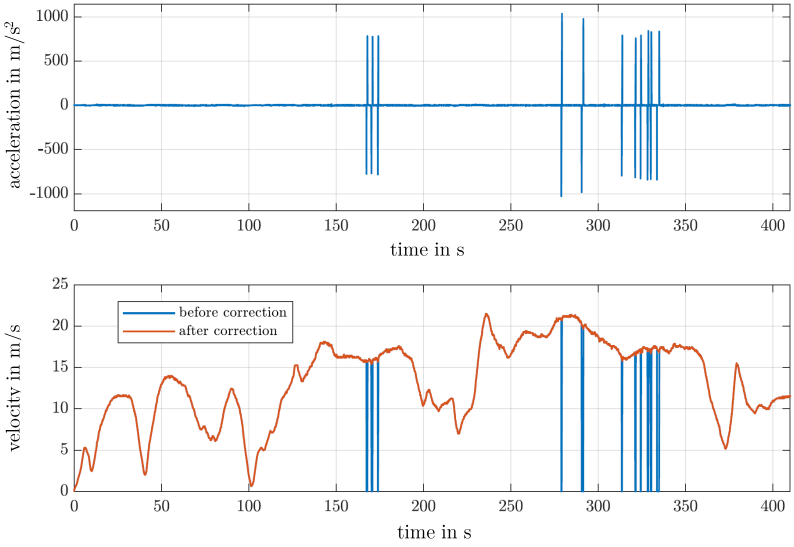

Figure 3.2: Acceleration spikes (top panel) caused by tracking dysfunction of the acceleration sensor. Velocity values erroneously set to zero (bluish line in bottom panel) lead to acceleration rates tending towards infinity. Corrective measures involve identifying affected velocity segments and replacing them with cubic spline interpolation (reddish line in bottom panel).

using heavily altered driving cycles in subsequent enrichment procedures, we flag interpolated segments in the time series accordingly.

Thirdly, we address signal denoising to reduce inaccuracies of power demand estimation in vehicle simulation, akin to challenges encountered in dynamometer testing [69]. Highly variable speed profiles can cause errors in state of charge (SOC) and energy consumption calculations. Commonly addressed using binomial smoothing or Savitzky-Golay filters (e.g., [146, 189]), we reduce the sampling rate of velocity profiles from 50 Hz to 1 Hz by averaging 50 sample points, as we do not require data resolution greater than 1 Hz. This adjustment effectively mitigates white noise while maintaining computational efficiency.

As expected, the rate of data conditioning markedly affects driving metrics, as shown in Table 3.1. This table compares driving characteristics of four selected drive cycles (I-IV) in their raw and conditioned states across varying

Table 3.1: Comparison of driving metrics between raw and conditioned velocity profiles at various interpolation ratios η_{int}. Symbols $\hat{x}$, $\overline{x}$, and $\tilde{x}$ denote maximum, mean, and median velocity and acceleration values, respectively. Metrics further include total distance traveled d_{tot}, standard deviation σ_{x}, idling ratio η_{idle}, and distance-normalized number of stops N. Deviation exceeding 30% and 60% from the unprocessed raw data is highlighted in yellow and red, respectively, to indicate significant deviations.

		η_{int}	d_{tot}	$\hat{v}$	$\overline{v}$	$\tilde{v}$	σ_{v}	$\hat{a}+$	$\hat{a}-$	σ_{a}	η_{idle}	N
		%	km	km/h	km/h	km/h	km/h	m/s^2	m/s^2	m/s^2	%	1/km
I	raw	-	59.1	108.0	55.3	56.6	14.3	0.43	-0.49	1.19	4.9	0.46
	filtered	3.0	60.5	108.0	56.6	57.0	13.5	0.30	-0.34	0.49	2.8	0.23
II	raw	-	7.8	63.2	21.1	18.7	10.6	1.29	-1.31	2.16	34.4	3.95
	filtered	9.6	9.1	63.2	24.5	25.4	10.8	0.40	-0.44	0.56	29.9	0.77
III	raw	-	6.7	60.1	17.9	10.7	10.2	1.21	-1.54	2.18	41.8	5.11
	filtered	14.7	9.0	60.1	24.1	26.7	10.8	0.39	-0.48	0.53	31.0	0.78
IV	raw	-	13.3	85.0	15.1	1.2	11.5	1.40	-1.31	2.04	47.0	9.27
	filtered	25.8	22.6	85.0	25.7	23.4	12.7	0.38	-0.39	0.52	26.9	1.55

interpolation ratios (η_{int})[3]. Notably, higher interpolation rates have minimal effect on maximum velocity ($\hat{v}$) and velocity standard deviation (σ_{v}), but significantly alter mean and median velocity ($\overline{v}$ and $\tilde{v}$). Acceleration metrics, such as maximum positive and negative accelerations ($\hat{a}+$, $\hat{a}-$) and acceleration standard deviation (σ_{a}), show more pronounced differences due to the removal of artificial acceleration spikes. Significant deviations are also observed in the vehicle's idling ratio (η_{idle})[4] and its number of stops (N) per kilometer. Both metrics are significantly reduced by data conditioning, greatly influencing EV energy consumption and charging behavior.

Overall, gaps in signal acquisition have a greater impact on driving metrics than initially anticipated, emphasizing the necessity and potential risks of over-processing data during conditioning. Extensive data conditioning alters natural driving behavior, making heavily modified driving cycles inappropriate for our cycle enhancement goals. Consequently, we discard all cycle segments that, when segmented into one-minute intervals, exhibit an interpolation ratio

[3] η_{int}: The proportion of sampling points that have been altered within a time series.
[4] η_{idle}: The proportion of time the vehicle remains stationary with the engine operating, despite not moving.

$\eta_{\text{int}} > 20\%$. As a result, from the initial 42 hours of raw driving cycle data (approximately 2,520 one-minute intervals or 'mini-cycles'), 2,188 remained after data cleansing and preconditioning. Of these, 1,954 valid mini-cycles met the interpolation criteria and were used for enrichment purposes.

This minute-wise segmentation serves as a preparatory step for our enhancement procedure. Our goal is to replace MATSim's average link-speed profile with real-world cycles that share similar driving metrics, applied in a piece-wise manner. In doing so, we aim to overcome MATSim's limitations in spatial congestion modeling (as detailed in Sec. 2.5.2) by diffusing severe congestion on short links across space and time. If a full driving cycle does not end at a multiple of 60 seconds, zeros are appended to the time series to account for typical vehicle stopping behavior.

3.1.2 MATSim driving cycle enhancement procedure

The method presented here refines our approach published in [209]. In that preliminary study, we replaced MATSim's simplified speed profiles with minute-wise segments of synthetic nature (as opposed to real-world measurements) having similar average velocities. Figure 3.3 illustrates the results, with the orange and blue lines representing MATSim's raw and enriched profiles, respectively.

In [209], we found our initial method satisfactory in terms of increased realism. However, their application to larger test cases revealed certain shortcomings that we aim to mitigate in this more advanced approach. The limitations of this initial enrichment procedure were as follows [209]:

- The underlying drive cycle dataset lacked real-world measurements at the time of its publication, relying instead on a limited set of synthetic driving cycles[5]. This artificial dataset failed to capture the full spectrum of human driving behavior, impacting the realism of the enriched MATSim profiles.
- The developed enrichment procedure was deterministic, favoring optimal solutions based solely on average velocity and acceleration criteria. This

[5] For more information on the synthetic driving cycles employed, please refer to [209].

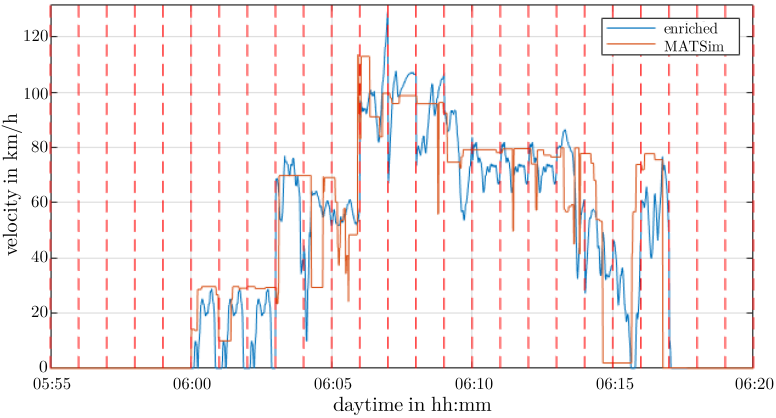

Figure 3.3: Illustration of an enriched average link-speed profile from MATSim using the method outlined in [209]. Red dashed lines denote the transition between two consecutive one-minute segments.

approach lacked a stochastic component, leading to driving dynamics dominated by a few well-fitting segments rather than reflecting the variability of longer velocity profiles, such as daily trajectories.

- Acceleration rates between consecutive segments were limited to acceptable values (< 5 m/s^2) in the initial approach, with segments failing this criterion discarded and replaced iteratively. However, this simplified approach frequently led to errors during vehicle simulation. For instance, the controller struggled to accurately follow the target signal, especially under conditions of high acceleration rates on steep gradients, resulting in distorted EV energy consumption.

To address these shortcomings, we propose the approach outlined in Figure 3.4. We start by providing a step-by-step description to clarify the intuition behind the procedure. Where appropriate, the procedure is mathematically formalized as we proceed. Additional equations and explanations are provided at the end of this section.

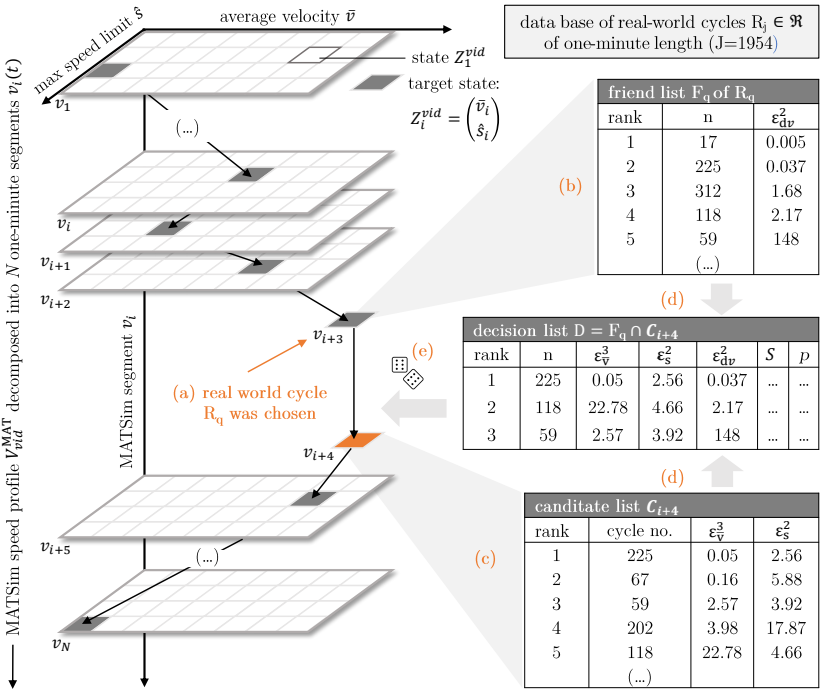

Figure 3.4: Schematic representation of the enrichment procedure. A selected MATSim speed profile $V_{vid}^{\mathrm{MAT}}(t)$ is divided into N one-minute segments $v_i(t)$, similar to all real-world cycles $R_j \in \mathfrak{R}$. For each segment $v_i(t)$, the average speed $\bar{v}$ and the maximum speed limit $\hat{s}$ encountered by the vehicle (vid) during the MATSim simulation are calculated. These metrics define the desired state Z_i^{vid}, represented by the dark gray tiles within the time-dependent $\bar{v}, \hat{s}$-planes. The enrichment process proceeds as follows: for the MATSim segment v_{i+3}, the real-world cycle segment R_q is identified as a valid match (see **(a)**). Using the segment's *friend list* F_q, real-world cycles $R_j \in \mathfrak{R}$ with similar speed values at their shared boundary with R_q are identified (see **(b)**). These candidates are ranked based on the error margin ϵ^2_{dv}, which penalizes deviations between the last velocity value of R_q and the first speed value of its potential successor R_j. Simultaneously, all cycles $R_j \in \mathfrak{R}$ are evaluated for their ability to match the desired state Z_{i+4}^{vid} using the time-dependent *candidate list* C_{i+4} (see **(c)**). Ranking metrics $\epsilon^3_{\bar{v}}$ and $\epsilon^2_{\hat{s}}$ are used to penalize discrepancies in average speed and speed limit. The sets F_q and C_{i+4} are then combined, with intersecting elements forming the *decision list* D (see **(d)**). Based on all three error margins, an element-wise score $S_{j,i}$ is calculated, which ultimately determines the probability $P_{j,i}$ that a cycle $R_j \in D$ will be chosen to replace the MATSim segment v_{i+4} (see **(e)**). The final enriched velocity profile $V_{vid}^{\mathrm{ENR}}(t)$ is then assembled from the sequence of chosen real-world cycles R_j.

A selected MATSim speed profile, denoted as $V_{vid}^{\mathrm{MAT}}(t)$ for a given vehicle (vid), is aligned with its corresponding speed limit profile[6] $S_{vid}^{\mathrm{MAT}}(t)$. Both profiles are segmented into one-minute intervals, expressed as $v_i(t)$ and $s_i(t)$. Here, $v_i(t)$ represents the velocity profile during the i-th one-minute cycle, where $t \in [0, 60)$ and $i = 1, 2, \ldots, N$, with N being the total number of one-minute segments in $V_{vid}^{\mathrm{MAT}}(t)$. This segmentation process is similarly applied to all real-world driving cycles $\mathfrak{R} = \{R_j \mid j = 1, \ldots, J\}$, where J represents the total of 1,954 valid one-minute driving cycles.

For each one-minute segment i, the average speed $\bar{v}_i$ from $v_i(t)$ and the maximum speed limit $\hat{s}_i$ from $s_i(t)$ encountered by the vehicle during the MATSim simulation are calculated. These metrics define the desired state Z_i^{vid}, represented by the dark gray tiles within the time-dependent $\bar{v}, \hat{s}$-planes in Figure 3.4. The desired state is mathematically expressed as:

$$Z_i^{vid} = \begin{pmatrix} \bar{v}_i \\ \hat{s}_i \end{pmatrix} = \begin{pmatrix} \frac{1}{60} \sum_{t=0}^{59} v_i(t) \\ \max\limits_{t \in [0,60)} s_i(t) \end{pmatrix}. \tag{3.1}$$

The challenge is to identify an appropriate real-world cycle R_j for each MATSim segment $v_i(t)$ that matches the desired state Z_i^{vid} – thus the target statistics $(\bar{v}_i, \hat{s}_i)$ – with sufficient precision. Suppose that for the velocity segment v_{i+3}, the real-world cycle R_q is identified as a valid match (highlighted in orange as **(a)** in Figure 3.4). To identify a suitable real-world cycle R_j for the subsequent MATSim segment v_{i+4}, the following steps are applied.

Since the final enriched speed profile $V_{vid}^{\mathrm{ENR}}(t)$ is composed of a series of real-world driving cycles $\mathfrak{R}$ (see Eq. 3.7), we first assess the inter-compatibility of these cycles. For a given segment R_q, we refer to its *friend list* F_q (see **(b)** in Figure 3.4) to identify potential real-world cycles R_j with compatible speed values at their shared boundary. A ranking is performed based on the error margin ϵ_{dv}^2 (Eq. 3.2), which penalizes deviations between the last velocity value of the cycle R_q and the first velocity value of its potential successor R_j.

[6] The speed limit profile is a time series that tracks the prevailing speed limit for each second the vehicle operates, derived from post-processing of MATSim network and event file.

Next, we evaluate each real-world driving cycle $R_j \in \mathfrak{R}$ for its ability to match the desired state Z_{i+4}^{vid}[7] using the time-dependent *candidate list* C_{i+4}**(c)**[8]. The ranking metrics used are $\epsilon_{\bar{v}}^3$ and $\epsilon_{\hat{s}}^2$ (detailed in Eq. 3.2), which penalize discrepancies in average speed $\bar{v}$ and maximum legal speed limit $\hat{s}$. This approach prevents congested real-world motorway cycles or access ramp maneuvers from being mixed into urban MATSim profiles. The sets F_q and C_{i+4} are then combined, with their intersecting (K) elements forming the *decision list* D_{i+4} **(d)**. Based on all three error metrics (ϵ_{dv}^2, $\epsilon_{\bar{v}}^3$, and $\epsilon_{\hat{s}}^2$), an element-wise score $S_{j,i}$ (Eq. 3.3) is calculated. This score serves as an indicator of the probability $P_{j,i}$ (Eq. 3.4), which represents the likelihood that cycle R_j is chosen as a valid solution for segment v_{i+4} **(e)**.

If the decision list D is empty[9], the computation resets to two MATSim segments prior (here v_{i+2}), generating alternative series of R_j by reassessing the probabilities $P_{j,i}$ to select the next suitable segment R_j. If this still leads to a dead end, the back iteration expands to even earlier MATSim segments. However, this situation was rarely encountered, underlining the robustness of our enrichment procedure.

The element-wise selection of appropriate real-world cycles R_j and their assembly into the enriched profile $V_{vid}^{\mathrm{ENR}}(t)$ can be mathematically expressed as follows. Let ψ represent a feature variable from the attribute set $\Psi = \{\bar{v}, \hat{s}, \mathrm{d}v\}$. We calculate the error margin ϵ_ψ using

$$\epsilon_\psi = \frac{|\psi_{R_j} - \psi_Z|}{\max(|\psi_{R_j} - \psi_Z|)}, \tag{3.2}$$

where ψ_Z represents the corresponding attribute of the desired state Z, and $\max(|\psi_{R_j} - \psi_Z|)$ provides normalization across all real-world cycles $R_j \in \mathfrak{R}$. For the error margin ϵ_{dv}, which penalizes speed deviations at the boundary

[7] Specifically, the statistics $(\bar{v}_{i+4}, \hat{s}_{i+4})$

[8] Unlike the static *friend list* (F_q), the dynamic *candidate list* (C_{i+4}) is not precomputed. Its elements are generated on the fly for each $v_i(t)$, adapting over time to minimize memory usage.

[9] This phenomenon occurs when the preceding *friend list* F or *candidate list* C are sparsely populated due to inadequately fitting target statistics $(\bar{v}, \hat{s}, dv)$. Given $D = F \cap C$, the *candidate list* D may be empty as a consequence.

between two consecutive cycles R_j in $\mathfrak{R}$, we use the attribute ψ_{R_f} of R_j's *friend* R_f[10] as the reference, instead of the desired state Z.

The element-wise scoring parameter $S_{j,i}$ is then computed as

$$S_{j,i} = \epsilon_{\bar{v}}^3 + \epsilon_{\hat{s}}^2 + \epsilon_{\mathrm{dv}}^2. \tag{3.3}$$

In this scoring process, we assign the greatest weight to the error margin in average velocity $\epsilon_{\bar{v}}^3$. This is done to minimize deviations in the overall distance traveled per day between the MATSim speed profile $V_{vid}^{\mathrm{MAT}}(t)$ and the enriched speed profile $V_{vid}^{\mathrm{ENR}}(t)$. Given the scores $S_{j,i}$ for all elements R_j in the decision list D, we calculate the probability $P_{j,i}$ as follows:

$$P_{j,i} = \frac{\exp\left(-S_{j,i}\right)}{\sum_{k=1}^{K} \exp\left(-S_{k,i}\right)}. \tag{3.4}$$

Here, the term $\exp(-S_{j,i})$ guarantees that lower scores, which represent better matches, result in higher probabilities. Meanwhile, the sum in the denominator normalizes these probabilities, so that the total probability across the K elements in the decision list $D = \{d_1, d_2, ...d_k\}$ adds up to 1.

To determine which element is selected based on the given probabilities, we first compute the cumulative probabilities for each element in the decision list. This is achieved by summing the probabilities sequentially. For an element $R_j \in D$, the cumulative probability C_j is given by:

$$C_j = \sum_{k=1}^{j} P_{k,i}, \tag{3.5}$$

where $P_{k,i}$ is the probability associated with the k-th element and C_j represents the cumulative probability up to and including element R_j. We then generate a random number r uniformly distributed between 0 and 1 to simulate the outcome of a probabilistic dice roll. To select the element, we identify the

[10] By analogy, each R_j in the friend list F is referred to as a potential friend.

smallest index j for which the cumulative probability C_j is greater than or equal to the random number r. In other words, we find j such that:

$$C_j \geq r. \tag{3.6}$$

By following these steps, we ensure that the selection of each element R_j is proportional to its probability $P_{j,i}$. The outcome of the dice roll determines the real-world cycle R_j that is assigned to the MATSim segment $v_i(t)$.

This selection is recorded in a mapping function, which is then used to assemble the final enriched profile $V_{vid}^{\mathrm{ENR}}(t)$. Let $\sigma(i)$ be a function that maps each MATSim segment $v_i(t)$ to a specific real-world cycle $R_{\sigma(i)}$. This mapping function $\sigma(i)$ identifies which real-world cycle R_j is selected to replace the i-th segment of the original MATSim profile (e.g., $\sigma(i+3) = q$ for segment v_{i+3} as shown in Figure 3.4). Since $\sigma(i)$ is not necessarily sequential, R_j may not follow the order $j = 1, 2, \ldots, N$ and can be repeated. The enriched velocity profile $V_{vid}^{\mathrm{ENR}}(t)$ is then formed by concatenating these selected cycles:

$$V_{vid}^{\mathrm{ENR}}(t) = R_{\sigma(i)}\,(t - 60(i-1)) \quad \text{for } t \in [60(i-1), 60i). \tag{3.7}$$

Here, the time t is adjusted by $60(i-1)$ to properly align each selected cycle within the overall profile. Over the entire duration of the MATSim profile (consisting of N one-minute segments), the enriched velocity profile $V_{vid}^{\mathrm{ENR}}(t)$ can also be expressed as a piecewise function:

$$V_{vid}^{\mathrm{ENR}}(t) = \begin{cases} R_{\sigma(1)}(t) & \text{for } t \in [0, 60) \\ R_{\sigma(2)}(t - 60) & \text{for } t \in [60, 120) \\ \vdots & \vdots \\ R_{\sigma(N)}(t - 60(N-1)) & \text{for } t \in [60(N-1), 60N). \end{cases} \tag{3.8}$$

The so generated profile $V_{vid}^{\mathrm{ENR}}(t)$ is considered acceptable if the daily traveled distance deviates by no more than 5% from the daily mileage of the original MATSim profile $V_{vid}^{\mathrm{MAT}}(t)$. The relative error $e_{vid}^{\text{day, rel}}$ is calculated as follows:

$$e_{vid}^{\text{day, rel}} = \frac{\left| \left(\sum_{t=1}^{T} V_{vid}^{\text{ENR}}(t) \cdot \Delta t \right) - \left(\sum_{t=1}^{T} V_{vid}^{\text{MAT}}(t) \cdot \Delta t \right) \right|}{\sum_{t=1}^{T} V_{vid}^{\text{MAT}}(t) \cdot \Delta t}, \tag{3.9}$$

where vid represents the vehicle's identification number, T signifies the total number of seconds in one day, and Δt is the time step (1 second).

Given the possibility of multiple enriched velocity profiles $V_{vid,b}^{\text{ENR}}(t)$ for a single MATSim driving cycle $V_{vid}^{\text{MAT}}(t)$ due to stochastic variations in the enrichment process, we generate $B = 1000$ such profiles per MATSim vehicle. The enriched profile $V_{vid,b}^{\text{ENR}}$ with the smallest relative error $e_{vid,b}^{\text{day, rel}}$ in traveled distance (Eq. 3.9) is selected as the best match. The results of this analysis are shown in Figure 3.5.

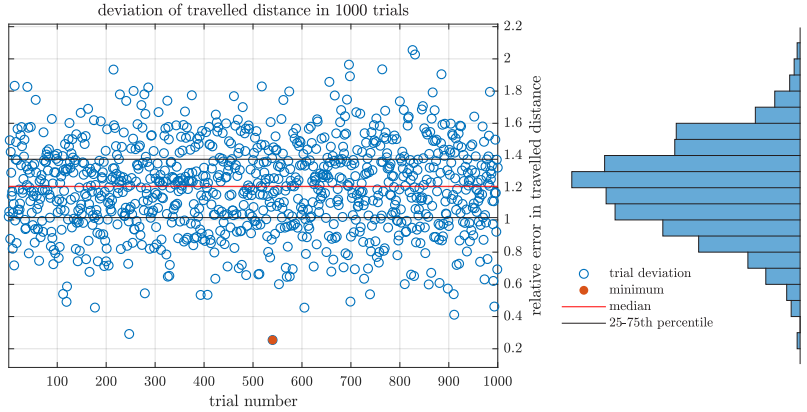

Figure 3.5: Stochastic variations in the enrichment process for a given MATSim profile $V_{vid}^{\text{MAT}}(t)$ across 1000 trials ($B = 1000$). The left panel shows the relative deviations in daily mileage between multiple enriched profiles $V_{vid,b}^{\text{ENR}}(t)$ and the original MATSim profile. On average, the deviation is approximately 1.2%, with the best result featuring a relative error of 0.24%. The right panel presents a scaled histogram of the frequency distribution of these relative errors.

In our scenario, the average deviation between the enriched and original profiles is approximately 1.2%, with deviations exceeding 2% being rare. These results fall within an acceptable range. For comparison, the acceleration sensor's neglect of latitudinal headway contributes to larger offsets (greater than 3%) in daily traveled distance.

3.1.3 Profile smoothing for autonomous driving

Since real-world measurements do not capture autonomous driving behavior, we apply a smoothing technique to our enriched profiles, similar to the approach in [131, 120]. Specifically, we use a smoothing spline that minimizes the following objective function[11]:

$$p \underbrace{\sum_i w_i (y_i - s(x_i))^2}_{\text{MSE-term}} + (1-p) \underbrace{\int \left(\frac{\mathrm{d}^2 s}{\mathrm{d}x^2}\right)^2 \mathrm{d}x}_{\text{curvature-term}}, \tag{3.10}$$

where w_i denotes the specified weight (in our case set to one), and p is the smoothing parameter that controls the extent of smoothing. The first term represents the mean square error (MSE) between the value y_i and its smoothed counterpart $s(x_i)$ at point x_i. The second term describes the second derivative of s with respect to x, reflecting the curvature of s at x_i.

The smoothing parameter p ranges between 0 and 1, balancing the two terms: $p = 0$ corresponds to a least-squares linear fit, while $p = 1$ favors a cubic spline interpolation with a stronger emphasis on minimizing the MSE term. The influence of the smoothing parameter p on the velocity profile and acceleration distribution is illustrated in Figure 3.6.

The choice of smoothing level depends on the specific AV scenario [131]:

Autonomous driving profiles should minimize extreme events such as abrupt accelerations and sudden braking, typical of human-operated vehicles. Ad-

[11] "Smoothing Splines Documentation," MATLAB, https://de.mathworks.com/help/curvefit/smoothing-splines.html. Accessed: Jun 19, 2023.

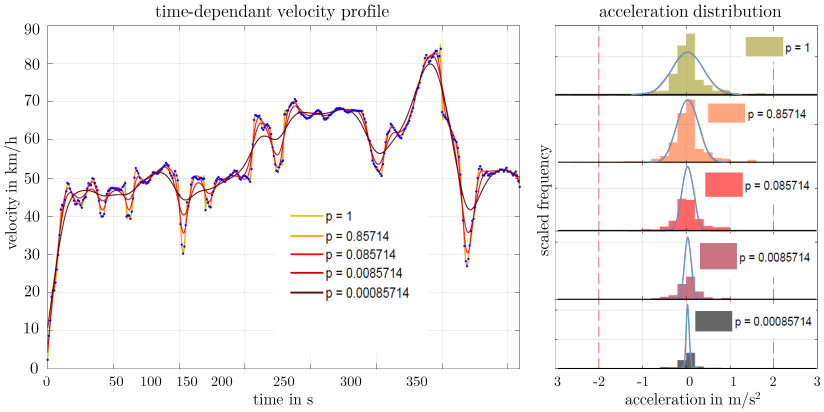

Figure 3.6: Impact of smoothing parameter p on velocity profile (left) and acceleration distribution (right). When $p = 0$ it produces a linear interpolation with least-square error (maximal smoothness), while $p = 1$ results in a cubic spline approximation (no smoothing). Increased smoothing flattens extreme driving events, reducing acceleration rates, as shown by the bluish kernel density estimates in the right panels. Pre-published in [120].

vanced capabilities in intelligent and connected vehicles allow anticipation of upcoming driving conditions, enabling smoother responses to dynamic traffic situations. Thus, profiles of fully automated vehicles in highly intelligent transport environments are expected to exhibit very smooth behavior ($p \to 0$).

Conversely, autonomous vehicle movements are heavily influenced by interactions with other vehicles, especially in congested traffic conditions where human-driven vehicles still dominate. Factors like traffic signals and signs significantly shape AV driving behavior. Therefore, during the initial phase of AV implementation, speed profiles should mirror those of human-driven cars to prevent delays, missed signals, or collisions with other vehicles. In this context, excessive smoothing should be avoided ($p \to 1$).

As shown by the bluish kernel density estimates in the right panels of Figure 3.6, increased smoothing results in lower acceleration rates. To determine an appropriate smoothing factor p, we refer to the recommended AV acceleration rates discussed in [144]. The authors suggest that AV acceleration rates should strike a balance between the smooth behavior of rail-based public transport and typical human driving. Given the absence of commercially available automated

vehicles, optimal AV driving behavior is inferred from various literature values and experimental studies, as outlined in Table 3.2.

Table 3.2: Anticipated longitudinal and lateral acceleration rates in ms^{-2} for AD, derived from the smooth driving behavior of rail-based public transport (leftmost column) to typical human driving (rightmost column) (partially from [144, 114]).

	(a) Light Rail Transit [40, 144]	(b) semi AD (Tesla S) [144]	(c) AD (Wizard) [226]	(d) AD (FKFS) [144]	(e) comfort formula [127, 144]	(f) typical human driver [114, 104, 36, 193]	(g) upper limit human driver [114, 104, 36, 193]
a^{+}_{lon}	1.3	1.3	2.0	2.05	2.1	2.47	4.86
a^{-}_{lon}	-1.3	-1.3	-2.6	-2.35	-2.1	-3.27	-7.47
a^{+}_{lat}	1.3	1.3	2.1	2.05	2.1	4.10	5.3

(a) Acceleration rates in rail-bound transport are maintained relatively constant to minimize jolts and jerks, enabling passengers to engage in secondary activities such as working, reading, or watching a movie. Real-world experiences report maximum longitudinal acceleration rates of $1.3\,ms^{-2}$ for S-Bahn Stuttgart, consistent with international Light Rail Transit (LRT) standards [144, 40].

(b) Semi-automated vehicle behavior provides another benchmark. During road measurements with a Tesla Model S, automated lane changes and overtaking maneuvers recorded maximum accelerations around $1.3\,ms^{-2}$, aligning with LRT values [144].

(c) A study by Eindhoven University simulated autonomous vehicle driving scenarios using a specially trained driver (known as a 'wizard') while participants evaluated comfort levels and acceleration preferences [226]. Maximum longitudinal accelerations of up to $2\,ms^{-2}$ and decelerations of $-2.6\,ms^{-2}$ were deemed acceptable, with some participants preferring lower acceleration values when in a passenger role.

(d) Recommended AD acceleration rates, as discussed in [144], are guided by the smooth driving behavior of rail-based public transport on one end and typical human driving behavior on the other.

(e) An additional reference point is the ride quality score *UIC rq note* [127, 201] issued by the international Union of Railways (UIC) to estimate railway route comfort. For automated vehicles, a comparable UIC rq value of 1.4 is desired, corresponding to a defensively average parameterization. A suitable combination of relevant parameters was iteratively determined by [144], resulting in $2.1\,ms^{-2}$ for both longitudinal and lateral acceleration.

(f-g) Data on typical human acceleration rates were extracted from [114], who synthesized findings from [104, 36, 193].

Deviating slightly from the recommended AD acceleration rates by [144] (highlighted in yellow in Table 3.2 (d)), we select a smoothing factor of $p = 0.08$, ensuring that 99.99% of all velocity-acceleration pairs remain below $2\,ms^{-2}$ (cf. Figure 3.7).

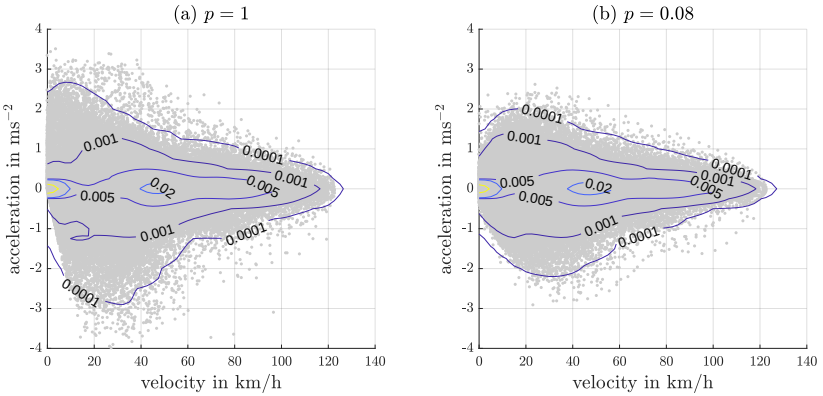

Figure 3.7: Effect of smoothing factor p on velocity-acceleration pairs (approximately 17,000 total). The left panel shows kernel density estimates for v-a events in original (unsmoothed) driving cycles ($p = 1$), while the right panel shows smoothed cycles ($p = 0.08$). The scaled contour lines facilitate comparison between panels. Noticeable reduction in extreme acceleration events is observed.

3.2 Microscopic traffic simulation in SUMO

Our second approach to enhancing MATSim's average link-speed profiles involves subjecting mesoscopically simulated vehicle trajectories to additional microscopic traffic simulation, as discussed in [209, 207].

Numerous microscopic traffic simulation tools are available, each offering distinct features and tailored to specific fields of application. For our work, we select the open-source framework SUMO[12] (*Simulation of Urban Mobility*) [135], which is widely utilized in traffic management, traffic light evaluation, and vehicular communication studies. SUMO is well-established within the scientific community and provides multiple interfaces that allow external applications to interact with the simulation in real-time. We employ the *Traffic Control Interface* (TraCI) to retrieve and manipulate object attributes instantaneously. Additionally, our choice of SUMO is reinforced by its existing

[12] SUMO, https://sumo.dlr.de/docs/index.html. Accessed: Sep. 29, 2024.

use in corporate applications, which facilitates the seamless integration of our solution with established tool pipelines.

We opted against a city-wide simulation of SAEV fleets directly in SUMO for three primary reasons. First, the simulation of SAEV travel demand requires an agent-based modeling component, as no real-world data are available. Since SUMO is not inherently agent-based [168], it necessitates integration with the *Java Agent Development Framework* (JADE) [13] to model SAEV mode acceptance. Second, estimating SAEV travel demand in competition with other modes (such as public transport and private cars) is computationally intensive, requiring iterative simulations to achieve a quasi-user equilibrium. SUMO is not capable of performing such extensive multi-modal simulations within a reasonable timeframe. Even with MATSim, which is known for efficiently handling large-scale scenarios, achieving equilibrium in DRT simulations can take several weeks, depending on the simulated population share (typically 1% to 10%) and the number of iterations. Third, SUMO requires more detailed network data than MATSim, and the quality of OSM data is often insufficient [79]. This necessitates extensive network refinement, which is impractical on a large scale due to the significant time and resource demands. In contrast, MATSim is more tolerant of network modeling errors.

As a result, we decided to leverage MATSim for SAEV travel demand modeling and other computationally intensive tasks, while reserving SUMO for localized network areas to enhance detail, particularly in driving dynamics. This approach optimizes computation runtime. After conducting extensive simulations in MATSim, as detailed in Chapter 2, we explore SUMO in a subsequent test-case analysis in Sec. 3.3. Specifically, we assess (i) whether SUMO meets our requirements for detailed driving dynamics, (ii) how effectively SUMO's operational principles and interfaces can be integrated, potentially leading to a robust coupling of the two tools, and (iii) whether this multi-layered traffic simulation justifies the additional modeling effort compared to our less elaborate, data-driven enrichment procedure.

3.2.1 Car-following model by Krauss

Subjecting MATSim SAEV vehicles to subsequent microscopic simulation in SUMO enables us to replace the spatial queue model—which is insufficient for

our use case—with a sophisticated car-following model. This provides a more realistic representation of the ego-vehicle's interactions[13] with other vehicles and network infrastructure.

Traffic dynamics in SUMO are modeled using car-following models, such as the *Krauss* model [122] and the *Intelligent Driver Model* (IDM) [204], as well as lane-change models like *LC2013* [71]. In this study, we employ the default Krauss model, where vehicles travel as fast as possible while maintaining a safe distance from the vehicle ahead. The safe speed (v_{safe}) is calculated as follows [121]:

$$v_{\text{safe}}(t) = v_l(t) + \frac{g(t) - v_l(t)\tau}{\frac{\bar{v}}{b} + \tau} \tag{3.11}$$

where $v_l(t)$ represents the speed of the leading vehicle, $g(t)$ is the gap to the leader, τ denotes the reaction time, b is the maximum deceleration of the following vehicle, and $\bar{v}$ is the mean velocity of both the following and leading vehicles.

Since v_{safe} may exceed the prevailing speed limit v_{limit} or the vehicle's maximum capability v_{capa}, the desired velocity is constrained to $v_{\text{des}} = \min\{v_{\text{safe}}, v_{\text{limit}}, v_{\text{capa}}\}$. Additionally, SUMO includes a driver imperfection parameter σ, which introduces random deceleration to simulate speed fluctuations and spontaneous traffic jams under high-density conditions. Each vehicle also draws an individual `speedFactor` from a normal distribution to reflect a range of human driving styles, including drivers who notoriously stay above or below the legal speed limit. The `speedFactor` is defined as `norm(mean, dev)`, where `mean` is the mean of the normal distribution, and `dev` is the standard deviation. For instance, a `speedFactor="norm(1,0.1)"` produces a speed distribution where 95.4% of vehicles travel between 80% and 120% of the legal speed limit.[14] To mitigate extreme deviations in driving behavior, such as excessive slowing or speeding, the individual `speedFactor` can be constrained within specified lower and upper bounds. This is implemented using a truncated normal distribution, specified as `normc(mean, dev, min, max)`.

[13] The term "ego-vehicle" refers to the simulated vehicle at the center of the study, with all observations, analyses, and interactions considered from its perspective.

[14] This range is derived from the standard deviation of normally distributed data.

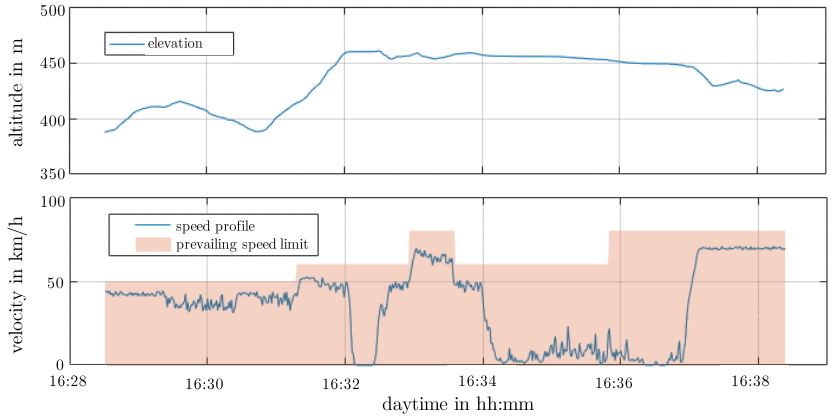

Figure 3.8: Elevation (top panel) and velocity profile (bottom panel) for an exemplary ego-vehicle simulated with SUMO. The reddish area in the bottom panel indicates the prevailing speed limit encountered by the vehicle throughout its trajectory.

Figure 3.8 illustrates an exemplary velocity profile obtained from a default SUMO simulation[15]. The figure also displays information on the prevailing speed limit and the vehicle's elevation profile.

3.2.2 Parameter fitting to adapt for autonomous driving

There are numerous contributions to simulate automated driving behavior in SUMO. Notable models include the integrated *Adaptive Cruise Control* (ACC) car-following model[16] [151, 220] and the *Cooperative Adaptive Cruise Control* (CACC) car-following model[17] [151, 220, 221]. These models divide

[15] For details on default parameters and vehicle specifications, refer to https://sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html#car-following_model_parameters and https://sumo.dlr.de/docs/Vehicle_Type_Parameter_Defaults.html. Accessed: Sep. 03, 2024.

[16] "ACC Documention," SUMO, https://sumo.dlr.de/docs/Car-Following-Models/ACC.html. Accessed: Sep. 29, 2024.

[17] "CACC Documention," SUMO, https://sumo.dlr.de/docs/Car-Following-Models/CACC.html. Accessed: Sep. 29, 2024.

(cooperative) adaptive cruise control into three modes: (i) cruise control, (ii) gap control, and (iii) gap-closing control. A fourth mode, (iv) collision avoidance, was introduced within the *TransAID* project[18] when both models were integrated into SUMO. Modes (iii) and (iv) aim to enable stable and safe car-following behavior in SUMO, addressing string instabilities[19] observed in the original publication [151], which revealed unstable behavior in ACC-controlled vehicle platoons.

To simplify the process, we use a modified Krauss model instead of the aforementioned advanced AD car-following models. This choice improves computation runtime, which is crucial during the early stages of solution screening and toolchain development. Should a MATSim-SUMO tool coupling prove beneficial for our use case, we will ensure that existing contributions from both MATSim and SUMO, such as car-following models and dispatching strategies, remain interchangeable. For similar reasons, we have chosen to omit EV energy consumption and charging behavior from the forthcoming test-case analysis in Sec. 3.3, given that these elements are already thoroughly addressed by the MATSim framework.

Drawing from the literature summarized in Table 3.3 and our analysis in Section 3.1.3, we define two parameter sets for fully automated driving vehicles: AD-III and AD-IV, as highlighted in yellow in Table 3.3.

(i) The AD-III set is intended for scenarios where human-driven vehicles remain in operation. As outlined in the yellow-highlighted row of Table 3.3, this model specifies a minimal gap (g) to leading vehicles that is greater than the technically necessary minimum. This adjustment prevents human drivers from feeling pressured. Additionally, the desired speed (v_{des}) is slightly above the legal speed limit to avoid obstructing human-driven vehicles, which often exceed speed limits.

(ii) The AD-IV set is designed for simulating connected autonomous vehicles (CAVs) within fully optimized transportation systems. This model allows for smaller minimal gaps without the necessity of exceeding speed limits.

[18] TransAID, https://www.transaid.eu/. Accessed: Sep. 29, 2024.

[19] String instabilities refer to traffic flow patterns in dense traffic conditions, characterized by the formation of localized clusters or "strings" of vehicles exhibiting unstable behavior, such as stop-and-go traffic or oscillations in speed and spacing.

Table 3.3: Parameters in the Krauss car-following model across different levels of automated driving. The AD levels correspond to the automation levels defined in SAE J3016 [106], with 0 denoting no driving automation and 5 representing full driving automation. Here, g is the offset to the leading vehicle in traffic jams; $+a/-a$ are the acceleration and deceleration rates for the vehicle type; $-\hat{a}$ represents the maximum emergency deceleration; σ denotes the driver imperfection ranging from 0 to 1; τ is the desired (minimum) time headway (reaction time); and the `SpeedFactor` represents the individual speed multiplier drawn from a normalized distribution characterized by mean and standard deviation.

selected Krauss-based car-following models	AD level	g m	$+a$ m/s^2	$-a$ m/s^2	$-\hat{a}$ m/s^2	σ	τ s	`speedFactor` mean	`speedFactor` deviation
SUMO default	0	2.5	2.6	4.5	9	0.5	1.0	1	0.1
human-I [183]	0	-	2.0	4.0	-	0.5	0.7	1.03	0.103
human-II [183]	0	-	2.0	4.0	-	0.5	0.7	1.04	0.161
from human to	0	2.5	2.6	4.5	8	0.5	1.0	-	-
autonomous	1	2	3.05	4.5	8	0.4	0.95	-	-
driving [136]	2	1.5	3.5	4.5	8	0.3	0.9	-	-
with	3	1.25	3.6	4.5	8	0.2	0.8	-	-
adaptions	4	0.75	3.7	4.5	8	0	0.7	-	-
from [123, 181]	5	0.5	3.8	4.5	8	0	0.6	-	-
AD-I [84]	5	-	1.3	1.3	-	0	0.1		
AD-II [183]	5	-	2.0	4.0	-	0	0.4	1	0
AD-III	5	2.5	2.0	2.0	8	0	0.4	1.1	0
AD-IV	5	0.5	2.0	2.0	8	0	0.4	1	0

Both Krauss parametrizations assume fully automated driving capabilities (*Level 5* as per SAE J3016 [106]), with zero driver imperfection ($\sigma = 0$) and uniform vehicle speed capacities (i.e., zero `speedFactor` deviations). Note that lateral driving dynamics and cooperative driving strategies are not incorporated in these models.

3.3 Test case analysis with focus on human driving behavior

In this section, we evaluate the extent to which the methods described in Sections 3.1 and 3.2 can produce sufficiently realistic driving dynamics within a case study of limited spatial scope. This assessment aims to refine the focus of our thesis and provides an initial indication of whether a multi-level traffic simulation approach, integrating MATSim and SUMO, is viable for the broader context of our use case. The test case analysis presented here has been previously published in our work [209].

3.3.1 Methodological approach and test case design

We use an existing MATSim model for the Stuttgart region to extract and populate three distinct network areas. The MATSim model, developed by Robert Bosch GmbH in collaboration with the *Verband Region Stuttgart*, utilizes activity-chain based travel demand generated by *mobiTopp* [145]. This agent-based travel demand model was developed by the *Institute for Transport Studies (IfV), Karlsruhe Institute of Technology* and integrated into MATSim based on the work of [39]. We have identified three test cases, each varying in road type, network topology, and right-of-way rules:

1. The *Bergheimer Steige* test case features no crossroads but includes sharp turns and road gradients up to 15%. This scenario aims to analyze the impact of slopes and curves on vehicle speed in simulations.
2. The *Motorway A8* (from Kreuz Stuttgart to AS Stuttgart Möhringen) is used to examine traffic dynamics on motorways.
3. The *Kräherwald* test case (from Kräherwald/junction Zeppelinstraße to the University of Stuttgart) exhibits a mixed inner-city and highway character and forms part of the FKFS cycle, as illustrated in Figure 3.9.

All test cases are simulated for a single day in both MATSim and SUMO. For each test case, driving cycles are generated through three distinct approaches: (a) standard MATSim simulation (Sec. 2.1), (b) enhanced MATSim simulation incorporating real driving cycles through an enrichment procedure (Sec. 3.1),

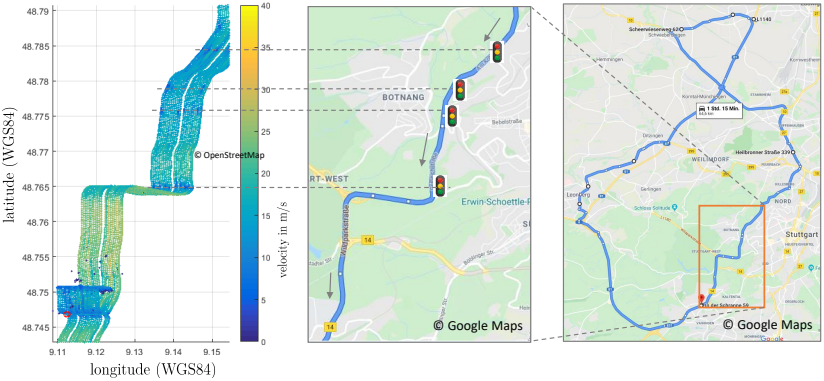

Figure 3.9: The right plot shows the FKFS circuit, with the grey rectangle highlighting the *Kräherwald* test case. The zoomed trajectory of this test case is at the center, and the left plot presents the spatial velocity profiles of 22 measured FKFS cycles for the *Kräherwald* test case. To enhance readability, each cycle was given an offset in the longitudinal direction.

and (c) SUMO simulation using travel demand synthesized from MATSim and (Sec. 3.2). The results are evaluated with respect to driving dynamics and energy-related KPIs, and are benchmarked against real-world driving cycles. Although our primary aim is to simulate autonomous driving, the focus here is on human driving characteristics to facilitate profile validation with real-world data.

Energy-related KPIs are derived from vehicle simulations conducted in *GT-Suite*[20], utilizing the vehicle specifications outlined in Table 3.4. We use an existing EV model [190] (see Sec. 8.2), which is specifically designed to identify the optimal EV architecture and powertrain configuration for particular use cases. The energy-related KPIs, along with driving dynamics, are compared to 22 GPS-tracked FKFS cycles for the *Kräherwald* test case. To ensure comparability and to account for systematic errors introduced be the vehicle model in GT-Suite, measured velocity profiles are also utilized in GT-Suite simulations to calculate

[20] GT-SUITE, https://www.gtisoft.com/gt-suite/gt-suite-overview/. Accessed: Sep. 29, 2024.

Table 3.4: Vehicle specifications as used during GT-Suite simulation

parameter	value
vehicle mass m	1545 kg
rolling resistance coefficient μ_r	0.011
frontal area of the vehicle A	2.2 m^2
drag coefficient C_d	0.27
friction coefficient μ	1
battery capacity E_{bat}	60 kWh
engine power P_{eng}	200 kW

energy consumption, despite the availability of real-world energy consumption data.

3.3.2 Network generation and travel demand synthesis

To construct a SUMO network based on an existing MATSim model, we independently import the relevant geographical area from OSM using SUMO's `NETCONVERT` tool. Although it is possible to import SUMO networks directly from MATSim, this method is suboptimal for our purposes because MATSim omits certain network details that SUMO requires.

Figure 3.10 illustrates the differences between MATSim and SUMO networks for the *Bergheimer Steige* area in Stuttgart. MATSim networks can be imported using an OSM network reader with varying levels of detail: simpler networks with fewer links (Fig. 3.10, panel b)) or more complex networks that more accurately capture curved road shapes (using the `keep-path` configuration, as shown in Fig. 3.10, panel c)). However, the more detailed network performed poorly in our simulations due to numerous short links, which caused artifacts as discussed in Section 2.5.2.

MATSim paths, regardless of their import resolution, integrate adjustments for road geometry and elevation differences, leading to path lengths that closely

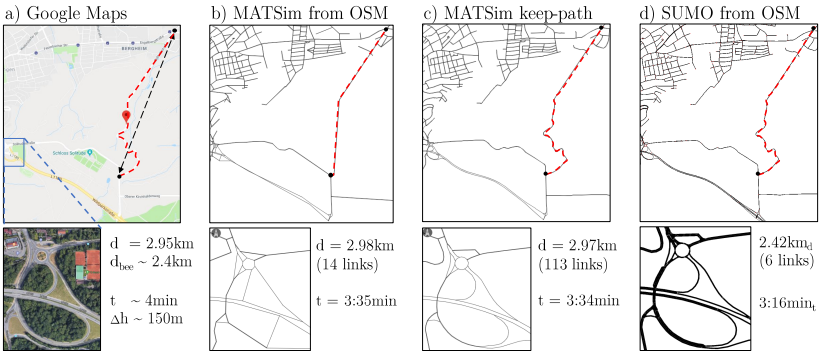

Figure 3.10: Comparison of network representations for the Bergheimer Steige test case, as described in Section 3.3.1. The top panels provide an aerial view of the entire test case, highlighting differences in road geometry and network topology across the various network import methods in MATSim and SUMO. The bottom panels focus on a selected traffic node, illustrating variations in network granularity and realism. Key metrics presented in the respective bottom-right corners include the path length d, the number of road segments (links) into which the test case is divided, and the travel time t under free-flow conditions. In panel a) (Google Maps), additional metrics such as beeline distance d_{bee} and height differences $\triangle h$ are also shown.

align with those from Google Maps[21]. In contrast, SUMO networks, while featuring advanced network design (Fig 3.10, panel d)), do not automatically account for length variations due to altitude differences in their two-dimensional representation. To address this, we incorporate altitude data from an additional elevation model.

Travel demand in our SUMO simulation is entirely derived from MATSim. We begin by identifying all MATSim links surrounding the selected test cases and recording the vehicles traversing these links, including their vehicleID, route, and entry and exit times. As MATSim handles a sampled population, we scale the travel demand in SUMO by injecting *cloned* vehicles. To prevent severe gridlocks, we introduce a random time offset, drawn from a Gaussian distribution, to the entry times of these cloned vehicles. Once departure times

[21] Google Maps, https://www.google.com/maps. Accessed: Sep. 29, 2024.

are established, we manually map MATSim linkIDs to SUMO edgeIDs[22] and generate the final trips file for SUMO. These vehicle trips are then converted into vehicle routes using SUMO's `DUAROUTER` tool. Due to the – by design – limited scale of the test cases, characterized by single roads with minimal alternative routes, the routes in SUMO closely align with those in MATSim.

3.3.3 KPI comparison

The assessments presented refer to Table 3.5, which summarizes aggregated KPIs for each test case and simulation approach (MATSim, enriched, and SUMO). To optimize computation runtime, only a subset of simulated vehicles was tracked at a microscopic level for each test case. The table specifies the respective number of simulated and microscopically tracked vehicles in parentheses. Data labeled 'enriched' refers to the initial enrichment procedure described in Section 3.1.2 and [209], rather than the advanced method outlined later in the same section, which was developed to overcome limitations in the original approach.

Aggregated vehicle dynamics and energy metrics The average traveled distance of all tracked vehicles is comparable across simulation scenarios, with variations primarily due to differences in network design and import methods. In both MATSim and SUMO, changes in road attributes in OSM result in the creation of new links or edges. However, while SUMO represents road segments as continuous (curved) edges, MATSim uses straight lines, introducing artificial nodes to approximate road geometry. Consequently, one SUMO edge often represents multiple MATSim links, resulting in longer distances in SUMO due to the representation of road curvature. In small test cases like ours, deviations in average travel distance between MATSim and SUMO may seem significant, but these differences diminish on a larger scale.

The distances calculated for the enriched scenario are artificial and do not align with actual target trajectories, which is a major drawback of our initial enrichment procedure. These discrepancies in traveled distance also affect

[22] For further details on network representation and terminology in MATSim and SUMO, please refer to Sec. 4.2.1.

Table 3.5: Aggregated KPIs for three test cases (Bergheimer Steige, Motorway A8, and Kräherwald). The number of tracked and simulated ego-vehicles for each test case is indicated in parentheses. Data previously published in [209].

	KPI	MATSim	enriched	SUMO
Bergheimer St. (359/7170)	∅ traveled distance in km	2.9	2.8	3.1
	∅ traveled time in min	4.3	4.8	4.6
	∅ congestion rate	0.85	0.72	0.84
	∅ velocity in km/h	43	36	42
	∅ energy consumption in kWh/100km	9.2	11.6	16
Motorway A8 (385/379591)	∅ traveled distance in km	6	5	6.8
	∅ traveled time in min	3.7	4.1	3.7
	∅ congestion rate	0.88	0.55	0.97
	∅ velocity in km/h	92	59	101
	∅ energy consumption in kWh/100km	23.8	15.8	24
Kräherwald (444/110779)	∅ traveled distance in km	5.5	5.4	5.7
	∅ traveled time in min	7.8	8.2	6.6
	∅ congestion rate	0.82	0.76	0.85
	∅ velocity in km/h	52	48	53
	∅ energy consumption in kWh/100km	11.8	12.4	15.6

secondary metrics such as travel time and energy consumption. Therefore, our more sophisticated enrichment approach emphasizes selecting enriched cycles with a high fit in daily mileage (cf. Figure 3.5) to mitigate this issue. The Motorway test case, in particular, exhibits discrepancies in traveled distance exceeding the 5% error margin. These deviations are also attributed to the limited availability of high-speed one-minute segments. The driving cycle dataset referenced in Sec. 3.1.1 was not available at the time of our initial enrichment procedure described in [209]. Thus, the accuracy of the enrichment procedure highly depends on a diverse set of measured driving cycles.

Average travel time, velocity, and energy consumption are highly dependent on congestion. Different approaches in modeling traffic dynamics (queue vs. car-following models) and road networks (node-based vs. sophisticated intersections) lead to different traffic conditions, causing the same vehicle to experience varying delays in MATSim and SUMO. The left panel of Figure 3.11

shows discrepancies in traffic states λ between MATSim and SUMO, displayed in a log-log plot where each dot represents a tracked vehicle. The traffic state, defined as the ratio of free-flow travel time to actual travel time, indicates free-flow conditions at $\lambda = 1$ and blocked roads near $\lambda = 0$. Due to SUMO's allowance for overspeeding (up to 20%) and link length discrepancies with MATSim, traffic states greater than $\lambda = 1$ can result. A perfect match in traffic state would theoretically result in a diagonal line, but Figure 3.11 (left) reveals that this alignment is rarely achieved, warranting further investigation. The histograms on the right indicate that MATSim traffic conditions are generally too optimistic, as vehicle queues fail to propagate upstream properly (see Sec. 2.5.2), or occasionally too pessimistic, particularly on short links.

Significant differences in velocity and acceleration distributions emerge across all simulation approaches. Figure 3.12 shows scaled velocity and acceleration histograms for 359 simulated vehicles in the Bergheimer Steige test case, with both metrics tracked every second. Pure MATSim simulation yields unrealistic driving dynamics, considering only average link speeds and showing minimal acceleration variations. Acceleration rates predominantly remain zero, with abrupt transitions between links constrained by the speed limits of the subsequent links. Although the enriched profiles are more realistic, they partially inherit this deficiency in driving behavior from MATSim and do not fully overcome MATSim's limitations in spatial congestion modeling, as discussed in Section 2.5.2. In contrast, SUMO exhibits bell-shaped distributions centered around local maxima. Both enriched MATSim and SUMO simulations limit maximum acceleration to $\pm 5\,\mathrm{m/s^2}$ by design; however, SUMO's acceleration rates are excessively uniform and exhibit overly smooth distributions compared to real-world data, as also noted in [7, 102, 199].

Time-dependent speed profiles Figure 3.13 illustrates the time-dependent velocity profile for a selected MATSim vehicle in the Bergheimer Steige test case (blue line). The corresponding vehicle representations for the enrichment approach and SUMO simulation, each having the same vehicle ID, departure time, and trajectory, are depicted in red and green respectively. Although the departure times are synchronized across all simulation approaches[23], the vehicle

[23] This is not universally the case in SUMO; significant congestion on a vehicle's departure link may introduce artificial delays in its network entry time.

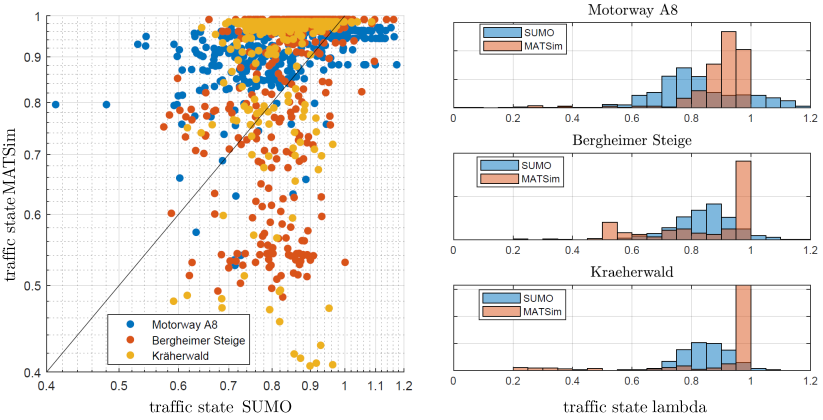

Figure 3.11: Inconsistencies in traffic state ($\lambda = 1$ for free-flow driving conditions, $\lambda = 0$ for blocked roads) between MATSim and SUMO across all investigated test cases. Left: Log-log plot of tracked vehicles. Right: Histograms of traffic states, separated by test case.

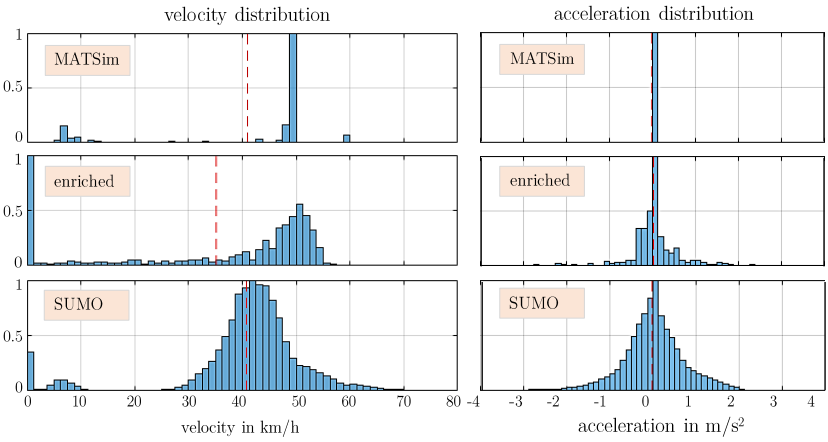

Figure 3.12: Scaled velocity (left) and acceleration (right) distributions for 359 simulated and tracked vehicles in the Bergheimer Steige test case with dt = 1s resolution. Red dashed lines indicate mean values.

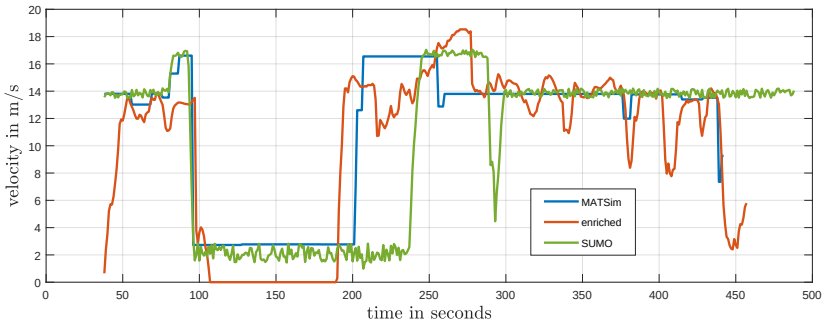

Figure 3.13: Time-dependent velocity profiles of a selected vehicle in the Bergheimer Steige test case, as simulated by different approaches.

experiences varying delays due to differences in traffic conditions, network distances, and the modeling of right-of-way rules at intersections.

The step-like nature of MATSim's profile is a result of its average link speed representation, while SUMO exhibits notable oscillations around the target velocity, likely attributable to driver imperfection (σ). However, the uniform amplitude and high frequency of SUMO's oscillations deviate from real-world driving dynamics. While these discrepancies could be mitigated with a better-tuned car-following model, further refinement is unnecessary as our primary focus is on future AD applications, where accurately replicating human driving behavior is not a priority.

3.3.4 Comparative analysis of simulated versus real-world driving cycles

In this section, we evaluate the accuracy of our simulations by comparing the simulated driving cycles for the Kräherwald test case with 22 GPS-tracked real-world cycles (refer to the FKFS test course detailed in Sec. 3.1.1). The analysis is restricted to the portion of the FKFS cycles that corresponds to the Kräherwald test case, as illustrated in the right panel of Figure 3.9.

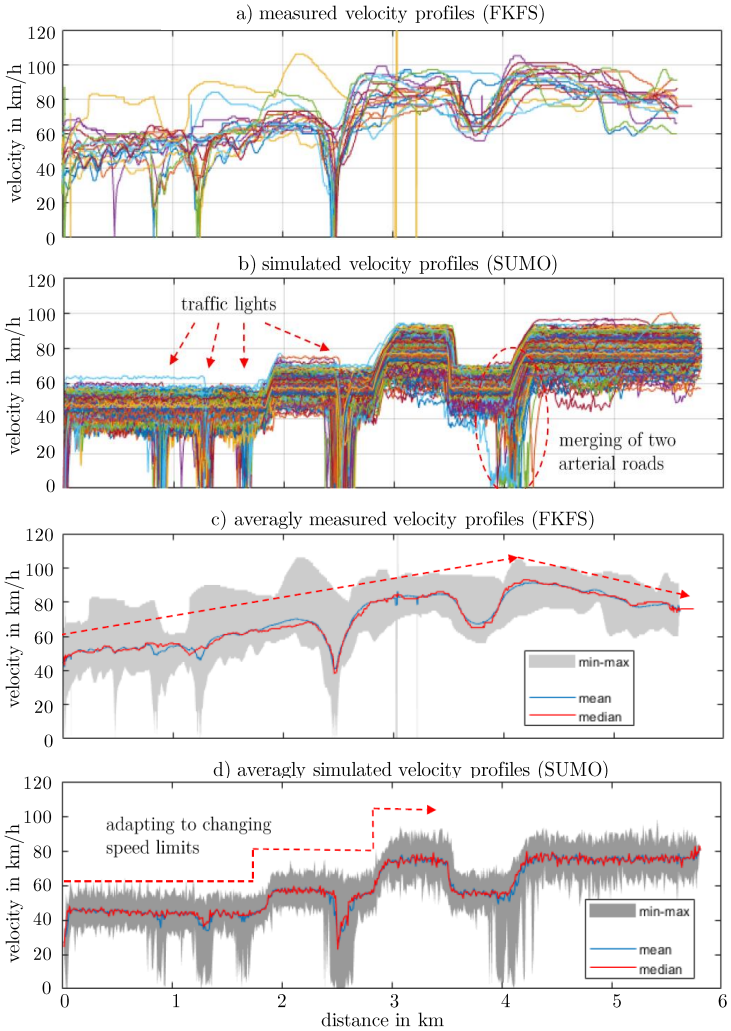

Figure 3.14: Stacked velocity-space profiles for 22 measured FKFS cycles (a) and 444 simulated SUMO vehicles (b) in the Kräherwald test case. Panels (c) and (d) present the minimum, mean, and maximum velocity values for both measured and simulated data at each point along the trajectory.

Spatial velocity profiles and their variations Figure 3.14 presents a comparison of the stacked velocity-space profiles for 22 real-world FKFS driving cycles (panel a) and 444 simulated SUMO vehicles (panel b) in the Kräherwald test case. Panels (c) and (d) display the minimum, mean, median and maximum velocity values for both measured and simulated data at each point along the trajectory. The trajectory, as shown in Figure 3.9, includes four successive traffic lights and a merge onto another arterial road, resulting in sudden velocity drops in both datasets (panels a and b). While the FKFS data represent free-flow driving conditions only, the SUMO simulation captures congestion and extended queues, particularly at traffic signals and intersections (panel b). Furthermore, real-world drivers typically adjust their speed in anticipation of upcoming speed limit changes (highlighted by the red dashed lines in panel c), whereas the simulation reflects these changes in a more step-like manner (panel d). Although this simplification may not adequately represent human driving behavior, it may be advantageous for autonomous vehicles, which could potentially adapt to speed limits in a similar fashion.

Vehicle-based performance metrics For further validation, we limit our comparison to simulated vehicles that exhibit traffic conditions analogous to those in the real-world FKFS data. As all FKFS driving cycles represent free-flow traffic, the analysis of congested scenarios remains infeasible. Table 3.6 reports the driving KPIs for a selected simulated vehicle in conjunction with

Table 3.6: Benchmarking of simulated vehicle KPIs against measured FKFS data under free-flow driving conditions ($\lambda \approx 1$). The simulated vehicle is identical (i.e., has the same vehicle ID, departure time, and route); however, its spatio-temporal trajectory was derived using three distinct methods (MATSim, enriched, SUMO). As real-world reference, we selected three randomly chosen FKFS vehicles.

	simulated cycles			measured FKFS cycles		
performance metrics	MATSim	enriched	SUMO	car 1	car 2	car 3
distance in m	5489	5460	5698	5951	5955	5942
average velocity in m/s	16.4	15.2	15.6	15.0	14.5	17.2
travel time in min	5.6	6.0	6.1	6.6	6.9	5.8
energy consumption in kWh/100km	20.1	18.3	18.8	18.8	22.0	19.6
traffic state λ	0.95	0.88	0.9	0.87	0.84	0.99

three randomly chosen FKFS vehicles. Overall, the KPIs demonstrate strong agreement under free-flow conditions, regardless of the method used for driving cycle extraction (MATSim, enriched, SUMO).

The velocity distributions under free-flow conditions reveal significant discrepancies. As illustrated in Figure 3.15 (left), real-world drivers, represented by FKFS cycles (red), generally exhibit higher driving speeds compared to those simulated in SUMO (blue). The SUMO simulations show velocity peaks at approximately 45 and 75 km/h, whereas the FKFS data peaks are observed around 62 and 82 km/h. Additionally, SUMO displays a higher frequency of near-zero velocities, indicative of frequent stops at traffic signals. This is further confirmed by the cumulative velocity distribution in Figure 3.15 (right), where the SUMO distribution shifts toward lower velocities, attributed to delays caused by traffic lights. In contrast, MATSim's velocity distribution (purple) shows poor alignment with FKFS data, likely due to its simplified queuing model. While the enrichment technique (yellow) mitigates some of these discrepancies, it continues to closely follow the trends observed in MATSim.

The validation of simulated driving cycles against FKFS measurements is constrained by two key factors: (i) the 22 measured cycles are statistically insufficient to represent driving behavior for the Kräherwald test case over a full day, and (ii) accurate validation of individual driving profiles necessitates

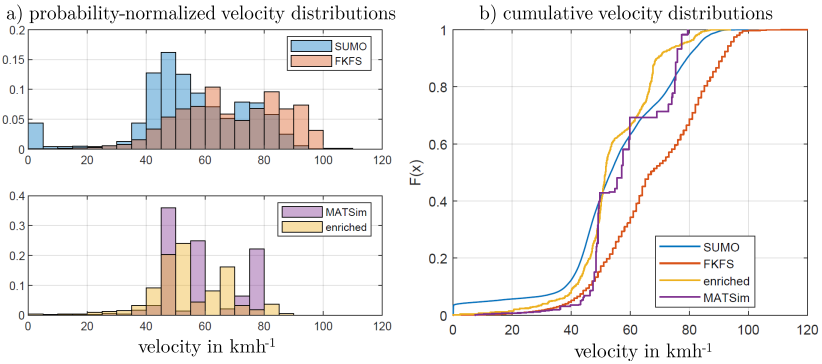

Figure 3.15: Probability-normalized velocity distribution (left) and cumulative velocity distribution (right) for free-flow driving conditions across all simulated (MATSim, enriched, SUMO) and measured (FKFS) data.

precise modeling of the ego-vehicle's environment, including surrounding traffic and traffic signals—capabilities that neither MATSim nor SUMO can fully achieve. Comprehensive environmental modeling would require the integration of radar and LIDAR data.

3.3.5 Discussion and implications for thesis scope

This section presents a summary of the quantitative results from the preceding sections and aligns them with the key requirements *(KR0-5)* outlined in Section 1.4. We seek to identify a suitable solution approach for SAEV drive cycle deduction and pinpoint areas warranting further research to address remaining gaps and challenges. We also discuss the feasibility and automation potential of a prospective MATSim-SUMO tool coupling for this purpose.

Given the uncertainty regarding the future role of SAEV fleets, we require an effective method to determine SAEV travel demand *(KR0)* in a scenario-based manner. SAEV fleets will be accepted if they prove more beneficial in daily use compared to other existing travel modes. Due to its agent-based nature, MATSim is particularly well-suited for this type of modeling, with the enrichment procedure also offering this capability by inheritance. Although SUMO, empowered by JADE, can perform agent-based simulations, its high level of detail and data requirements make it less efficient for repetitive, large-scale multi-modal simulations. Therefore, we favor MATSim-based approaches to meet *KR0*.

A similar reasoning applies to the city-wide deduction of SAEV driving patterns *(KR1)*. Due to the coarser granularity and lower data requirements of MATSim and its ability to cope with sampled population data, MATSim surpasses SUMO in terms of modeling effort, computation time, and scalability. SAEV fleet simulation involves numerous parameters, each significantly impacting SAEV mode adoption. Testing different charging, dispatching, or relocating strategies demands substantial computational power. Combined with the need for repetitive simulations to gauge SAEV travel demand, DRT simulations are particularly time-consuming, especially when tracking all fleet vehicles microscopically. Addressing these issues with purely microscopic traffic simulation is not practical, as its added level of sophistication further strains

simulation time. Here again, MATSim's runtime advantages and comprehensive DRT and EV features also benefit the enrichment approach.

For tracking individual vehicle states *(KR2)* and additional environmental data *(KR4)*, both MATSim and SUMO provide effective solutions. In particular, SUMO excels in this aspect, allowing for the querying of instantaneous vehicle states with high temporal resolution, down to millisecond intervals if needed. In contrast, MATSim's spatial-temporal resolution is coarser than that of SUMO. Its link-wise scope leads to less detailed time-series data, with metrics aggregated or static over road segments. This limitation is mitigated by enriched MATSim profiles, although these profiles become synthetic and do not reflect specific vehicle trajectories. With minute-wise drive cycle enhancement, longer downtime phases in MATSim, such as charging and extended idling periods, remain sufficiently consistent. However, shorter downtime durations, lasting only seconds, are overwritten with new driving behavior. Since MATSim lacks an accurate representation of stop-and-go patterns and intersection dynamics, overwriting these details is deemed acceptable for our purpose.

The core of this thesis focuses on deducing realistic SAEV speed profiles *(KR3)* that capture complex driving dynamics. Both MATSim and SUMO effectively capture aggregated trip statistics such as average velocity, traveled distance, and time, under similar traffic conditions. However, traffic conditions for a given vehicle can vary significantly between approaches due to differences in traffic dynamics modeling and network representation.

MATSim's simplified queuing model results in unrealistic velocity and acceleration profiles, rendering it unsuitable for automotive prototyping. Although the enrichment procedure produces velocity profiles that closely resemble real-world driving, its reliance on historical data presents a major drawback for autonomous driving applications. This data-driven approach lacks sensitivity to varying driving styles and secondary effects, such as platooning or V2X efficiencies. Conversely, SUMO enables the deduction of detailed speed profiles. While its driving dynamics may seem too artificial for realistic human driving, they are suitable for representing autonomous vehicles, which involve fewer stochastic modeling elements. Additionally, when integrated with

VANET[24] simulators, such as Veins[25], SUMO gains access to a powerful set of features for studying intelligent transport systems and (connected) autonomous driving. This integration allows for exploration of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications and their impacts on traffic flow, safety, and efficiency. Leveraging these functionalities through a SUMO-based approach offers clear advantages.

SUMO also excels in its extensive features for adjusting driving dynamics through the selection and parameterization of appropriate car-following and lane-changing models. Driving profiles deduced from SUMO are sensitive to actual traffic conditions and transport infrastructures, displaying a wide range of driving maneuvers. However, SUMO does not account for the reduced velocities human drivers adopt in narrow curves or on road gradients for comfort and vehicle power limitations. Addressing these shortcomings necessitates advanced vehicle simulation tools, such as GT-Suite or *CarMaker*[26].

While road slopes and their impact on energy consumption can be modeled in both MATSim and SUMO[27] in simplified manners, accurate height data is necessary. Unfortunately, non-proprietary data (such as OSM) often reflects the Earth's surface only and does not account for the real altitude of road tunnels, leading to incorrectly steep road gradients. While this issue may be neglected in flat areas like Berlin, it becomes problematic in hilly and tunnel-rich regions such as Stuttgart. When vehicle simulators use such deficient elevation profiles combined with corresponding speed profiles, the simulation controller frequently fails to achieve desired speeds in power-demanding situations (e.g., in instances where higher velocities coincide with steep road gradients), resulting in fatal simulation errors or significantly overrated energy consumption. As our focus progressively shifted to Berlin, we did not pursue this issue further, despite recognizing the need for improvement.

[24] VANET stands for Vehicular Ad Hoc Network, a network architecture used in intelligent transportation systems and wireless communication.

[25] Veins [198] combines the network simulator OMNeT++ [214] with the road traffic simulator SUMO.

[26] IPG CarMaker, https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/. Accessed: Sep. 29, 2024.

[27] See MATSim's EV modeling in Section 2.1.3 and [143, 126, 188, 187, 213, 116] for SUMO capabilities.

In summary, neither pure MATSim nor SUMO alone is suitable for deriving representative SAEV driving cycles, as compiled in Table 3.7. While MATSim excels in *KR0* and *KR1*, it lacks the spatial-temporal resolution and detail required for accurate driving dynamics *(KR3)* and time-series data *(KR2-4)*. Conversely, SUMO excels in detailed, microscopic simulations and modeling connected autonomous vehicles within intelligent transport systems. However, it faces challenges with large-scale simulations, especially when the travel demand for new mobility concepts is uncertain. Enhancing MATSim to improve detail and driving dynamics introduces limitations similar to those of microscopic traffic simulators, such as significantly increased computation time and modeling effort. The enrichment procedure enhances detail but relies on historic human-driven cycles, making it inflexible to emerging driving concepts. Additionally, the enrichment only partially mitigates MATSim's congestion modeling deficiencies. Thus, we prefer a model-based solution to a data-driven approach.

In terms of practical (large-scale) feasibility and automation capability *(KR5)*, the three simulation approaches exhibit significant differences. MATSim-based approaches excel with lower data demands and greater robustness against simulation errors and imperfect networks. In contrast, SUMO is less resilient. SUMO's highly detailed networks require precise data, including accurate junction information and traffic light positions. However, SUMO's autogenerated networks often suffer from issues due to the incomplete or inaccurate data from underlying OSM sources[28], and the complexities of the data can challenge default import functionalities[29]. This results in problems like faulty turning lanes, inaccurately placed traffic lights, inadequately configured junctions, and uncoordinated traffic light operations, with manual corrections being highly time-consuming. Furthermore, SUMO approaches are prone to artificial deadlocks, such as conflicting vehicle turns, which do not resolve naturally and require specialized solutions[30].

[28] Although proprietary map providers offer more detailed data, they also lack comprehensive traffic light information, such as cycle times and control methods (fixed or actuated), as this is not crucial for navigation. Thus, integrating multiple data sources does not fully resolve the issue.

[29] Experience has shown that adjusting import heuristics, such as the spatial parameters for aggregating neighboring junctions into a single cluster, can resolve certain network issues but may simultaneously worsen problems in other network areas.

[30] SUMO options such as `time-to-teleport` or `ignore-junction-blocker` may address some issues but necessitate extensive post-processing if the ego-vehicle is affected. A vehicle

Table 3.7: Evaluation of the suitability of different approaches for modeling detailed SAEV driving profiles and the potential effectiveness of a MATSim-SUMO coupling (✓ suitable, o limited suitability, - not suitable, () anticipatory only, not tested, ? uncertain)

	key requirements	pure MATSim	enriched MATSim	pure SUMO	MATSim-SUMO coupling
KR0	agent-based modeling of SAEV travel demand	✓	✓	-	✓
KR1	efficient SAEV fleet simulation	✓	✓	(o)	✓
	- EV range & charging constraints	✓	✓	(o)	✓
	- fleet dispatching & routing	✓	✓	(o)	✓
	- pricing strategy & service area	✓	✓	(o)	✓
KR2	extraction of vehicle states	o	✓	✓	✓
KR3	realistic velocity profiles				
	- human / autonomous driving	-/-	✓/o	o/(✓)	o/✓
	- reflecting traffic conditions and	o	o	✓	✓
	- divers transport infrastructures	-	-	✓	✓
	- platooning & C2X-communication	-	-	✓	✓
KR4	environmental data such as - height / speed limit / occupancy profiles	o/✓/✓	-/-/o	o/(✓/✓)	o/✓/✓
KR5	suitability & feasibility				
	- fulfillment of *KR0-4*	-	o	-	✓
	- data availability	✓	o	o	?
	- resilience to imperfect networks	✓	✓	-	?
	- automation capability	✓	✓	✓	?
	- computational demands	✓	✓	o	✓
	overall suitability	-	o	-	(✓)

Considering the key requirements *KR0-KR4*, coupling MATSim with SUMO appears to be the most effective approach for deriving representative SAEV drive cycles. Such a hybrid simulation approach leverages the strengths of both frameworks, where MATSim efficiently handles large-scale, city-wide SAEV simulations, and SUMO manages detailed, local traffic dynamics and vehicle interactions. However, the practical feasibility and automation potential of such a tool-coupling are still uncertain. While it capitalizes on the strengths of both MATSim and SUMO, it may also inherit their limitations, such as the extensive effort needed to set up SUMO's microscopic networks. To address these challenges, strategies must be developed to simplify SUMO's network setup, automate network adjustments and traffic light configurations, or reconcile SUMO's high data requirements with MATSim's less restrictive mesoscopic demands to improve SUMO's resilience to network imperfections.[31]

Another challenge in coupling MATSim with SUMO is their inherent incompatibility. Significant differences in traffic flow and network modeling raise concerns about whether these systems can be effectively integrated. Specifically, it is crucial to determine if SUMO can accurately incorporate and simulate the travel demand generated by MATSim without causing severe gridlocks in SUMO's more congestion-prone microscopic environment. To address this, we must analyze the discrepancies between the frameworks in detail, focusing on global network capacities, achievable traffic flows by junction type, and the effects of differing network representations and traffic dynamics on routing and traffic performance.

The complexity of this problem constitutes a significant academic challenge and will henceforth be the central focus of this thesis. In Part II, we detail the design of a multi-level simulation approach for deriving microscopic SAEV driving profiles from mesoscopic transport simulation. We begin with the design and implementation of a robust meso-micro network matching routine

positioned first at an intersection is automatically teleported to the next free road segment of its route if it remains stationary for an extended period. In contrast, when the junction blocker option is activated, vehicles do not account for the presence of other vehicles obstructing traffic flow within intersections.

[31] While enhancing SUMO's network setup with superior data import heuristics may seem straightforward, the task is complex. Despite our initial enthusiasm, we found that achieving this automation within the limited resources of this thesis is unfeasible. Scientific communities have grappled with these issues for over a decade, due to the inherent complexity of the problem.

in Chapter 4. This enables the automated population of SUMO networks from travel time-calibrated MATSim simulations (Chapter 5) and facilitates the aforementioned consistency assessments (Chapter 6). Based on the results, we will further refine our tool-coupling approach as necessary.

Part II

Design of a Multi-Level Simulation Approach for Deriving Microscopic Driving Profiles from Mesoscopic Transport Simulations

4 Automated matching of meso and microscopic network elements

In this chapter, we address the matching of road networks from meso- and microscopic traffic simulations, using MATSim and SUMO as representative tools. We begin with an introduction to graph theory in Section 4.1, providing the foundational concepts necessary for exploring network structures. In Section 4.2, we explore the intricacies and differences between meso- and microscopic network representations. Finally, in Section 4.3, we propose a method for the automated matching of network elements. The matching concepts introduced are applicable to other meso- and microscopic traffic simulation frameworks. However, when applied to different frameworks, adjustments tailored to the unique characteristics of each tool will be required to resolve discrepancies in network representations.

4.1 Fundamentals of graph theory and their MATLAB implementation

To establish a foundation for our network matching approach, we provide a concise introduction to graph theory and its significance in network analysis. Graph theory is a branch of mathematics that provides a systematic framework for modeling and analyzing relationships between objects. Its applications extend across diverse fields, including computer science, mathematics, social sciences, and biology.

At its core, graph theory examines the concept of a "graph," which consists of a set of "vertices" (or junctions, in our case) and "edges" (representing road segments). The edges define the pairwise connections between the vertices. Graphs can be classified into various types, such as weighted, bipartite, directed, or undirected, depending on the specific nature of the problem.

An *undirected graph* is a mathematical structure denoted as an ordered pair $G = (V, E)$, where V represents a set of vertices and $E \subseteq \{\{x, y\} \mid x, y \in V \text{ and } x \neq y\}$ is a subset of unordered pairs $\{x, y\}$, with x and y being distinct vertices in V [15]. In this context, the vertices x and y are referred to as the endpoints of the edge $\{x, y\}$. The term *multiple edges* describes cases where two or more edges exist between the same pair of vertices. When looped edges[1] are allowed, the definition of E extends to include cases where $x = y$.

In *directed graphs* (or digraphs), the edge set is defined as $E \subseteq \{\{x, y\} \mid x, y \in V^2 \text{ and } x \neq y\}$, where each edge has a direction, with x and y denoting the tail and head of the directed edge, respectively [15]. In this framework, the edge $\{y, x\}$ is considered the *inverse edge* of the directed edge $\{x, y\}$.

A *subgraph* refers to a smaller graph derived from a larger graph by selecting a subset of its vertices and the edges between them [63]. Formally, a graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if and only if $V' \subseteq V$ and $E' \subseteq E$.

In the context of GPS navigation and travel planning, weighted directed graphs are frequently utilized. In such graphs, each edge is associated with one or more numerical values known as *weights*. These weights typically represent parameters such as road length, travel time, or other associated costs.

The MATLAB library[2] provides a comprehensive set of tools for working with weighted directed graphs. In MATLAB, the representation of vertices (or nodes) and edges, along with their respective weight attributes, is managed through data structures referred to as `NodeTables` and `EdgeTables`. These data structures function as repositories for the relevant graph information.

In forthcoming work, the term *MATSim digraph* denotes a directed graph representation of the MATSim road network, while *SUMO digraph* refers to the analogous directed graph representation of SUMO's microscopic network. After parsing and converting both MATSim and SUMO network data into corresponding MATLAB digraphs, we employ various object functions to perform operations on these graph structures. These operations include tasks such as extracting subgraphs, removing isolated edges, conducting breadth-first

[1] A looped edge is an edge that connects a vertex to itself.
[2] "Digraph Documentation," MATLAB, https://de.mathworks.com/help/matlab/ref/digraph.html. Accessed: Sep. 29, 2024.

searches[3,4] [119] and visualizing the graph. Given its relevance to our matching routine, we next address the challenge of finding the shortest path between two nodes.

In network routing, the *shortest path problem*[5] refers to the task of identifying a path between two nodes that minimizes the total weight of the connecting road segments. Specifically, for undirected graphs, the shortest path problem can be formulated as follows [62]:

A *path* is a sequence of vertices denoted as $P = (v_1, v_2, \ldots, v_n) \in V \times V \times \cdots \times V$, where each vertex v_i is *adjacent* to v_{i+1}. Two vertices are considered adjacent if they share a common edge, which we denote as $e_{i,j}$, connecting v_i and v_j. For an undirected graph G and a real-valued weight function $f : E \longrightarrow \mathbb{R}$, the shortest path from vertex v_i to vertex v_n can be defined as the path $P = (v_1, v_2, \ldots, v_n)$ that, out of all possible paths of length n, minimizes the sum $\sum_{i=1}^{n-1} f(e_{i,i+1})$.

The Dijkstra algorithm [64] is a widely recognized method for efficiently finding the shortest path in weighted graphs with non-negative edge weights. The algorithm maintains a set of visited vertices and iteratively explores the unvisited vertices with the smallest tentative distances from the source vertex. In the following, we employ the MATLAB implementation of Dijkstra's algorithm to address shortest path problems.

Alternative approaches include the Bellman-Ford algorithm [14, 80] and the A* algorithm [91]. The Bellman-Ford algorithm handles graphs with arbitrary edge weights, including negative values, by leveraging dynamic programming. This ensures the shortest path is found even in the presence of negative weight cycles. In contrast, the A* algorithm [91] is a heuristic search method particularly suited for large graphs. It combines a cost function (the actual cost to reach a node) and a heuristic function (an estimate of the cost to reach the destination

[3] *Breadth-first search* [119] is an algorithm used for the systematic exploration of tree and graph data structures. It starts traversal from a given node and explores all neighboring nodes at the present depth level before moving on to nodes at deeper levels. This method contrasts with *Depth-First Search* [119], which explores a branch of the graph to its maximum depth before backtracking and exploring other branches.

[4] "BFsearch Documentation," MATLAB, https://de.mathworks.com/help/matlab/ref/graph.bfsearch.html. Accessed: Sep. 29, 2024.

[5] For further details on this topic, refer to [173].

from a node) to prioritize node exploration, efficiently examining the most promising paths first.

4.2 Preparatory steps for MATSim-SUMO network alignment

4.2.1 Differences in network representation

Assuming both MATSim and SUMO derive their network data from the same map provider (in our case, OSM), a key distinction between MATSim's mesoscopic network representation and SUMO's microscopic interpretation lies in the dimensionality of the network. As illustrated in the lower-left panel of Figure 4.1, MATSim network elements are one-dimensional, where *nodes* represent junctions in a point-like manner, and *links* denote road segments with linear appearances. In contrast, SUMO networks provide a more realistic representation with *junction* and *edge* objects that possess distinct shapes (see lower-right panel of Figure 4.1).

In the context of MATSim, each node is uniquely identified by a distinct identifier (ID) and is associated with corresponding XY-coordinates. Although MATSim can incorporate additional intersection-related features such as traffic lights or lane configurations [237, 236], traffic stream priorities are typically determined based solely on the outflow capacity Q_i of the links, as detailed in Section 2.5.1.

In contrast, link objects in MATSim have a more complex set of attributes. The *from* and *to* attributes specify the originating and terminating nodes, defining the geometric structure of the network. Additional attributes include link length, capacity, speed limit, the number of lanes, and permitted transportation modes. By default, the original OpenStreetMap road identifier and street type are also retained within the MATSim network file. It is important to note that all MATSim links are unidirectional. The structural composition of a MATSim `network.xml` file is as follows:

```
<network>
  <nodes>
    <node id="100222579" x="3512258.722172523" y="5402776.230694266" >
    </node>
  </nodes>
  <links capperiod="01:00:00" effectivecellsize="7.5" effectivelanewidth=
     "3.75">
    <link id="1" from="490652" to="75789739" length="30.52" freespeed="
     13.89" capacity="600.0" permlanes="1.0" oneway="1" modes="car" >
      <attributes>
        <attribute name="origid" class="java.lang.String" >600579017</
     attribute>
        <attribute name="type" class="java.lang.String" >tertiary</
     attribute>
      </attributes>
    </link>
  </links>
</network>
```

Listing 4.1: Structural composition of a MATSim `network.xml` file, illustrating an examplary `node` and `link` configuration. The example may not be internally consistent.

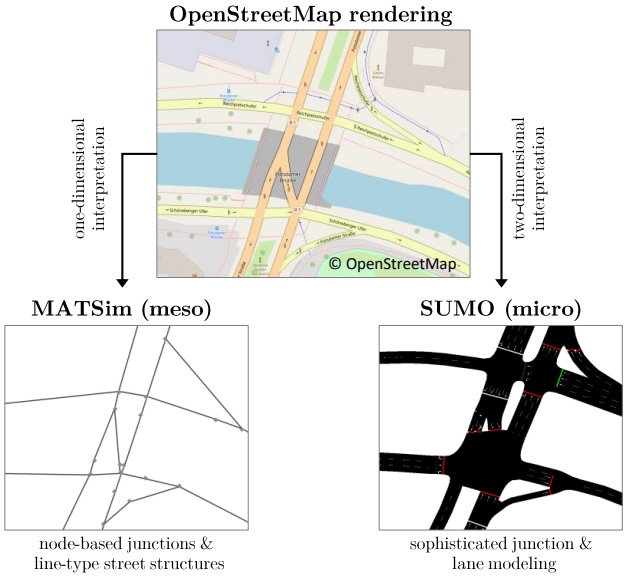

Figure 4.1: Visualization of OpenStreetMap network data interpretation by MATSim and SUMO: An exemplary case study of Potsdamer Brücke, Berlin.

Similar to MATSim, SUMO also employs unidirectional edges, with opposing edges sharing negative IDs. Each SUMO edge is comprised of *lane* objects characterized by attributes such as ID, disallowed transport modes, speed, length, and geometric shape. Junctions are two-dimensional and are defined by centroids with XY-coordinates, outlined by closed polygonal chains. "Internal links" standardize the connections between incoming and outgoing lanes within junctions.

Right-of-way rules, traffic light strategies (e.g., "actuated," "time-based," or "queue-based"), and other lane interactions are defined within junction objects or are assigned to dedicated objects (e.g., `tlLogic`, `connection`). If not explicitly specified in OSM data, these rules and strategies are heuristically determined during network import. Multiple junction objects can be consolidated into dedicated `roundabout` objects. An exemplary SUMO network configuration is illustrated as follows:

```
<net
    <edge id="-24751154#0" from="268986891" to="69343073" priority="11"
     type="highway.secondary">
        <lane id="-24751154#0_0" index="0" disallow="tram rail_urban"
     speed="13.89" length="14.89" shape="1772.20,651.44 1781.46,663.10"/>
        <lane id="-24751154#0_1" index="1" disallow="tram rail_urban"
     speed="13.89" length="14.89" shape="1769.69,653.43 1778.96,665.09"/>
    </edge>
    <tlLogic id="100224184" type="static" programID="0" offset="0">
        <phase duration="82" state="GGGG"/>
        <phase duration="3"  state="yyyy"/>
        <phase duration="5"  state="rrrr"/>
    </tlLogic>
    <junction id="100214531" type="priority" x="4063.81" y="1732.46"
     incLanes="-124815278#4_0 124815278#3_0" intLanes=":100214531_0_0
     :100214531_1_0" shape="4063.62,1735.66 4064.21,1729.28
     4063.79,1729.26">
        <request index="0" response="00" foes="00" cont="0"/>
        <request index="1" response="00" foes="00" cont="0"/>
    </junction>
    <connection from="-11283287" to="11283287" fromLane="0" toLane="0"
     via=":100225887_0_0" dir="t" state="M"/>
    <roundabout nodes="3426630575 48783203 48783226 48783230 48783231
     48783235" edges="22723674 335564471 34707121 34707122 38424364
     38424502"/>
</net>
```

Listing 4.2: Structural composition of a SUMO `net.xml` file, illustrating examples of `edge`, `tlLogic`, `junction`, `connection`, and `roundabout` configurations. The example may not be internally consistent.

4.2.2 Comparison and harmonization of network attributes

Harmonizing import heuristics and typemaps for network consistency
Our network matching procedure primarily relies on the congruence of network attributes between MATSim and SUMO. Consequently, we have observed a significant enhancement in the quality of matched network elements when default values during network import are harmonized.[6] This alignment is especially critical for the various *typemap* configurations, which specify the legal speed limits assumed by both MATSim and SUMO during network import from OSM for different road types. As presented in Table 4.1, substantial differences exist in these speed limits.

Table 4.1: Comparison of typemaps in MATSim and SUMO, outlining legal speed limits (in km/h) for various road types.

OSM	MATSim		SUMO	
road type	`highway defaults`	`VSP`	`default`	`urbanDe`
motorway	120	120	140	100
-link	80	80	80	60
trunk	80	80	100	50
-link	50	50	80	50
primary	80	80	100	50
-link	60	60	80	50
secondary	30	15	100	50
-link	30	15	80	50
tertiary	25	12.5	80	50
-link	25	12.5	80	50
unclassified	15	7.5	50	–
residential	15	7.5	50	–
living street	10	5	10	–

By default, MATSim utilizes the so-called `highway defaults` to preset legal speed limits for specific road types, unless explicitly overridden. To better reflect realistic urban traffic conditions, including delays from traffic lights, right-of-

[6] It is important to note that the actual traffic simulations in MATSim and SUMO utilize their respective best-practice parameter values. The adapted import settings are solely employed to align network elements and facilitate the network matching process.

way rules, and stop-and-go patterns, speed limits on secondary and below are reduced by 50% through the `VSP` setting (also referred to as `VSPAdjustments`). Additionally, primary road capacities are reduced by 33%, and secondary road capacities by 20%.

In contrast, SUMO's `default` freespeed values are notably higher, even after country-specific adjustments. To ensure network compatibility, we align SUMO's default speed limits with MATSim's `highway defaults`. The actual traffic simulations use `VSP` settings in MATSim and `urbanDe` settings in SUMO.

In addition to typemap settings, MATSim introduces non-integer lane numbers (e.g., 0.5 or 1.5 lanes), which influence link capacity – a concept non-existent in SUMO. Therefore, for the purposes of our matching routine (and not for actual traffic simulation), we round MATSim's lane numbers to the nearest integer values to ensure compatibility.

Refining network element attributes for effective matching Prior to initiating the network matching process, we enrich both the MATSim and SUMO digraphs by augmenting them with additional edge attributes (weights). The greater the distinctiveness of MATSim's link attributes, the easier it becomes to identify the corresponding SUMO edge based on similarity criteria. For the purpose of node matching, we rely exclusively on the x and y coordinates, which serve as appropriate matching attributes.

For edge matching, four primary attributes are employed as the basis for object matching: edge length, speed limit, number of lanes, and edge orientation. The first three attributes are natively available in both the MATSim and SUMO network files. The edge orientation, denoted as α, is computed in degrees using the spatial positions of the start and end nodes and their respective geographic coordinates (xy):

$$\alpha = \left(\text{atan2}(x_2 - x_1, y_2 - y_1) \times \frac{180^\circ}{\pi} + 90^\circ\right) \mod 360^\circ \tag{4.1}$$

Here, $dx = x_2 - x_1$ and $dy = y_2 - y_1$ represent the differences in longitude and latitude, respectively. Adding 90° adjusts the angle to fit the convention where north-facing edges are assigned an orientation of 0°, and subsequent cardinal

points are determined clockwise (East = 90°, South = 180°, West = 270°). The modulo 360° operation ensures that the angle is within the range $[0°, 360°)$.

In all shortest-path calculations, we rely exclusively on the free-speed travel time, denoted as $T = \frac{L}{V}$, as the relevant weight. Previous attempts that focused on link length (L) yielded unsatisfactory results, such as cases where curved highways in SUMO were erroneously matched with parallel exit and access ramps in MATSim, as illustrated in Figure 4.2. By prioritizing free-speed travel times, our approach mitigates this suboptimal matching behavior and better accounts for highway hierarchies through the free-speed parameter (V).

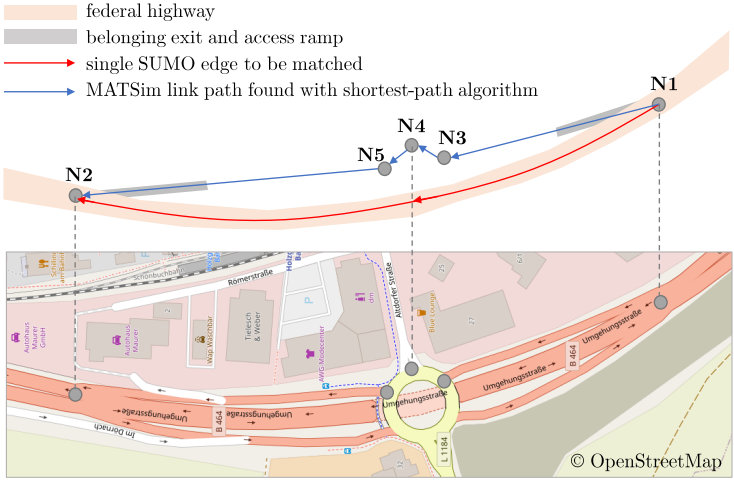

Figure 4.2: Shortest path search relying on link length L as the decision criterion, illustrated for the German federal highway B464 exit Holzgerlingen. Instead of matching with its corresponding MATSim highway, the SUMO highway (red line) is erroneously matched with an off-highway link path alternative (blue lines) due to the shortest path length criterion. To preserve highway hierarchy and improve accuracy, we recommend using free-speed travel time T as the decision attribute.

Figure 4.3 presents probability-normalized attribute distributions for MATSim and SUMO, covering various node and edge characteristics. The Berlin metropolitan area, as defined in Table 4.2, serves as the geographic region of interest for this analysis. Although adjustments to link speed and the number of lanes have resulted in a relatively good alignment of these attributes, notable

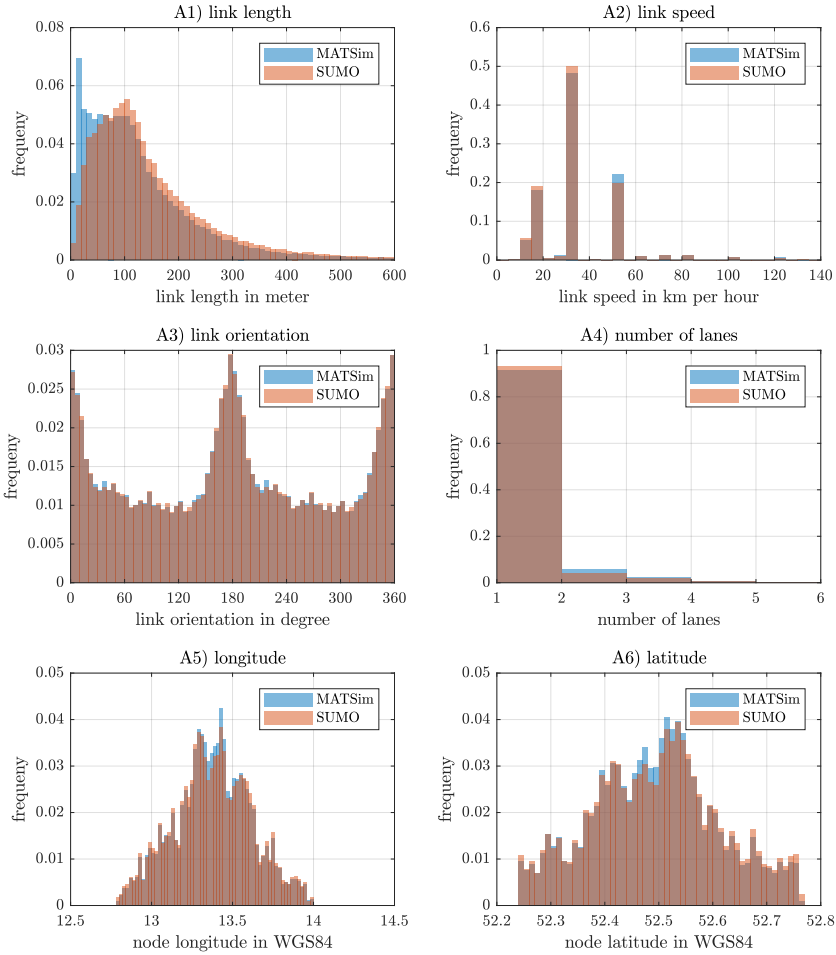

Figure 4.3: Probability-normalized distributions of node and edge attributes in MATSim (blue) and SUMO (red) for the Berlin area, as defined in Table 4.2.

Table 4.2: Network characteristics of the Berlin metropolitan area, defined by coordinates 12.789459, 52.762061 (upper-left) to 14.000702, 52.240415 (lower-right), for different network import options in MATSim and SUMO. In MATSim, the `VSPAdjustments` reduce speed limits on secondary and lower-class links by 50%, primary road capacities by 33%, and secondary road capacities by 20%. SUMO networks without "internal links" suppress lane-level connections within intersections, causing vehicles to traverse junctions without detailed simulation. To prevent systematic shortening of route lengths, edge lengths are measured as the distance between junction centers rather than external junction boundaries.

	MATSim		SUMO	
	- `VSP` settings -		- internal links -	
metric	with	w/o.	with	w/o.
no. nodes	82221		62726	
no. edges	192020		157949	
total length in km	26476		24658	26857
edge length in m	137.9		156.1	170.1
∅ edge speed in kmh^{-1}	25.0	33.0	39.8	

discrepancies remain in link length. Differences are particularly significant for edges shorter than 100 meters, while they are less pronounced for edges longer than 100 meters.

The higher prevalence of shorter links in MATSim can be attributed to two primary factors. Firstly, MATSim introduces new nodes during network import whenever there is a significant change in road curvature, whereas SUMO generates nodes only in response to changes in link attributes. Consequently, MATSim has a higher proportion of shorter links compared to SUMO, despite both networks having similar total lengths, as detailed in Table 4.2. Specifically, for the Berlin metropolitan area, MATSim includes 31% more nodes and 22% more links than SUMO.

Secondly, SUMO consolidates multiple nodes into a single junction ("cluster"), with the cluster's center typically located at the geometric center of the consolidated shape. This approach in SUMO excludes short node-to-node edges (road segments that would have connected the nodes now clustered), replacing them with "internal links" (lanes within an intersection). This design choice leads to a reduced number of short edges compared to those present in MATSim.

SUMO networks *without* internal links suppress lane-level connections within intersections, causing vehicles to traverse junctions without detailed simulation. To prevent systematic shortening of route lengths, edge lengths are measured as the distance between junction centers rather than external junction boundaries. Since this approach closely aligns with MATSim's point-to-point representation of links, we utilize SUMO networks *without* internal links throughout the matching procedure. However, we *enable* "internal links" during simulation to accurately capture driving behavior within junctions.

4.2.3 Automated vertex offset correction

Both network representations in MATSim and SUMO rely on a planar spatial reference system, which involves modeling the Earth's surface as flat. In this standardized Cartesian coordinate system, all spatial positions are defined relative to an arbitrary origin point along orthogonal x and y axes, approximately aligned with the cardinal directions. Various map projections are available for transforming the Earth's inherently curved surface into a planar representation.

For instance, our SUMO networks use the *Universal Transverse Mercator* (UTM) projection, specifically 'UTM33' with an additional offset for Berlin region. The UTM system divides the Earth into zones denoted by numbers and letters, strategically positioned to minimize distortion within each zone. This approach mitigates challenges associated with representing a spherical or ellipsoidal surface on a flat map.

In contrast, the Open Berlin Scenario [234] in MATSim uses the geodetic *Gauss-Krüger* (GK) projection, specifically 'GK4' for the Berlin region, to represent its typically larger network areas on a two-dimensional plane. In this approach, the Earth's curved surface is approximated as an ellipsoid, and geographic coordinates (latitude and longitude) are transformed into Cartesian form, comprising easting and northing values.

To facilitate network attribute comparisons and spatial alignment, we reverse these map projections by converting planar coordinates back into a geodetic reference system, where distances and directions are measured directly on the

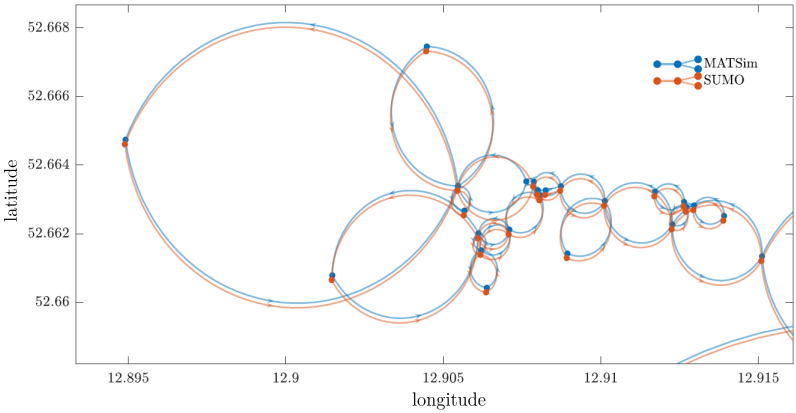

Figure 4.4: MATLAB digraph representation of an exemplary network excerpt in Berlin revealing a minor coordinate offset in both longitudinal and latitudinal direction.

Earth's curved surface. For this transformation[7], we employ the *World Geodetic System 1984* (WGS84) as the reference system.

After performing the coordinate transformation, a minor node offset remains between MATSim and SUMO, as shown in Figure 4.4. Upon visual inspection using OSM as reference, it appears that MATSim coordinates are generally accurate. To systematically address this issue, we recalibrate all SUMO coordinates against a sample of reference checkpoints, specifically nodes with IDs that are present in both MATSim and SUMO (approximately 70% of the total nodes). The presence of identical node IDs indicates that their OSM IDs have not been altered by import heuristics and are, therefore, reliable.

The observed spatial offsets for this subset of nodes are not constant. Panel A of Figure 4.5 exemplifies the longitudinal offsets of SUMO nodes relative to their MATSim counterparts. While the longitudinal coordinates match closely on the left side of the map excerpt, deviations increase substantially toward the

[7] We used the `Geodetic Toolbox` (https://www.mathworks.com/matlabcentral/fileexchange/15285-geodetic-toolbox) by Mike Craymer (2019), MATLAB Central File Exchange. Accessed: Jun. 11, 2019.

right. Combined longitudinal and latitudinal offsets result in spatial deviations ranging from 10 meters to 24 meters (see Figure 4.5, panel C).

Based on the scattered longitudinal offsets in SUMO (Fig. 4.5, panel A), we construct an auxiliary plane[8] (Fig. 4.5, panel B) onto which all SUMO nodes are projected. The longitudinal coordinates of the SUMO nodes are then adjusted according to the offsets indicated by this plane. A similar method is applied to correct the latitudinal offsets. This process effectively eliminates the offset between corresponding MATSim-SUMO nodes, reducing the residual discrepancies to negligible levels (Fig. 4.5, panel D).

4.3 Mathematical procedure for automated node and edge matching

4.3.1 Scoring algorithm for evaluating network element compatibility

Our objective is to match all SUMO edges and nodes with their potential counterparts in MATSim, based on the similarity of their attributes. To quantify this, we define a scoring metric, denoted as Z^{β}, where $0 \leq Z^{\beta} \leq 1$, for each network element $\beta \in \{\text{node}, \text{edge}\}$. Here, a score of 0 indicates no match (i.e., significant discrepancies in attributes), while a score of 1 denotes a perfect match, where all attributes of the node or edge are fully aligned.

Let $A^{\text{edge}} = \{\text{length}, \text{speed}, \text{lanes}, \text{orientation}\}$ represent the set of edge attributes. For a specific SUMO edge $s \in S$, where S denotes the set of SUMO edges, let $\alpha_s \in A^{\text{edge}}$ represent a specific attribute of edge s. Similarly, let $\tilde{\alpha}_m$ represent the corresponding attribute of a MATSim edge $m \in M$, where M is the set of MATSim edges. The edge-specific score $Z^{\text{edge}}_{s,m}$, which evaluates the similarity between SUMO edge s and MATSim edge m, is defined as follows:[9]

[8] Utilizing the `Gridfit Toolbox` (https://www.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit) by John D'Errico (2019), MATLAB Central File Exchange. Accessed: Jun. 27, 2019.

[9] For the attribute "orientation," additional adjustments are applied to ensure that angular differences are limited to a maximum of 180°.

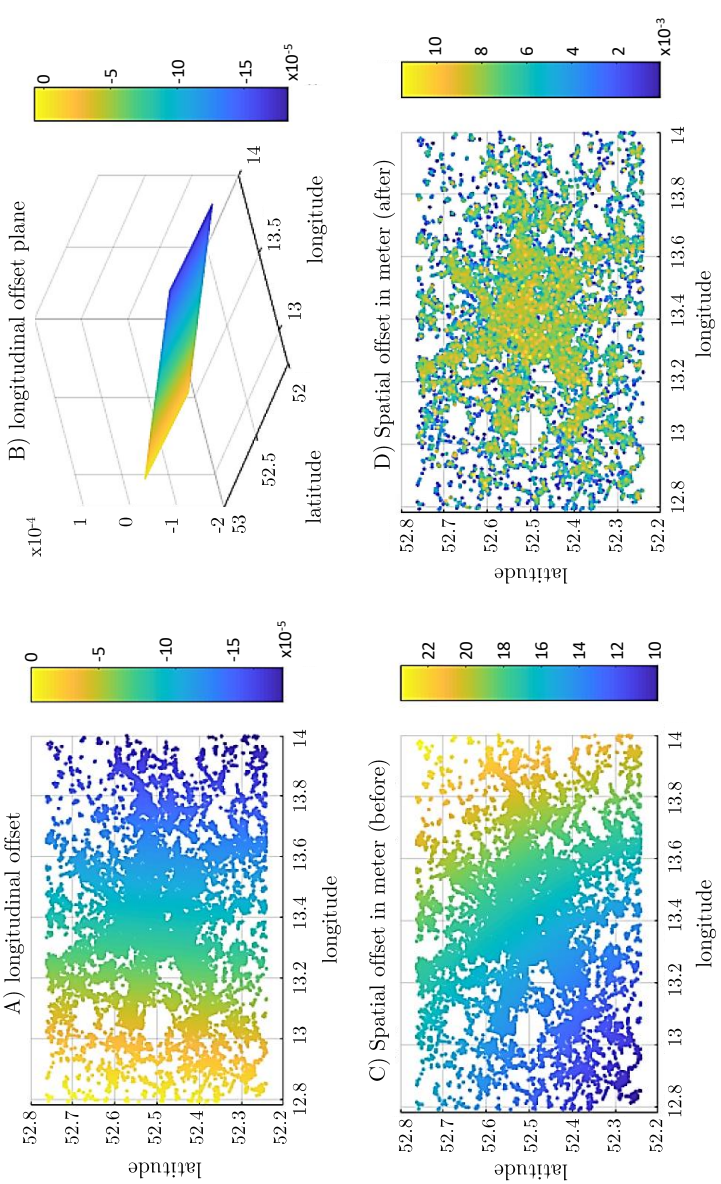

Figure 4.5: Spatial offsets between corresponding MATSim-SUMO nodes and their correction. Based on the scattered longitudinal offsets in SUMO (Panel A), an auxiliary plane was constructed (Panel B), onto which all SUMO nodes were projected. The longitudinal coordinates of SUMO nodes were then adjusted according to the offsets indicated by this plane. A similar procedure was applied to correct the latitudinal offsets. While pre-correction spatial deviations ranged from 10 meters to 24 meters (Panel C), the post-correction deviations were negligible (Panel D).

$$Z_{s,m}^{\text{edge}} = \sum_{\alpha \in A^{\text{edge}}} \omega_{\alpha}^{\text{edge}} \times \underbrace{\left(1 - \min\left(\frac{|\tilde{\alpha}_m - \alpha_s|}{\alpha_s}, 1\right)\right)}_{Z_{\alpha}^{\text{edge}}} \tag{4.2}$$

$$\sum_{\alpha \in A^{\text{edge}}} \omega_{\alpha}^{\text{edge}} = 1 \tag{4.3}$$

Here, $\omega_{\alpha}^{\text{edge}}$ represents the weight factor associated with the attribute $\alpha \in A^{\text{edge}}$. The term $\left(1 - \min\left(\frac{|\tilde{\alpha}_m - \alpha_s|}{\alpha_s}, 1\right)\right)$ accounts for the relative difference between the SUMO and MATSim attributes and is referred to as the sub-score Z_{α}^{edge}. This relative difference is normalized to ensure that the score remains within the defined range. If the difference between $\tilde{\alpha}_m$ and α_s exceeds 100% of α_s, the minimum function ensures the α-specific sub-score is set to zero. This prevents large discrepancies in sub-scores Z_{α}^{edge} from disproportionately affecting the overall score $Z_{s,m}^{\text{edge}}$.

Similarly, the node-specific score Z^{node} is calculated using the node attribute set $A^{\text{node}} = \{\text{longitude, latitude}\}$ in the same manner.

4.3.2 Methodology for optimizing attribute weights

With the general scoring function defined, the next step involves the parameterization of the attribute-specific weight factors ω_{α}^{β}. Rather than assigning equal weights to all attributes $\alpha \in A^{\beta}$, we adopt an approach based on the principle that attributes with greater similarity in their probability-normalized distributions between MATSim and SUMO should receive higher weight. This principle is illustrated in Figure 4.3, which demonstrates that, intuitively, the edge attribute orientation should be assigned greater weight in network matching, while the attribute link length should play a comparatively smaller role.

To quantify the similarity between two distributions, we initially considered using the Kolmogorov-Smirnov (KS) divergence [92, 184]. The KS divergence measures the maximum vertical deviation between two cumulative distribution

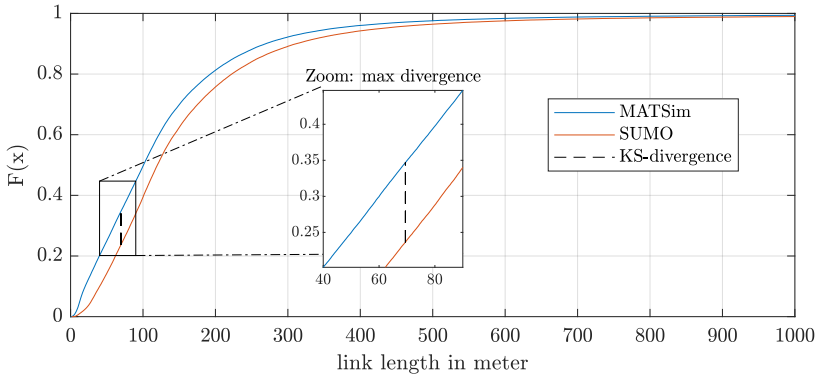

Figure 4.6: Visualization of the Kolmogorov-Smirnov (KS) divergence based on the cumulative distribution functions for link length in MATSim (depicted by the blue line) and SUMO (illustrated by the red line).

functions, as illustrated in Figure 4.6. Using cumulative distributions offers the advantage of reduced sensitivity to binning effects. However, we did not employ the KS test in this case because it is applicable only to continuous variables[10] and is less sensitive to variations in the tails of the distribution, where vertical separation is less pronounced.

Instead, we use the Kullback-Leibler (KL) divergence [124, 125] to quantify the similarity between our empirical distributions, and thereby derive the appropriate weight factors. The KL divergence measures the extent to which one probability distribution deviates from another. Let M and S represent two attribute distributions within the same probability space X. For discrete distributions, the KL divergence is defined as [41]:

$$D_{\mathrm{KL}}(M||S) = \sum_{x \in X} M(x) \log \left(\frac{M(x)}{S(x)} \right). \tag{4.4}$$

[10] Among the six distinct attributes associated with nodes and edges, two are discrete: "edge speed" and "number of lanes."

In the case of continuous random variables, the KL divergence is represented as an integral:

$$D_{\mathrm{KL}}(M||S) = \int_{-\infty}^{\infty} m(x) \log\left(\frac{m(x)}{s(x)}\right) \mathrm{d}x. \tag{4.5}$$

Here, $m(x)$ and $s(x)$ represent the probability density functions of the distributions M and S, respectively. It is important to emphasize that the Kullback-Leibler (KL) divergence is always non-negative, with $D_{KL}(M||S) = 0$ if and only if $M = S$. When the probability of an event is high in distribution M, but low in distribution S, the KL divergence becomes significantly large, indicating substantial deviation between the two distributions [41]. Additionally, it is important to note that the KL divergence is asymmetric, as shown by the following inequality:

$$D_{\mathrm{KL}}(M||S) \neq D_{\mathrm{KL}}(S||M). \tag{4.6}$$

Figure 4.7 illustrates the characteristic asymmetry observed when applying the KL divergence to all six node and edge attribute distributions. In this context, the KL divergence (represented by the red dashed line) can be interpreted as the cumulative result of the discrete function values. Specifically, the positive contributions (indicated by blue bars) are partially offset by negative contributions (represented by red bars), leading to the overall divergence between the distributions.

To assign weight to the edge attributes $\alpha \in A^{\mathrm{edge}}$, we first compute the relative KL divergence for each attribute:

$$D_{\mathrm{KL}}^{\mathrm{rel},\alpha} = 1 - \frac{D_{\mathrm{KL}}(M||S)_{\alpha}}{\sum_{\alpha' \in A} D_{\mathrm{KL}}(M||S)_{\alpha'}} \tag{4.7}$$

In this formula, $D_{\mathrm{KL}}(M||S)_{\alpha}$ represents the Kullback-Leibler divergence between the probability distributions M (MATSim) and S (SUMO) for the edge attribute α. The set A refers to the collection of all edge attributes, A^{edge}, which includes length, speed, lanes, and orientation. The denominator $\sum_{\alpha' \in A} D_{\mathrm{KL}}(M||S)_{\alpha'}$ is the total sum of KL divergences across all edge

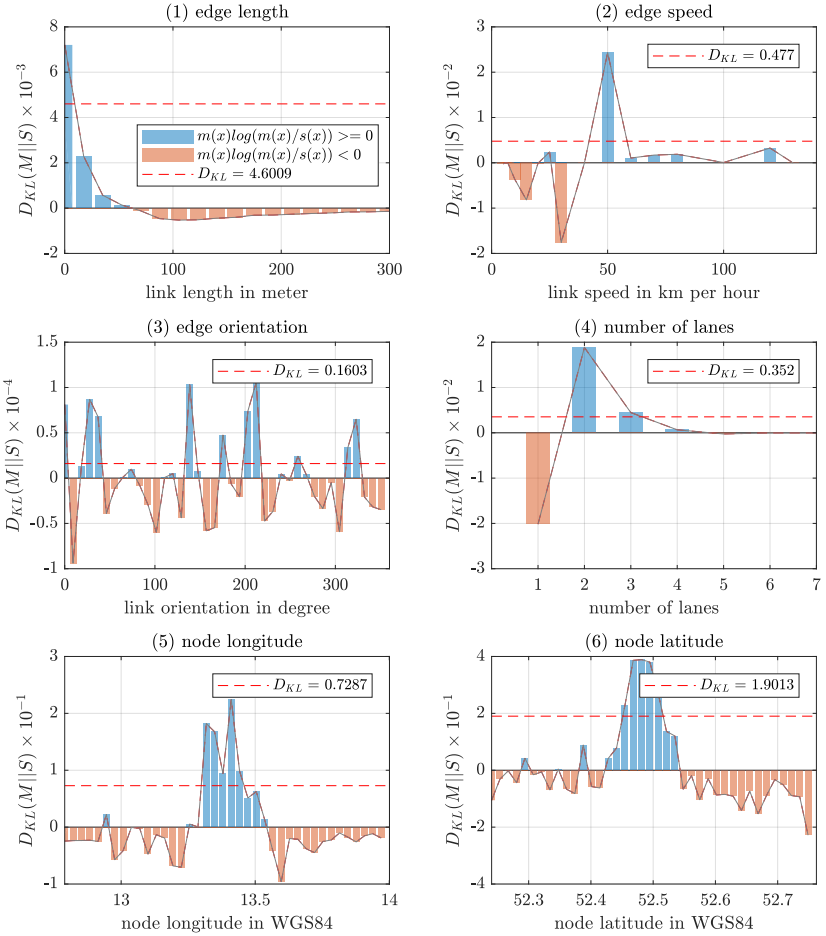

Figure 4.7: Visualization of the KL-divergence (indicated by the red-dashed line) for the corresponding node and edge attributes within a sample map excerpt of MATSim and SUMO. To enhance clarity, positive contributions are represented in blue, while negative contributions are denoted in red.

attributes in the set A^{edge}. The subtraction from 1 is applied to convert the KL divergence into a similarity score, so that a smaller divergence indicates a greater similarity between distributions, which in turn results in a higher relative weight for that attribute.

Next, we normalize these relative KL divergences to obtain the attribute-specific weight factors:

$$\omega_{\alpha}^{\text{edge}} = D_{\text{KL}}^{\text{rel, norm},\alpha} = \frac{D_{\text{KL}}^{\text{rel},\alpha}}{\sum_{\alpha' \in A} D_{\text{KL}}^{\text{rel},\alpha'}}. \tag{4.8}$$

This step is essential to ensure that the sum of all weight factors for the edge attributes equals 1, making the weights comparable across attributes. The term $D_{\text{KL}}^{\text{rel},\alpha}$ represents the relative KL divergence of a specific edge attribute $\alpha \in A^{\text{edge}}$, which is then normalized by dividing it by the sum of all relative KL divergences across the attribute set A^{edge}.

The result of this equation, $\omega_{\alpha}^{\text{edge}}$, represents the weight factor assigned to the edge attribute α, which quantifies its relative importance in the overall network matching process. Table 4.3 provides a comprehensive summary of the weight factors computed for all edge and node attributes.

Table 4.3: Calculated KL-divergences $D_{KL}(M||S)$ and weight factors ω_{α}^{β} for all edge and node attributes. The cumulative weights of the respective node and edge attribute sets are normalized to sum to one. This normalization ensures that the total contribution of all attributes within each set – whether node or edge – remains balanced, allowing for a proportional influence of each attribute in the matching process. Attributes labeled as "c" represent those of continuous type, while attributes labeled as "d" denote discrete variables.

attributes $\alpha \in A^{\beta}$	edge attributes $\alpha \in A^{\text{edge}}$				node attribute $\alpha \in A^{\text{node}}$	
	length	speed	orientation	no. lanes	longitude	latitude
types of data	c	d	c	d	c	c
$D_{KL}(M\|\|S) \times 10^{-2}$	0.460	0.477	0.002	0.352	-	-
$D_{KL}(M\|\|S)_{\alpha}^{\text{rel}}$	0.643	0.631	0.998	0.727	-	-
ω_{α}^{β}	0.214	0.211	0.333	0.242	0.5	0.5
$\sum_{\alpha \in A} \omega_{\alpha}^{\beta}$	1				1	

In the case of edge matching, edge orientation is assigned the highest weight, reflecting its strong influence on the similarity between MATSim and SUMO networks. Conversely, edge speed and length receive considerably lower weight factors, indicating their comparatively lesser role in the matching process. This distribution of weights aligns well with the visual patterns observed in Figure 4.3.

For node attributes $\alpha \in A^{\text{node}}$, a different approach is adopted. We assign equal weights to longitude and latitude, as both attributes are inherently intertwined in defining node locations, and treating them separately would undermine the integrity of the node matching process. Thus, $\omega^{\text{node}}_{\text{longitude}} = \omega^{\text{node}}_{\text{latitude}} = 0.5$, ensuring that both attributes contribute equally to node similarity.

4.3.3 Detailed procedure for network element matching

Although both MATSim and SUMO use the same source (OSM) to build their networks and generally preserve the original OSM object IDs for nodes and edges where possible, relying solely on matching elements by identical IDs has proven ineffective. While nearly 70% of nodes can be matched using this trivial approach, very few edges can be matched. Table 4.2 highlights that MATSim networks contain significantly more edges (+22%) and nodes (+31%) than SUMO, despite having the same road granularity. Previous research [209] has shown that SUMO networks can be imported from MATSim, but this method is also impractical due to MATSim omitting key data that SUMO requires. Forcing MATSim to retain this additional data (e.g., lane information, junction shapes, and traffic signaling) complicates network file management due to increased size. Consequently, we adopted the similarity-based network element matching approach, developed throughout this chapter, with the detailed procedure outlined below.

The matching procedure begins with a breadth-first search traversal of the MATSim digraph[11]. Each MATSim and SUMO network object is matched sequentially using the scoring function specified in Equation 4.2. Successful matches with a score $Z^{\beta} > 0.8$ are recorded in dictionaries that catalog the IDs of the corresponding MATSim and SUMO objects along with their achieved

[11] defined as MATLAB's graph-based representation of the MATSim network

scores. A separate dictionary is maintained to document rejection codes for objects that could not be matched. It is noteworthy that not all MATSim network objects will necessarily have suitable counterparts in SUMO. Nevertheless, this limitation does not hinder the tool coupling process, as our traffic assignment methodology, described in Chapter 5, effectively handles incomplete path sequences for ego vehicles. The matching procedure comprises five distinct techniques:

Part A – Clustered nodes The initial phase involves matching all SUMO cluster nodes with their corresponding MATSim counterparts. A cluster node aggregates multiple SUMO network nodes into a complex junction, often with shared traffic lights, and is identified by a name prefixed with "cluster_" followed by an enumeration of the merged node IDs, which align with MATSim node IDs. After successfully matching all cluster nodes, we identify MATSim links within the boundaries of these SUMO cluster nodes. Since these internal links do not have direct counterparts in SUMO (represented by lane "connections" rather than edges), they are assigned a match value of NaN[12]. In this step, approximately 10% of MATSim nodes and 4% of MATSim network links are successfully matched.

Part B – Non-clustered nodes This step involves identifying all remaining MATSim nodes whose node IDs are also present in the SUMO digraph. MATSim nodes not found in the SUMO digraph (approximately 20%) are assigned a match value of "NaN", indicating that a viable node counterpart in SUMO does not exist. This completes the node matching process.

Part C – Simple edges This phase focuses on matching less complex links where each SUMO edge corresponds to a single MATSim link, rather than a sequence of links. We first identify all unmatched MATSim links with viable start and end nodes[13]. These nodes allow us to extract subgraphs (see Sec. 4.1) from both MATSim and SUMO digraphs, including all connecting links. This subgraph-based method is particularly effective as it also identifies opposing links with reversed orientations and multi-edges[14]. By leveraging the reduced

[12] "NaN" stands for "Not a Number." It is a special value used in computing to indicate that a value is missing, invalid, or cannot be calculated.

[13] Viable or valid MATSim nodes have a high-fidelity counterpart ($Z^{\beta} > 0$) in the SUMO network.

[14] Multi-edges are parallel links with identical start and end nodes, but differing in length. They should not be confused with path sequences involving intermediate nodes.

`EdgeTables` (see Sec. 4.1) of these subgraphs, we can match edge objects based on attribute similarity. In this process, an additional 73% of MATSim links are successfully matched, resulting in a cumulative match rate of 77% for all MATSim links.

Part D – Complex Path Sequences This module tackles the more complex task of matching entire edge paths in MATSim with their corresponding counterparts in SUMO, while also handling branched path-finding solutions. Figure 4.8 demonstrates this process, highlighting the challenges of network matching when there are significant differences between meso- and microscopic network topologies (upper and middle panel).

We begin by identifying MATSim links with a valid start node. For example, the blue root node `26984647` in Figure 4.8 (upper panel) exists in SUMO as a cluster node (middle panel). The goal is to match the MATSim link `{26984647,248564486}`. However, the target node `248564486` (blue diamond in the upper panel) does not exist in SUMO.

For MATSim links lacking a valid end node, we apply a breadth-first search on the MATSim digraph to explore the (branched) successors of the target node (`248564486`), as illustrated by the tree structure in Figure 4.8 (bottom panel). Exploration terminates when either a viable SUMO counterpart (a "border node", highlighted in red) is identified or the branch is going to join a previously discovered branch in the next exploration step (a "branch joiner", highlighted in yellow).

We then construct a MATSim subgraph from all the explored MATSim nodes in the tree structure (bottom panel) to reduce the complexity. For each ordered pair of (root node, border node), we apply the shortest-path algorithm on the SUMO digraph to identify the corresponding SUMO edge paths. The newly explored edges and nodes form a new SUMO subgraph, which is then compared with the MATSim subgraph. Scoring is conducted not on individual edges but on the entire path, by summing length attributes and averaging length-normalized speed, orientation, and lane attributes.

For example, as shown in Figure 4.8, the path to the red "border node I" (`723176262`) results in a successful match. This implies that all MATSim links defined by the node sequence `26984647,248564486,1722013566, 1722013583,723176262` (upper panel) are successfully matched with the SUMO edge `{26984647,723176262}` (middle panel). In contrast, the path

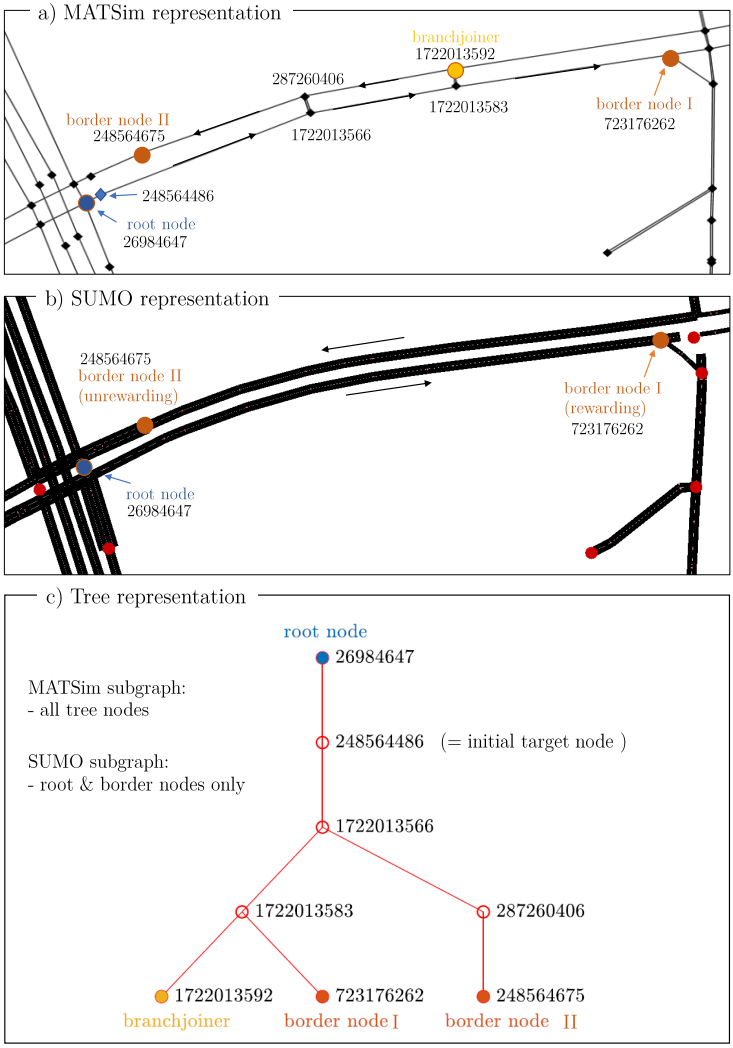

Figure 4.8: Exemplary depiction of the path finding procedure described in Part D of our matching routine. A chosen network excerpt is receptively illustrated in MATSim network visualization (top panel), SUMO network rendering (middle panel) and in MATLAB tree representation (bottom panel). For further explanation please refer to the main text.

to "border node II" (`248564675`) does not yield a match due to significant discrepancies in path attributes. Specifically, while the MATSim path takes a shortcut through nodes `1722013566` and `287260406` (upper panel), the corresponding SUMO path (middle panel) must traverse the major cluster junction in the right section of the image before looping back to `248564675`, resulting in a low score. By the end of PART D, 99% of all MATSim links are successfully matched.

Part E – Residual handling In the final stage of the matching procedure, any remaining unmatched MATSim links are assigned a value of NaN, indicating the absence of a corresponding counterpart in SUMO. This completes the matching process.

4.3.4 Goodness of fit

Finally, we evaluate the quality of the matching outcomes. For the Berlin area, as detailed in Table 4.2, we successfully matched over 82,000 MATSim nodes and 192,000 links. Figure 4.9 presents a flowchart depicting the edge-based scores (Z^{edge}) achieved across different matching stages (on the left axis) throughout the process, with corresponding scores shown on the right axis.

The majority of link matches, approximately 90%, were achieved in both Part C and Part D, with most scoring above 0.9 in Z^{edge}. Only about 4.3% of matches received scores below 0.8. Additionally, 5.7% of MATSim edges were matched with NaN, indicating that, as illustrated in the top and middle panels of Figure 4.8, certain SUMO counterparts effectively do not exist. Overall, the high fidelity of the matching results offers a solid foundation for our forthcoming tool-coupling efforts, particularly the traffic assignment procedure discussed in Chapter 5 and the MATSim-SUMO consistency assessments detailed in Chapter 6.

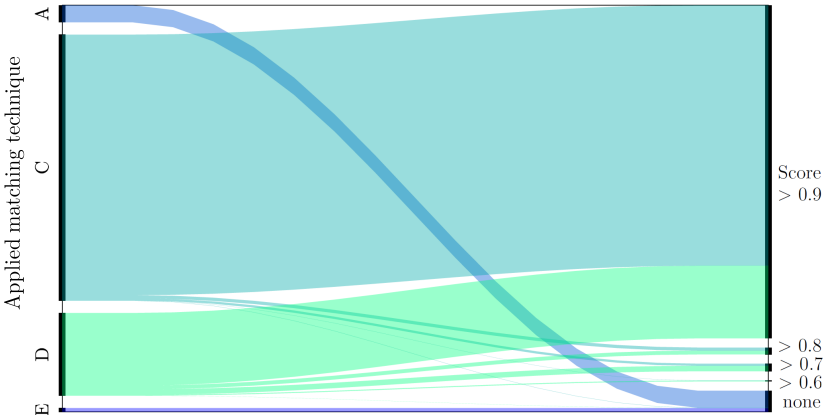

Figure 4.9: Flowchart depicting the scoring outcomes for all matched MATSim links. The chart categorizes these scores (on the right axis) according to the respective matching technique, labeled from Part A to E, displayed on the left axis. Note that Part B, while included in the overall process, did not contribute to edge matching, and therefore no data is available for this section.

5 Automated travel demand transfer across dynamic meso-micro network boundaries

This chapter marks the refinement of our initial, albeit rudimentary, effort to transfer mesoscopic travel demand data into a microscopic traffic simulation, as discussed in Section 3.3.2. In contrast to this preliminary approach, we now place a strong emphasis on the development of effective heuristics and data processing techniques to advance our travel demand transfer procedure from a previously manual, labor-intensive approach to a highly automated and reliable toolchain module with manageable computational demands.

To achieve this, we identify suitable fleet representatives from the pool of MATSim DRT vehicles (Sec. 5.1) to reduce the computational costs associated with microscopic ego-vehicle tracking. Furthermore, we introduce a method that efficiently confines SUMO's network boundaries to the selected trajectories of interest, thereby minimizing the scope of microscopic modeling (Sec. 5.2). Lastly, we present an automated approach for synthesizing microscopic travel demand from MATSim sample runs (Sec. 5.3) and populate relevant network areas. By the end of this chapter, we will have the capability to automatically generate customized SUMO networks and their associated vehicle populations from existing MATSim simulations.

5.1 Identifying suitable fleet representatives

Starting from an existing MATSim DRT simulation, the principal aim of this section is to identify vehicles from the entire pool of DRT vehicles that are representative of fleet usage, hereafter referred to as fleet representatives. Firstly, these selected representatives help defining microscopic network boundaries of interest, as detailed in Sec. 5.2. Secondly, these chosen representatives

exclusively undergo the computationally resource-intensive process of microscopic tracking[1] in subsequent SUMO simulation. Depending on the specific use case requirements, we present methodologies for extracting either a single DRT vehicle or a sub-fleet of DRT vehicles. Importantly, our selection process ensures that the chosen vehicles are representative for the original vehicle fleet as a whole.

Our emphasis lies on the examination of three pivotal vehicle attributes: daily mileage, operation time, and the number of served trips. These attributes are thoroughly evaluated to determine a vehicle's suitability to serve as a representative within the fleet context. While MATSim provides access to numerous additional attributes such as average or maximum velocity, occupant counts, or waiting times between clients, these attributes are deliberately excluded from our analysis for the sake of simplicity.

5.1.1 Selecting representative ego-vehicles

In the context of autonomous electric vehicle (AEV) development, identifying a representative driving cycle is a crucial step in the process of technical requirement engineering. A representative driving cycle serves as a benchmark that summarizes the typical behavior of the entire AEV fleet, allowing OEMs and automotive suppliers to design, validate and optimize their vehicles and systems effectively. Consequently, this driving cycle becomes a foundational tool for AEV engineers, as it ensures that the AEVs are designed to meet the real-world demands of their intended use cases.

When identifying the representative among thousands of DRT vehicles, various methods can be considered, each with its own implications. The choice between these methods depends on the specific objectives and context for which the driving cycle will be used. In total, we implement four different methods, providing flexibility in choosing the most suitable approach.

[1] Denotes the recording of spatio-temporal whereabouts of individual agents throughout a SUMO simulation.

(i) Mean vehicle. This approach involves the selection of a vehicle that best represents the average characteristics of the fleet. While this method provides a well-balanced representation, it may not adequately account for extreme or outlier scenarios. Hence, we advise applying this method only if the presence of extreme cases is of lesser concern. Technically, the "mean vehicle" serves as an abstract representation, which summarizes the aggregated fleet characteristics without necessarily corresponding to an actual vehicle within the dataset. However, to approximate a mean vehicle, we rely on the method introduced in Section 2.4.2 with the arithmetic mean serving as suitable ensemble estimate $\bar{\xi}_j$ (with j of type mileage, operation time, number of rides). According to Eq. 2.5, the mean vehicle is then defined as the entity i whose attribute values $\xi_{i,j}$ exhibit the smallest cumulative Euclidean distance (L2-norm) to the corresponding attribute centroids (mean values) across all vehicles. To ensure that attributes with varying scales do not disproportionately influence the averaging results, we also perform feature normalization as demonstrated in Eq. 2.6.

At this point, we would also like to underscore our earlier findings discussed in Section 2.6.4, which highlight the need for caution when extrapolating results from sampled MATSim scenarios to full-scale 100% scenarios. Fleet-related KPIs cannot simply be scaled by multiplying with the inverse of the sample size $x \in (0, 1]$. As discussed by [109] and further elaborated in Section 2.6.5, the average vehicle mileage in 10% MATSim sample runs should be adjusted by a factor of 1.9, while average operation times remain largely unchanged. These distortions must be carefully considered when selecting representative "mean" fleet vehicles. However, the impact of such artifacts on the KPIs of specific percentile vehicles remains unexplored and warrants further investigation.

(ii) Median vehicle. The median, in contrast to the mean, offers a certain degree of resilience against the influence of outliers, making this metric a particularly apt choice for the derivation of a robust representative vehicle. The "median vehicle" corresponds to an actual vehicle within the dataset, capturing the prototypical behavior of all DRT vehicles in the fleet. Let V_i represent the i^{th} vehicle in the fleet with its belonging attributes $\xi_{i,j}$. Borrowed from Equation 2.6, the normalized Euclidean distance ($\tilde{d}_{i,n}$) from a specific vehicle V_i to another fleet vehicle V_n in the simulation can be calculated as

$$\tilde{d}_{i,n} = \sqrt{\sum_{j \in J} \left(\frac{\xi_{i,j} - \xi_{n,j}}{M_j} \right)^2}, \tag{5.1}$$

where $\xi_{n,j}$ corresponds to the attribute set of the n^{th} fleet vehicle, and

$$M_j = \max_{i,n \in N} |\xi_{i,j} - \xi_{n,j}| \tag{5.2}$$

ensures the normalization of the different distance contributions across the entire fleet. After calculating the normalized Euclidean distances for each vehicle V_i in the dataset N, the median vehicle (V_m) is the one for which the sum of distances to all other vehicles $n \in N$ is minimized:

$$V_m = \arg\min_i \sum_{n \in N} \tilde{d}_{i,n} \tag{5.3}$$

In this formula, $\arg\min_i$ denotes the index of the vehicle that minimizes the sum of distances ($\sum_n \tilde{d}_{i,n}$). In essence, V_m maintains the closest proximity to all other vehicles within the dataset, representing the "median" behavior of the fleet.

(iii) 97.5th-percentile vehicle for worst-case scenario. This approach entails identifying the vehicle whose attributes closely approach the upper limit of the (mean-centered) 95-percent confidence interval, as demonstrated in Figure 5.1, within a dataset characterized by normally distributed data with a single attribute. This method is recommended for requirement engineering scenarios where extreme cases carry substantial significance, ensuring that the selected vehicle can handle worst-case scenarios with a high degree of confidence. It is a conservative approach that focuses on risk mitigation.

To identify a vehicle that covers 97.5% of all driving situations within a multi-variant environment, it is necessary to conduct a statistical analysis that accounts for attribute variability across the dataset. However, instead of calculating the mean and standard deviation for each attribute across all vehicles in the dataset, we propose an approach similar to that used for determining the median vehicle.

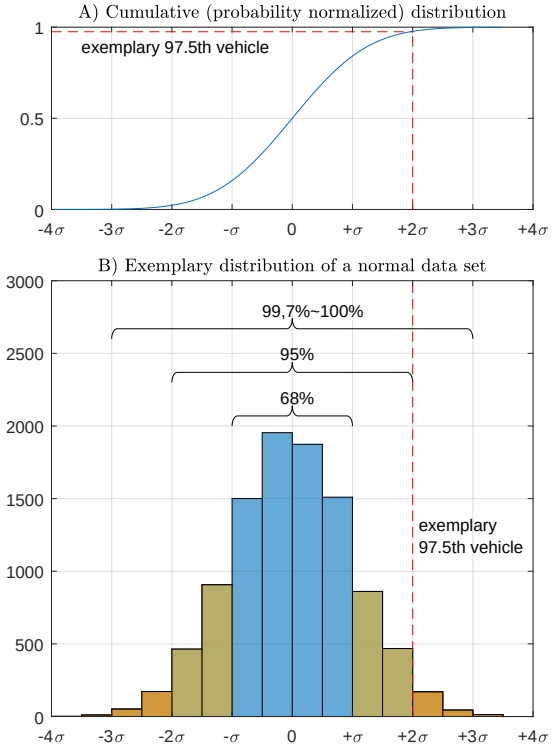

Figure 5.1: Exemplary visualization of a normally distributed dataset (bottom) with its cumulative distribution (top). The values within one standard deviation σ of the mean account for about 68% of the set; while within two and three standard deviations the values account for about 95% and 99.7% respectively. For our requirement engineering purpose, where extreme cases carry significant weight, we choose to identify the vehicle whose attributes closely approach the upper limit of the (mean-centered) 95-percent confidence interval.

For illustrative purposes, consider a DRT fleet comprising $N = 1000$ cars. If we designate the median vehicle (V_m) as the one for which the sum of distances $\sum_{n \in N} \tilde{d}_{i,n}$ to all other vehicles n within the dataset N is minimized (see Eq. 5.3), we define the 97.5th-percentile vehicle ($V_{97.5}$) to be located at the 975$^{\text{th}}$ position when sorted by $\tilde{d}_i$ in ascending order.

To focus on the right-tail of the distribution, which represents worst-case scenarios of high mileage, long operation time, and a large number of served trips, we need to introduce negative contributions for vehicles in the left tail. Therefore, we incorporate the signum function $\text{sgn}(\xi_{i,j} - \xi_{n,j})$ to distinguish whether a given attribute contributes positively (right-tail) or negatively (left-tail) to the distance. The normalized Euclidean distance is thus calculated as:

$$\tilde{d}_i = \sqrt{\sum_{j \in J} \text{sgn}(\xi_{i,j} - \xi_{n,j}) \left(\frac{\xi_{i,j} - \xi_{n,j}}{M_j} \right)^2} \tag{5.4}$$

where M_j is defined in Equation 5.2, and $\text{sgn}(\xi_{i,j} - \xi_{n,j})$ is the signum function of the difference between the attributes of vehicles i and n. For fleet sizes $N \neq 1000$, we identify the vehicle of interest based on its relative position within the sample size N.

(iv) Data clustering. We opt for a data clustering technique as the fourth method to identify representative vehicles. This approach is particularly well-suited for managing large and heterogeneous datasets by forming cohesive clusters, each characterized by distinct usage patterns. These clusters help summarize the diverse operational profiles within our dataset. By selecting the vehicle closest to the cluster centroid, we ensure that the representative vehicle accurately reflects the typical behavior of that group.

As an example, we employ the *k-means clustering* method to categorize DRT vehicles into three distinct clusters based on their usage: lightly used vehicles, moderately used vehicles, and heavily used vehicles. These categories are determined by attributes such as the number of served trips, daily mileage, and operational time, as illustrated in Figure 5.2.

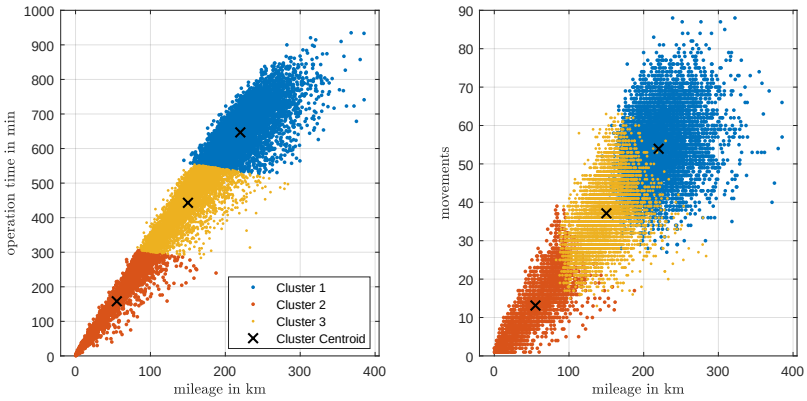

Figure 5.2: K-means clustering was applied to a dataset consisting of 14,127 DRT fleet vehicles, resulting in the formation of three distinct clusters based on various attribute combinations. The bluish cluster represents heavily used vehicles, comprising approximately 45% of the entire fleet. The yellow cluster signifies moderately-used duty vehicles (approximately 38%) characterized by moderate mileage, operational time, and the number of served trips. Meanwhile, the reddish cluster corresponds to lightly-used DRT vehicles, accounting for approximately 17% of the total fleet.

Next, we compute the centroid for each cluster, representing the average values of these attributes for the vehicles within that group. For each cluster, we identify the vehicle whose attributes are closest to the cluster's centroid in terms of normalized Euclidean distance (see Equation 2.6). This vehicle is then considered the representative for that specific usage group. Generally, the methodologies described in paragraphs (i) through (iii) could also be applied to the identified clusters.

5.1.2 Selection of a representative sub-fleet from a DRT fleet

Large-scale traffic simulations, which aim to replicate the daily mobility patterns of millions of agents, pose substantial computational challenges. Certain simulation output configurations, such as spatio-temporal vehicle tracking or detailed network state logging, significantly increase computation time and

memory requirements. To enhance the efficiency of our MATSim-SUMO toolchain, we are exploring solutions to only track a selected subset of fleet vehicles while preserving information about the entire fleet.

The most resource-efficient approach involves tracking a single vehicle, one that best represents the desired driving pattern. However, vehicle component design depends on various factors, including worst-case load profiles, cost-effectiveness, and safety, comfort, or environmental considerations. Consequently, tracking a single vehicle is not ideal, as it necessitates repetitive microscopic traffic simulations when sizing multiple vehicle components or loads of multiple cars are of concern.

Instead, we propose a versatile approach where we perform the computationally intensive traffic simulation only once but with the possibility to switch between several tracked vehicles during post-processing. Notably, we omit tracking fleet vehicles that either exhibit redundant mobility patterns or do not contribute new information to the overall fleet behavior.

The central question guiding our research is determining the optimal number and selection of MATSim fleet vehicles to be tracked in SUMO, ensuring that they remain statistically significant in terms of average fleet KPIs. This approach allows us to strike a balance between computing demand and the accuracy of our simulation results, ultimately improving the utility and practicality of large-scale traffic simulations.

To address this question, we conducted an analysis using a 10% sample run of MATSim, involving a total of 14,127 simulated fleet vehicles[2]. Table 5.1 provides a summary of all critical model inputs for this DRT scenario. Based upon this simulation, we explore the statistical significance of tracking a specific subset of vehicles in relation to average fleet KPIs.

Reducing a large dataset while preserving the distribution of its attributes is a well-known challenge in statistics and data analysis. Our approach employs a multi-variant form of *stratified sampling* [56], which involves segmenting the dataset into strata based on key attributes (such as daily mileage, operation

[2] It is crucial to emphasize that the fleet vehicle count in this analysis has not been adjusted to represent a 100% sample size. The question whether the derived subfleet accurately represents the entirety of a 100% sample remains subject for future investigation, but is beyond the scope of our current research.

Table 5.1: Model inputs of the 10% DRT scenario.

base scenario	MATSim Open Berlin Scenario [234]
service area	full city
pooling capacity	1
price (EUR/km)	0.18
base price (EUR)	0
min price (EUR)	1
constant daily car costs (EUR)	-10.6
car costs (EUR/km)	-0.2

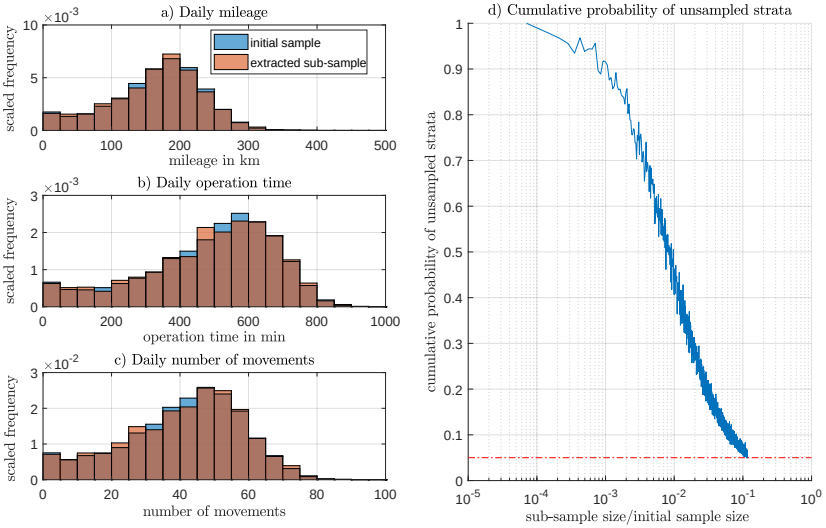

Figure 5.3: The left-side panels provide a visual representation of the scaled attribute distributions, where the bluish area represents the initial DRT fleet sample, and the reddish area represents the corresponding subset generated through multivariate stratified sampling. On the right, the logarithmically-scaled cumulative probability of the unselected strata is depicted throughout the sampling process. Sampling concluded when the cumulative probability of unsampled strata fell below 5% (red dashed line), ensuring that the selected sub-sample retains at least 95% of the original dataset's variability. Through this process, we effectively reduced the initial sample size from 14,127 DRT vehicles to 1,618 cars, while preserving a high degree of conformity between the two datasets.

time, and number of served trips) and then drawing samples from each stratum in proportion to its representation in the original dataset. Here, each stratum carries a specific probability of being selected, reflecting its contribution to the total dataset. The cumulative probability across all strata is one, ensuring that the entire population is fully represented in the sampling process. By nesting these attribute strata within each other, along with their associated probabilities, we account for interactions between attributes, thereby achieving a more accurate representation of the population in our sample. The sub-sampling process – marked by the iterative selection of samples from the strata – is considered complete once the cumulative probabilities of the unselected strata[3] in the original dataset fall below 5%. This ensures that the selected sub-sample retains at least 95% of the total variability present in the original dataset.

The high level of confidence achieved by this systematic stratified sampling approach is illustrated in Figure 5.3. In our case study, a final sub-sample size of 1,618 vehicles is considered sufficient to represent the entire DRT fleet, which initially comprised 17,127 vehicles. This sub-sample represents only 11% of the total fleet size. Furthermore, we have qualitatively observed that as the distribution of attributes approaches normality, the relative size of the representative subset decreases.

5.2 Implementation of dynamic meso-micro network boundaries

Given the resource-intensive nature of large-scale microscopic traffic simulations, our primary objective is to focus on detailed simulations of specific areas or selected trajectories, rather than re-simulating the entire mesoscopic network area. To achieve this, we developed three different approaches to tailor the boundaries of the SUMO network, as outlined in Figure 5.4. In the ensuing subsections, we will provide a concise overview of all three network extraction techniques, followed by a discussion of their respective advantages and areas of application.

[3] Strata that were not sampled.

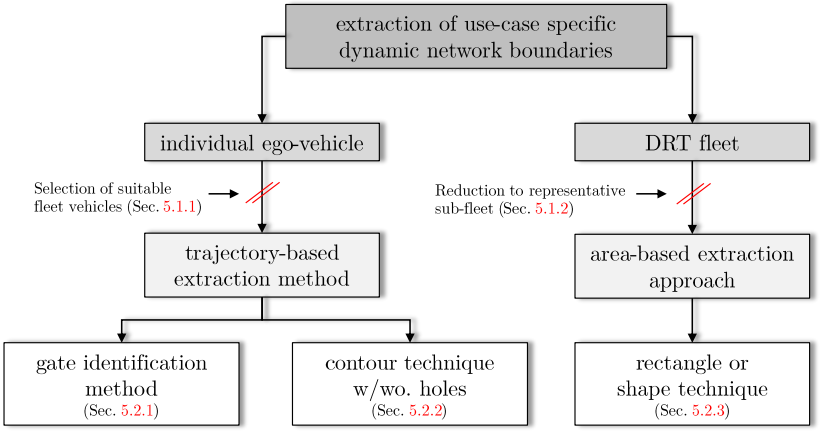

Figure 5.4: Overview of use-case specific network extraction techniques

5.2.1 Gate identification method

In this trajectory-based approach, we denote a *gate* as a conceptual node positioned at the outer boundary of the microscopic test case area. Gates basically delineate the nodes at which the junction between mesoscopic and microscopic network representation occurs.

The principle behind gate identification is as follows: Initially, we assign an *order* to each node and link within the MATSim digraph. Nodes and links of order 1 represent the trajectory of the selected DRT vehicle in MATSim. Nodes and links of order 2 are those directly adjacent to the order 1 elements. Similarly, nodes and links of order 3 are connected to elements of order 2, and this process continues for higher orders.

Through a breadth-first search traversal of the MATSim digraph, we identify suitable gates based on four key criteria: (i) A gate must be a node with an order of at least 3. (ii) A gate should be situated at a minimum distance of 100 meters from nodes of order 1. (iii) A gate node must have a corresponding SUMO counterpart with high fidelity ($Z > 0.8$; see Sec. 4.3.3). During this process, we systematically evaluate various gate hierarchies, as illustrated in Figure 5.5.

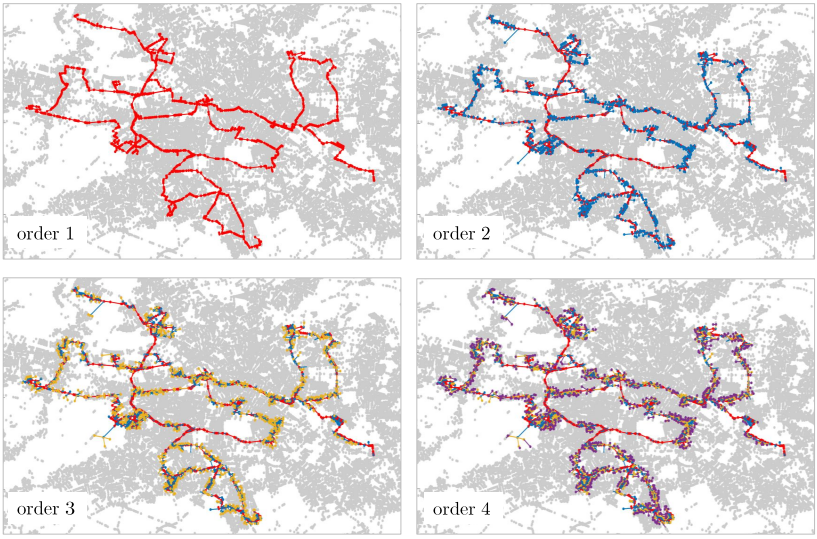

Figure 5.5: An illustrative representation of the gate identification methodology applied to the trajectory of a selected DRT fleet vehicle within the Berlin metropolitan area. Commencing with the primary trajectory of our ego-vehicle (depicted by the red path in the upper left panel), successive exploration of neighboring nodes is undertaken, with increasing levels of depth (order), until all four specified criteria are satisfied.

The fourth gate criterion (iv) is more intricate. In certain cases, a gate may become "ingrown," meaning it is surrounded by nodes and links of higher orders. For instance, consider a node G that initially satisfies the criteria (i) to (iii) for being a gate. However, in the immediate vicinity of G, adjacent nodes do not yet fulfill the criteria for gate status, necessitating further exploration of higher-order nodes. As a result, the initially designated gate G becomes ingrown and loses its eligibility for gate status, as it no longer serves as a boundary to the current test case area but instead forms an isolated feature within the microscopic network.

Based on the MATSim gate nodes, we define a geographic boundary using MATLAB's `boundary` function[4], where the *shrink factor* controls the degree of tightness in the gate enclosure, as illustrated in panels G2 and G3 of Figure 5.6. Conceptually, the gates can be visualized as nails hammered into a wooden surface. A shrink factor of zero represents a boundary analogous to an elastic band stretched around the outermost nails, enclosing only the peripheral points. In contrast, a shrink factor of one pulls the boundary inward, similar to the contraction observed when vacuum-sealing an object by extracting air, where the elastic band conforms tightly around inner nodes. For our purposes, we choose a shrink factor of one to enclose all gates while minimizing the inconvenience of importing excessive, microscopic street segments.

[4] "Boundary Documentation," MATLAB, https://de.mathworks.com/help/matlab/ref/boundary.html. Accessed: Sep. 15, 2022.

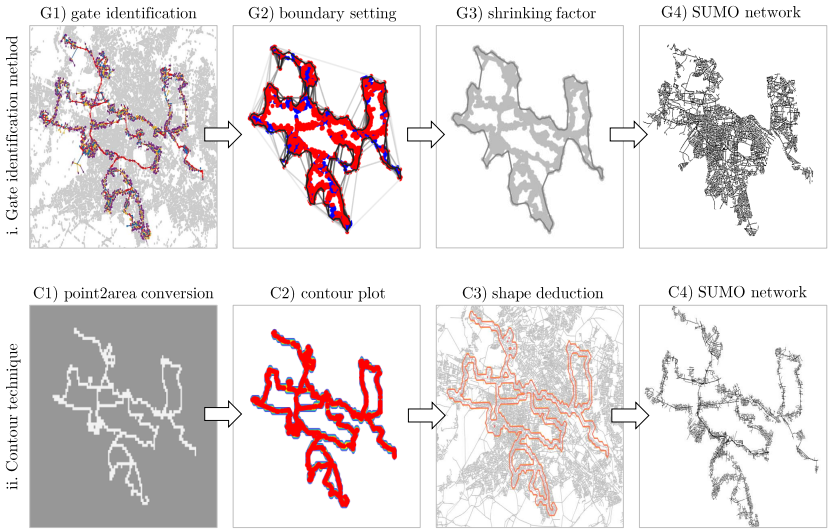

Figure 5.6: Comparative illustration of trajectory-based network generation approaches: gate identification method (top panels) vs. contour technique (bottom panels). Both methodologies start with the trajectory of the selected DRT vehicle (depicted in the left-side panels) and, through a series of distinct procedural stages, result in tailored SUMO networks (as exhibited in the right-side panels).

Once the boundary is defined, we convert it into a shapefile, which is then used to geographically delimit the SUMO road network during the OSM network import, as demonstrated in panel G4 of Figure 5.5. A key limitation of this approach, however, is its inability to create 'holes' within the network. For example, in Figure 5.5, our vehicle of interest repeatedly crosses its own path throughout the day, enclosing extensive urban areas. As a result, we are required to model these areas in their entirety within SUMO, significantly increasing the complexity of network cleaning, population, and simulation tasks.

5.2.2 Contour technique

The contour technique offers a more practical approach for defining SUMO network boundaries based on the trajectory of an ego-vehicle, as illustrated in the lower panels of Figure 5.6. Initially, we enclose all MATSim nodes and links associated with the vehicle's trajectory within a rectangular area (panel C1), which is then rasterized into pixels. For optimal resolution, we recommend each pixel to cover approximately 200-300 meters in real-world dimensions. If a pixel contains a node or is intersected by a link[5] that belongs to the ego-vehicle's trajectory, it is flagged, as indicated by the lighter color in panel C1 of Figure 5.6.

Next, we generate a contour plot (panel C2), which leads to the creation of a multilayered shapefile containing the polygonal courses (panel C3) of both the outer boundary and any inner boundaries ('holes'), arranged in a sorted order. The shapefile is then used once more during the SUMO network import process to define the microscopic network, including any internal 'holes' (panel C4).

One of the main advantages of this method is its flexibility, which allows for fine-tuning of the contour width and adjustment of the maximum distance from the trajectory. However, a significant limitation is that SUMO's boundary nodes may not always have a corresponding MATSim counterpart (see Sec. 4.3.3). Consequently, as detailed in Section 5.3, ambient traffic vehicles may be introduced into the SUMO network later in their routes, specifically at the first node with a viable counterpart. To mitigate potential impacts on our

[5] achieved by representing each link with a dense set of equidistant auxiliary points

ego-vehicle, we recommend setting the contour width (or, equivalently, the pixel resolution) to a sufficiently large value.

5.2.3 Area-based network generation

This approach involves the generation of SUMO networks either based on rectangular areas delineated by longitude and latitude coordinates or, alternatively, by employing pre-existing shape files, e.g. the shape file defining the DRT service area in MATSim. Other user-specified shape files are also qualified to serve as input.

5.2.4 Discussion

Initially, our primary objective was to keep the microscopic network as compact as possible in order to minimize network cleaning efforts. Accordingly, we focused on techniques designed to transfer a single DRT vehicle of particular interest – typically the most representative vehicle for a given use case – along with all the ambient traffic it encounters, into the microscopic network.

Using the ego-vehicle's trajectory in MATSim, we developed two trajectory-based methods to generate the corresponding road network in SUMO. In both approaches, the SUMO network was constructed to encapsulate only the specific vehicle trajectories and the road segments closely aligned with these trajectories. This strategy aimed to reduce the risk of erroneous junctions and traffic light configurations. However, the gate identification method, while comprehensive and effective for yielding appropriate MATSim-SUMO gate nodes, proved to be highly resource-intensive and impractical for routine use. As a result, we replaced this technique with the contour-based approach, which became our default method for trajectory-based network generation.

As our toolchain evolved, our objectives expanded beyond modeling a single ego-vehicle in SUMO. Consequently, we transitioned to area-based network modeling techniques, focused on transferring entire DRT fleets – or representative sub-fleets – from MATSim to SUMO. From an engineering perspective, modeling a set of DRT fleet vehicles along with their collective driving patterns yields a more comprehensive and versatile dataset, providing extensive insights

through a single microscopic simulation. Accordingly, we adopted the approach illustrated in the right branch of Figure 5.4. As discussed in Chapters 6 and 7, simulating traffic in larger (non-trajectory-based) areas generally enhances simulation fluidity. To address the challenges associated with extensive network cleaning, we are designing the upcoming MATSim-SUMO toolchain and, more critically, the microscopic traffic simulation in SUMO to be resilient to imperfections in the network.

5.3 Synthesis of microscopic travel demand from MATSim sample runs

This section focuses on synthesizing complete microscopic travel demand from mesoscopic transport simulations, where only a subset of agents is represented. The method presented here refines the approach outlined in Section 3.3.2, which lacked the necessary sophistication to automatically generate road-bound microscopic travel demand from large-scale MATSim simulations. A comparison of the two travel demand transfer methods is provided in Table 5.2.

Table 5.2: Key differences between our initial travel demand transfer procedure (middle column) and the revised approach (right column).

	initial procedure (Sec. 3.3.2)	new procedure (Sec. 5.3)
automation capability	manual	fully automated
microscopic network scope	solitary complexity-stripped street segments	complex urban areas
meso-micro boundaries	manually selected through visual inspection	dynamically selected based on MATSim output
translation of network elements	manual	automated matching routine
spatial injection of agents	entrance and exit on boundary edges only	entrance and exit on all network edges possible
travel demand disaggregation	cloning of agents with departure times adjusted by a temporal offset sampled from a Gaussian distribution	
microscopic traffic assignment	via trips; routes are determined by `DUAROUTER`	pre-defined routes or adhoc rerouting
network cleaning	manually	ignored

During the initial phase of our investigation (see Sec. 3.3), we focused on simplified test cases to assess the feasibility of transferring travel demand between mesoscopic and microscopic network models. This phase involved several key simplifications in our methodology. First, by concentrating on isolated street segments, we overlooked the complexities inherent in simulating larger urban network areas, including traffic flow bottlenecks stemming from complex junctions or network imperfections. Second, we restricted the entrances and exits of agents in the test case to designated boundary links[6], deliberately excluding such events on other street segments within the test case. We considered this a valid simplification, given that all three test cases outlined in Section 3.3.1 contained relatively few activity zones (such as residential, work, or leisure locations) that could have attracted agents in the MATSim simulation. Third, we initiated SUMO simulation based on vehicle trips only, without specifying complete vehicle routes. Instead, we utilized the `DUAROUTER`[7] tool to determine suitable routes prior to the actual traffic simulation. In general, these routes do not necessarily align with the corresponding MATSim routes; however, due to the limited scope of the test cases, the potential for significant route divergence was minimal.

In addition to these major simplifications, our previous travel demand transfer procedure involved substantial manual intervention, particularly in identifying suitable boundary links in the MATSim environment and translating them into corresponding SUMO counterparts. Additionally, significant manual effort was required to adapt the SUMO network to the desired test case shape. We used the SUMO network editor, `Netedit`[8] , to correct network anomalies caused by import heuristics and to refine lane connections within junctions and traffic signal control strategies. In hindsight, this network optimization process proved to be highly resource-intensive and time-consuming.

While this simplified approach to synthesizing microscopic travel demand yielded satisfactory results during our preliminary study in Section 3.3, its feasibility for larger test cases is limited. Its major drawback is the inability

[6] Boundary links delineate the microscopic test case area, representing the final line of street segments before adjacent roads were trimmed or clipped.

[7] "Duarouter Documentation," SUMO, https://sumo.dlr.de/docs/duarouter.html. Accessed: Mar. 13, 2019.

[8] "Netedit Documentation," SUMO, https://sumo.dlr.de/docs/Netedit/index.html. Accessed: Mar. 13, 2019.

to automatically adapt to new and more complex test cases. Additionally, the resource-intensive nature of network cleansing becomes unmanageable in the context of extensive, area-covering DRT fleet simulations. Although our network matching routine (see Section 4.3.3) and the implementation of dynamic meso-micro network boundaries (see Section 5.2) streamline certain processes, additional improvements to the travel demand transfer procedure are necessary and will be detailed in the following paragraphs. A solution to the network cleansing challenge will be addressed in Chapter 6.

Reconstruction of spatio-temporal vehicle routes Instead of tracking vehicle movements solely on boundary links, we analyze the MATSim event file (`events.xml`) for all "vehicle-enters/leaves-traffic" and "link-enter/leave" events to reconstruct the spatio-temporal trajectories of all vehicles. The event file logs all significant actions undertaken by agents during *mobsim* (see Section 2.1.1), as illustrated in Figure 5.7.

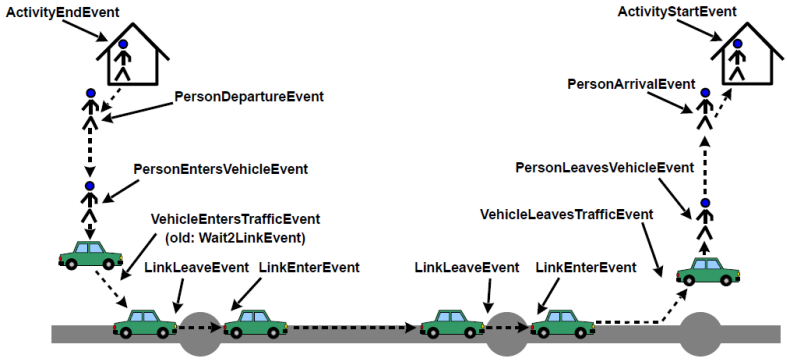

Figure 5.7: Schematic representation of the sequential actions an agent undertakes in MATSim when selecting the car travel mode. All simulation actions, including those where agents initiate activities or transition between links, are recorded as MATSim events. Each event is characterized by multiple attributes, typically including the event timestamp, the ID of the agent or vehicle involved, and the link ID where the event occurred. For DRT fleets, additional dispatch-related events, such as the start and end of a `dvrp` task, are also captured. This illustration is extracted from the MATSim User Guide [98].

Truncation of vehicle trajectories to microscopic network boundaries We then eliminate all vehicle events that occur outside the defined boundaries of our SUMO network. This filtering step is essential for addressing scenarios where MATSim vehicles merely cross the smaller SUMO network. To facilitate the integration of these vehicle trajectories into the SUMO simulation, further conditioning is necessary. Vehicles that pass through the defined microscopic area multiple times are treated as separate entities within SUMO, even if these movements originate from a single vehicle trip in MATSim. This truncation procedure is schematically illustrated in Figure 5.8. Moreover, it is necessary to adjust the trip departure times in SUMO to align with the timestamps at which vehicles crossed the meso-micro boundary.

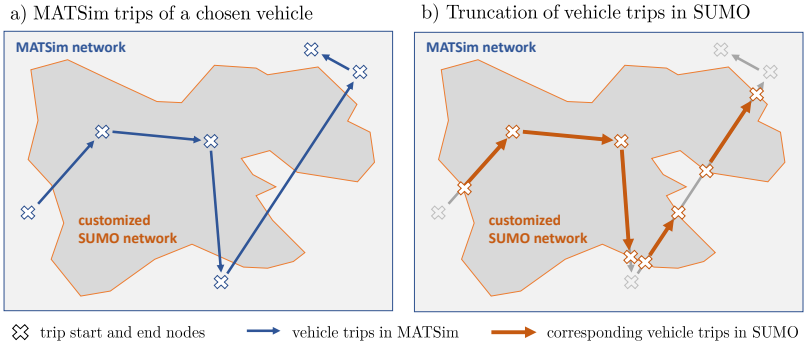

Figure 5.8: Visualization of the trip truncation procedure: MATSim simulations, represented by the bluish rectangles in both panels, often cover extensive areas, whereas SUMO simulations are focused on smaller network regions (indicated by the orange contoured area). When MATSim vehicle trajectories intersect but do not remain within the SUMO network boundaries (as shown by the bluish arrows in the left panel), these trajectories undergo alteration, resulting in the truncation of certain route segments or entire trips (depicted by the orange arrows in the right panel).

Traffic assignment based on vehicle routes Vehicle routes are incorporated into the SUMO simulation using the *via*[9] attribute. This approach

[9] Trip definition with incomplete routes (SUMO Documentation): https://sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html. Accessed: Apr. 25, 2023.

facilitates the management of incomplete SUMO edge paths, which can arise from the intentional exclusion of street segments with a matching score below our threshold ($Z > 0.8$; see Chapter 4.3.3). Initiating SUMO simulations with complete but potentially mismatched edge paths could lead to simulation failures. Currently, we do not implement any rerouting mechanisms;[10] both ego-vehicles and ambient traffic adhere strictly to their initially assigned routes.

Final modules for initiating SUMO simulation Figure 5.9 summarizes the key steps necessary to built a customized SUMO network (represented by the bluish boxes) and to derive the corresponding microscopic vehicle population (depicted by the reddish boxes) from an existing MATSim simulation. So far, we have successfully automated the generation of two of the three critical SUMO input files (highlighted in yellow boxes), namely the network and routes file. An essential – yet missing – component for initiating SUMO simulations is the configuration file. The parameters within this file play a pivotal role in the meso-micro calibration process, which is discussed in detail in Chapter 7. Therefore, the final parameter settings will be determined in that chapter.

Prior to this, however, we must address the compatibility between MATSim and SUMO regarding traffic dynamics and network capacity, and develop strategies to align both frameworks effectively within the context of our vehicle-centered use case. This issue constitutes a significant portion of this thesis and will be explored in depth in Chapter 6.

[10] This constraint will be relaxed as a result of the developments presented in Chapter 6.

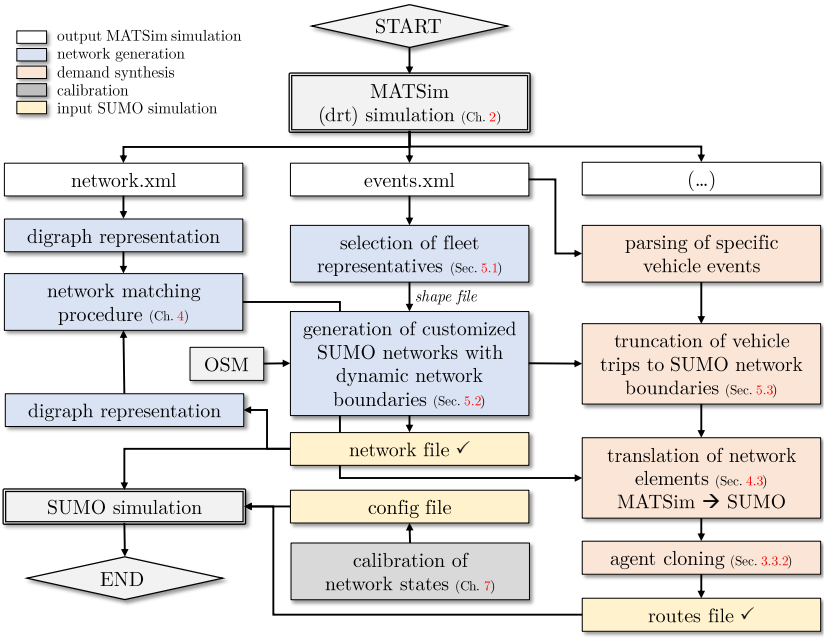

Figure 5.9: Foundational processes for creating a tailored SUMO network (depicted in bluish boxes) and its corresponding microscopic vehicle population (illustrated in reddish boxes) by leveraging existing MATSim simulations. At the end of this process, two of the three key SUMO input files (highlighted in yellow boxes) are automatically generated, paving the way for the initiation of the microscopic SUMO simulation. Of paramount importance is the configuration file, the parameters of which hold a central role in our forthcoming calibration procedure, as elaborated in Chapter 7.

6 Striving for synchronization – towards consistent MATSim and SUMO simulations

Shifting our attention from the foundational chapters, our focus now turns to the practical integration of mesoscopic transport planning and microscopic traffic simulation. In this context, MATSim and SUMO serve as the primary tools for this integration.

This chapter commences with a comprehensive literature review on the subject of hybrid and multi-level traffic simulation in Section 6.1, aiming to gain valuable insights into how to effectively harmonize simulation frameworks of varying aggregation levels and modeling regimes. Subsequently, Section 6.2 delves into a thorough analytical and numerical examination of traffic dynamics in both MATSim and SUMO, with the intent of establishing consistency in this crucial aspect.

As network capacity depends on both traffic dynamics and infrastructural elements, we conduct a systematic numerical analysis in Section 6.3 to assess the maximum achievable flow rates for various junction types for both MATSim and SUMO simulations. We discuss the implications of out findings for our tool-coupling efforts in Section 6.4.

The findings presented in this Chapter have already been documented in our prior publication [207]. This paper was developed in parallel with our calibration procedure outlined in Chapter 7. As a result of this concurrent development, certain components of the calibration methodology (pre-published in [208]) were already applied to the content of this chapter. These concurrent advancements are marked and discussed as we proceed.

6.1 Advancements in hybrid and multi-level simulation techniques

The deduction of SAEV drive cycles requires the fusion of two opposing objectives, namely large-scale DRT fleet simulation and accurate modeling of driving dynamics. While the former aspect is often addressed through mesoscopic agent-based transport simulation, the latter belongs to the domain of microscopic traffic modeling [207]. For SAEV drive cycle deduction we therefore need to combine both domains.

Multi-level traffic simulation has been object of research for many years [88]. The evident constraints of exclusive macro-, meso-, and micro-simulation approaches have prompted the emergence of a wide variety of hybrid methods that differ in terms of aggregation level, methodology, and scope of application. By modeling areas of interest in high detail while adopting coarser representations for surrounding regions, multi-level approaches are well-suited to capture network-wide effects of local phenomena [42, 94], optimize city-wide traffic management systems [44], and enhance simulation performance [228]. Typically, research in this domain either encompasses the evaluation of novel multi-level simulation techniques [228, 37] or the description of protocols that enable the coexistence of models [37, 108].

Distinctions are made between *macro-micro* and *meso-micro* approaches. *Macro-micro* models combine flow- and vehicle-based traffic models [94, 228, 37, 108]. They often incorporate *transition zones* [94, 37] where both regimes co-exist, facilitating the aggregation and disaggregation processes at network borders. Disaggregation is particularly challenging because it involves reconstructing information from lower-resolution models [228]. Furthermore, the process of (dis-)aggregation comes at the price of computational overhead. To address this issue, the usage of *multi-resolution entities* has been suggested to consolidate properties from various resolutions into a single object [228]. Further studies have explored hybrid models combining micro- and macroscopic representations of the *Lighthill-Witham-Richards (LWR) model* [37, 129, 182].

Meso-micro approaches, on the other hand, have also received attention [42, 203, 44, 54, 88], emphasizing the need for consistency across various facets, such as traffic dynamics, network representation, vehicle routing, traffic performance, and information exchange to enable co-simulation [44]. Dynamic modeling

of micro-meso boundaries has been explored through a *holonic approach*, where individual vehicles can operate autonomously or in grouped formations to optimize computational efficiency [203]. In the realm of microscopic simulation, concepts like *ghost vehicles* (used to prevent unwanted accelerations in border regions) and *virtual links* (facilitating route consistency) are commonly employed [37, 88, 42].

Despite the advantages of co-simulation approaches, they remain challenging to implement and are often abandoned due to their complexity [203]. In comparison, the potential of sequential tool-coupling approaches for specific use-cases in large-scale applications is frequently overlooked.

In contrast to existing literature, our tool-coupling endeavor is constrained by an established corporate tool landscape, which imposes specific limitations and requirements on the integration process. For instance, this landscape includes MATSim for evaluating new mobility concepts and services, and SUMO for vehicle-centric analyses. Consequently, we cannot simply choose compatible micro- and mesoscopic traffic flow models as the foundation for our work, as done in [37]. Instead, we must integrate and harmonize these two tools to enable the derivation of SAEV drive cycles. To the best of our knowledge, a robust MATSim-SUMO coupling for large-scale 24-hour simulations has yet to be demonstrated. Although the *High Level Architecture* approach outlined in [88] evaluates the runtime performance of a MATSim-SUMO coupling, it does not comprehensively address consistency issues.

In the ensuing analysis, we undertake an examination of the congruence between MATSim and SUMO, focusing primarily on traffic dynamics, and to a lesser extent, network capacity. The overarching objective is to ascertain a viable meso-micro coupling approach specifically tailored to our specific use-case. This also includes the identification of a feasible solution path in the event of non-compatible frameworks.

To accomplish this objective, we systematically investigate the traffic dynamics of both MATSim and SUMO across a series of three experiments of increasing complexity: (i) analytically on a homogeneous road segment in steady-state , (ii) numerically on a homogeneous road segment under non-stationary conditions for a synthetic test case, and (iii) numerically for a highly non-linear real-world test case in Berlin of medium size.

The simulation results from both frameworks are analyzed with respect to macroscopic flow-density-speed relations. It is important to note that the outcomes of the idealized experiments (i) and (ii) are largely determined by the methods employed to model traffic dynamics. In contrast, the findings from the realistic test case (iii) are additionally influenced by differences in network representation between MATSim and SUMO. In pursuit of a comprehensive understanding, we also delve into network *resistances*[1] within the Berlin test case. This involves determining the maximum flow rates at corresponding junctions in both MATSim and SUMO, and evaluating how these rates are affected by the junctions' right-of-way rules.

6.2 Analytical and numerical analysis of traffic dynamics in MATSim and SUMO

6.2.1 Analytical analysis of the steady-state equilibrium on homogeneous road segments

In *steady-state* conditions, also known as the *stationary case*, all macroscopic traffic variables – such as flow q, density ρ, and average speed v – remain constant over time. On a homogeneous road segment without additional on- or off-ramps or geometric irregularities, both vehicle flow and inter-vehicle spacing are stable. In this scenario, SUMO operates without stochastic variations, assuming that all vehicle-driver units have identical properties. Moreover, no interactions between vehicles occur, as acceleration is set to zero, capturing pure free-flow conditions. It is crucial that both MATSim and SUMO adhere to these boundary conditions. Without these prerequisites, neither framework can be expected to produce consistent results in terms of traffic dynamics.

However, neither MATSim's spatial queue model nor SUMO's car-following model directly incorporate macroscopic traffic quantities due to their micro-

[1] We deliberately use the term "resistance" here by analogy to electrical circuits, where resistance quantifies how much a component opposes the flow of electric current. Similarly, in traffic modeling, a narrow road segment can restrict the continuous flow of vehicles. A higher network resistance reflects greater difficulty for vehicles to navigate through the network, much like increased electrical resistance impedes current flow in a circuit.

scopic nature. Microscopic models focus on the interactions between individual vehicles and drivers, capturing how they respond to nearby vehicles and road infrastructure. The cumulative effect of these interactions shapes the overall macroscopic traffic flow. Macroscopic values can be viewed as a limiting case (*Grenzfall*) of ensemble averages that only emerge in microscopic simulations as the size of the system tends toward infinity. Therefore, deriving macroscopic quantities like average flow or density from these microscopic models on a single-link level is predominantly a theoretical exercise aimed at ensuring consistency between different modeling approaches under idealized conditions. Such macroscopic measures are only meaningful under specific boundary conditions, especially in steady-state scenarios where microscopic quantities like individual vehicle speed and inter-vehicle spacing align directly with macroscopic variables. In the following, we establish additional boundary conditions to clarify the relationship between microscopic dynamics and macroscopic quantities on a homogeneous road segment in steady-state.

Incorporation of road length in stationary conditions In stationary conditions, the flow entering a road segment (q^{in}) must be equal to the flow exiting the road segment (q^{out}):

$$q^{\text{in}} = q^{\text{out}} = q. \tag{6.1}$$

In MATSim, the number of vehicles N on a link is defined as [78]:

$$N(t) = UQ(t) = UQ(t-\delta) + \underbrace{\delta\left[q^{\text{in}}(t-\delta) - q^{\text{out}}(t-\delta)\right]}_{\delta q = 0 \Leftrightarrow stationary}, \tag{6.2}$$

which, under the condition given by Eq. 6.1, results in a constant number of vehicles $N = UQ$. Incorporating this into the fundamental definition of density, which represents the number of vehicles per unit length of the road (L), results in the following equation:

$$\rho = \frac{N}{L} = \frac{UQ}{L}. \tag{6.3}$$

Moreover, the fundamental relationship between macroscopic flow, density, and speed is given by:

$$q = \rho \cdot v. \tag{6.4}$$

This relationship typically holds for large or infinite road segments, where changes in vehicle positions do not significantly alter flow and density characteristics. However, for finite and especially short road segments, we need to incorporate the link length L into stationary conditions to establish an equilibrium between flow q, density ρ, and speed v that respects the spatial constraints of the link at each discrete simulation time step in both MATSim and SUMO.

To ensure that the inflow and outflow are spaced such that the link never becomes empty, we require that the free-speed link travel time, given by:

$$T_{\text{free}} = \frac{L}{v}, \tag{6.5}$$

corresponds to an integer multiple of the time headway[2] between consecutive vehicles, defined as:

$$\Delta t = \frac{1}{q}. \tag{6.6}$$

This leads to the boundary condition $T_{\text{free}} = k \cdot \Delta t$ (where $k \in \mathbb{N}$), which both MATSim and SUMO must satisfy under stationary conditions:

$$\frac{L}{v} = \frac{k}{q}. \tag{6.7}$$

Depending on the chosen average speed and flow rate, various steady-state configurations can be established using this boundary condition, each ensuring spatial consistency along the road segment.

Linking SUMO's car-following behavior to macroscopic traffic parameters Converting SUMO's microscopic car-following model into macroscopic quantities requires a slightly different but similar approach. Instead of a specific time headway (see Eq. 6.6) between consecutive vehicles, we require a well-defined gap function $g(t)$ to ensure that the Krauss model provided in Equation 3.11 satisfies stationary conditions. To achieve this, we employ once again the fundamental definition of density. Specifically, we express the density

[2] Time interval during which two consecutive vehicles cross a specific location with their respective front bumpers.

as the reciprocal of the sum of the average gap s and the vehicle length l_α, resulting in the following expression:

$$s = \frac{1}{\rho} - l_\alpha. \tag{6.8}$$

As previously mentioned, in steady-state, all individual vehicle speeds v_α are equivalent to the microscopic equilibrium speed v_e and, by extension, to the macroscopic speed v. When reformulating the Krauss model (see Eq. 3.11) by (i) replacing its gap function $g(t)$ with the static gap s from Equation 6.8, (ii) substituting all velocity-related terms with the macroscopic steady-state velocity v, and (iii) eliminating all time dependencies, we obtain:

$$\frac{\overbrace{\left(\frac{1}{\rho} - l_\alpha\right)}^{g(t)=s} - v\tau}{\frac{v}{b} + \tau} = 0. \tag{6.9}$$

From this, we derive:

$$v = \frac{1}{\tau}\left(\frac{1}{\rho} - l_\alpha\right). \tag{6.10}$$

Combining this result with the fundamental relationship in Equation 6.4, we derive:

$$\rho = \frac{1}{v\tau + l_\alpha}, \qquad q = \frac{1}{\tau}(1 - l_\alpha\rho), \tag{6.11}$$

illustrating how SUMO's microscopic quantities translate into macroscopic terms under steady-state conditions. Additionally, Equation 6.10 shows that the reciprocal of the reaction time, $1/\tau$, governs the relationship between equilibrium speed v_e and gap distance s. Therefore, SUMO achieves steady-state equilibrium only within a defined velocity-gap ratio. Under these conditions and the established boundary parameters, SUMO's car-following model aligns with MATSim's queue model in steady-state scenarios.

6.2.2 Numerical analysis of transient states on a homogeneous road segment

As discussed in the previous section, MATSim and SUMO show consistent traffic dynamics under steady-state conditions on a homogeneous road segment. However, such conditions are of limited practical relevance in real-world applications. Therefore, we now turn our attention to the non-stationary (transient) case through numerical simulations. In this scenario, macroscopic state variables vary over time, and both simulation frameworks account for non-linear effects such as vehicle interactions and heterogeneous vehicle-driver units.

Our investigation begins with a simplified roundabout scenario (see Figure 6.1), using the default settings for both MATSim and SUMO. The roundabout consists of four edges arranged in a diamond shape, each with an access and exit ramp at the cardinal points. All road segments are 1000 meters long, single-lane, and have a speed limit of 50 km/h to simulate urban driving conditions. The simulation starts by introducing 1200 agents (300 per on-ramp), evenly distributed along the far end of the network in both MATSim and SUMO. Once inserted, the agents must complete three-quarters of the roundabout before exiting. This setup gradually leads to congestion, particularly on the target road (highlighted in orange in Figure 6.1). To illustrate the transient behavior in both frameworks, we capture macroscopic flow, density, and speed at various time points along the inner roundabout edge.

The results are depicted in Figures 6.2 and 6.3. Each column of panels presents the outcomes for different parameter settings within the SUMO and MATSim simulation environments. The rows, on the other hand, illustrate the relationships between traffic flow and density, speed and density, and speed and flow, respectively.

The leftmost panels in Figure 6.2, labeled **A1** to **A3**, display the simulation results from SUMO and align reasonably well with empirical real-world observations. To clarify the temporal evolution, successive data points are connected by gray lines. Panel **A1** illustrates a linear increase in traffic flow as density rises, representing the free-flow regime. Once the system reaches its ultimate capacity – approximately 1400 vehicles per hour, at a critical speed of around 25 km/h – traffic flow starts to decline, signaling the onset of congestion. A hysteresis

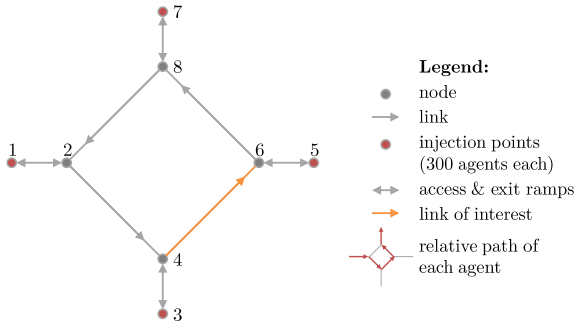

Figure 6.1: Schematic representation of the roundabout scenario design. Starting from nodes 1, 3, 5, and 7, a total of 1200 agents (300 per on-ramp) are evenly distributed across the network. Each agent then completes three-quarters of the roundabout circuit before being allowed to exit. This leads to increasing congestion along the target road (highlighted in orange). At the junctions where feeder roads meet the inner roundabout edges (nodes 2, 4, 6, and 8), MATSim assigns equal priority to all directions. In contrast, SUMO uses zipper merge configurations at opposing junctions (nodes 2 and 6) to facilitate vehicle entry, while prioritizing inner roundabout edges at nodes 4 and 8 to prevent gridlock.

effect becomes apparent as traffic stabilizes after congestion: before the traffic breakdown, the flow peaks at 1400 vehicles per hour at a density of around 55 vehicles per kilometer, but after the breakdown, traffic flow remains below 1000 vehicles per hour at similar densities.

In the middle panel **A2**, a decrease in space-averaged velocity with increasing traffic density is observed, which aligns with expectations. At the start of the simulation, drivers are largely unaffected by the presence of other vehicles, allowing them to travel at their maximum speed. However, the average free-speed velocity on the road segment remains around 40 km/h, which is below the legal speed limit of 50 km/h due to the need for vehicles to decelerate before making a 90-degree turn onto the next road. The yellowish outlier in the low-density region, visible in panels **A2** to **A3**, is attributed to the roundabout configuration. Towards the end of the simulation, as the observed road segment nears emptiness, platoons of vehicles from low-priority access roads arrive in bulk, leading to artificial traffic jams at the end of the link.

Understanding the traffic dynamics in MATSim is a more complex task, requiring the investigation of various simulation scenarios to pinpoint and explain the

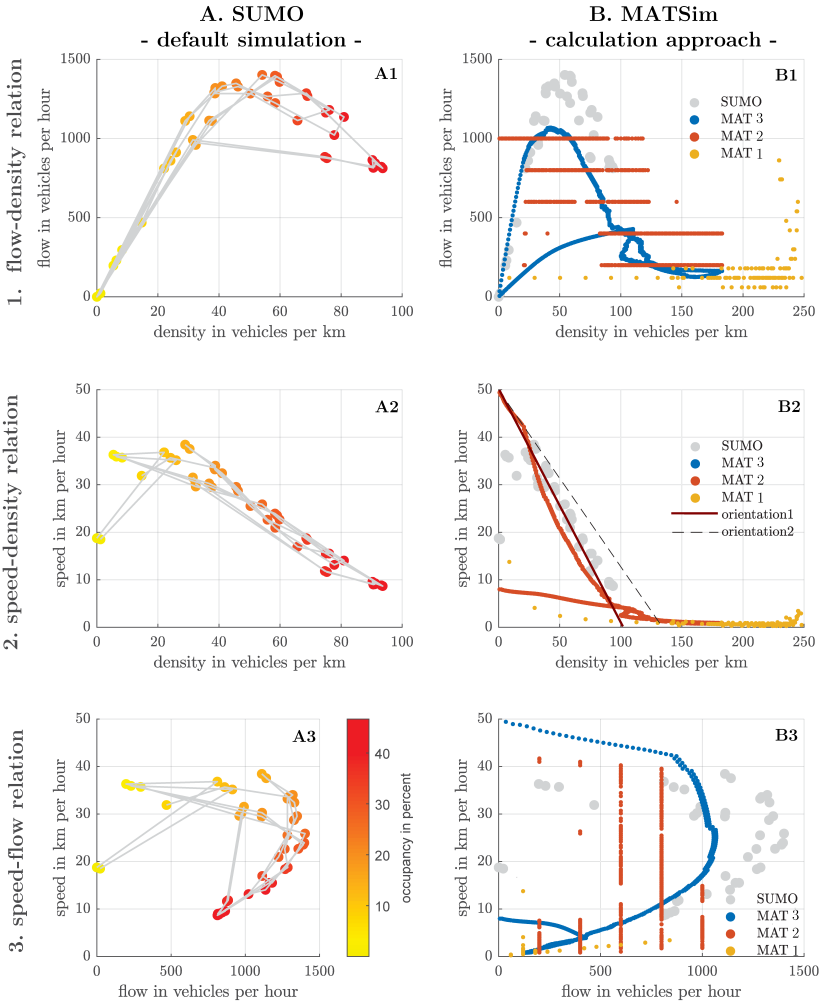

Figure 6.2: Macroscopic flow-density-speed relationships for a homogeneous road segment under transient conditions in the roundabout test case (Part I). Columns show different simulation scenarios for SUMO and MATSim, while rows represent flow-density, speed-density, and speed-flow relationships. SUMO results (gray dots) are included in all MATSim panels for comparison. See main text for details.

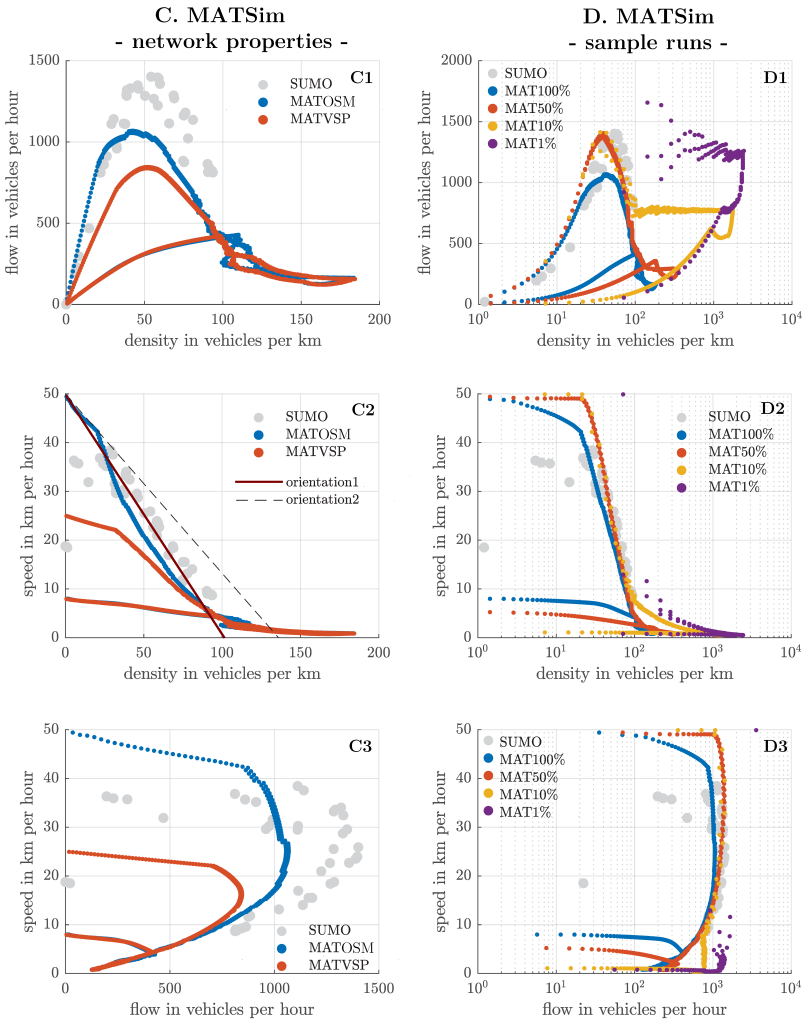

Figure 6.3: Macroscopic flow-density-speed relationships for a homogeneous road segment under transient conditions in the roundabout test case (Part II). Columns show different scenarios for MATSim, while rows represent flow-density, speed-density, and speed-flow relationships. SUMO results (gray dots) are included for comparison. MATSim sample runs (scenario D) have been re-scaled to a 100% sample size and plotted on a logarithmic scale. See main text for details.

sources of certain artifacts. **Scenario B** is crafted to highlight the influence of different data aggregation techniques, with significant variations in macroscopic quantities depending on the chosen methodology.

Initially, we aimed to calculate the time-mean (local) traffic flow, along with the space-mean (instantaneous) density and velocity within a 1-minute aggregation interval. However, the results, represented by the yellow line (MAT1) in panels **B1** to **B3**, proved of limited value. Specifically, the traffic density was significantly skewed by the discrete nature of the queue model and the activation of the *stucktime* parameter (see Section 2.5.2) during congested conditions. This parameter overrides the storage capacity constraint of downstream links when the leading vehicle in the queue remains stationary for an extended period (typically 30 seconds). As indicated by the yellow data in panel **B2**, it seemed that a 1-kilometer road segment could hold up to 250 vehicles. However, technically speaking, only 133 cars – each with a length of 7.5 meters – could physically fit on such a link if packed bumper-to-bumper (refer to orientation line 2 in panel **B2** of Figure 6.2). More realistically, accounting for an additional vehicle gap of 2.5 meters, a 1-kilometer link would accommodate only about 100 vehicles (see orientation line 1).

To mitigate spatial distortions, the reddish line (MAT2) employed a smoothing moving average with a window size of 5 minutes. However, this adjustment did not entirely eliminate the pronounced discretization effects. In contrast, the bluish line (MAT3) applied the fundamental hydrodynamic relationship $\rho = Q/V$, where the flow was computed based on the space-mean velocity and smoothed density at one-second intervals. Although not entirely without limitations, this approach provided the most interpretable data and was subsequently adopted as the preferred aggregation technique for all remaining panels.

In **scenario B**, the analysis of traffic dynamics reveals key differences between MATSim and SUMO:

1. **Ultimate Capacity**: SUMO shows a higher ultimate capacity, around 1400 cars per hour, compared to MATSim's 1050 cars per hour. This difference arises because SUMO calculates link capacities as simulation outputs, while MATSim uses predefined capacities (set by OSM import heuristics), limited by the flow capacity parameter (1000 cars per hour in this case). A MATSim capacity closer to 1500 cars per hour might better represent the road type.

2. **Free-Flow Velocity**: MATSim achieves a higher free-flow velocity of 50 km/h, compared to SUMO's 40 km/h. MATSim agents can reach the legal speed limit even on sharp curves, while SUMO accounts for necessary deceleration before turns.

3. **Recovery from Traffic Breakdown**: MATSim shows slower recovery from traffic breakdown compared to SUMO. This may be attributed to the roundabout configuration: as the MATSim simulation concludes and the road segment nears emptiness, platoons of vehicles from low-priority access roads arrive in bulk, creating artificial traffic jams at the end of the link. This is then followed by a sudden halt in vehicle inflow, leaving minimal time for recovery. In contrast, SUMO's gradual vehicle outflow allows for smoother traffic relaxation.

Scenario C illustrates the impact of varying network properties. In its default configuration (referred to as MATOSM in panels **C1** to **C3**), MATSim's simplified node model fails to account for the additional temporal delays associated with accurate intersection representation and traffic signal operations. Consequently, agents in the simulation often travel at speeds that exceed practical limits, adhering strictly to legal speed limits and link capacities provided by network import heuristics. To address this limitation, a global correction known as `VSPAdjustments`[3] was introduced. This adjustment (referred to as MATVSP) involves reducing speed limits by 50% for primary and lower roads and decreasing road capacities by 33% and 20% for primary and secondary roads, respectively. The effects of these modifications are evident in the simulation results, as shown in panels **C1** to **C3**.

In the context of our roundabout test case, MATVSP traffic dynamics exhibit more pronounced deviations from SUMO's behavior, particularly with traffic flow and vehicle speeds being overly pessimistic. These discrepancies may stem from the specific characteristics of the roundabout scenario. However,

[3] The `VSPAdjustments` in MATSim (see Section 4.2.2) were introduced by the MATSim community to align simulated travel times with real-world observations. MATSim simplifies vehicle behavior by assuming constant speeds, leading agents to travel at the maximum speed allowed by the speed limit under free-flow conditions. This simplification, while computationally efficient, does not capture the reality of urban driving, where drivers must navigate right-of-way rules and other traffic conditions. The `VSPAdjustments` address this by implementing reduced speed limits for urban roads and adjusting road capacities, thereby enhancing the realism of MATSim's traffic simulations.

it is uncertain whether these differences will persist in more complex and realistic urban environments, which feature a broader range of road types, varied right-of-way rules, and diverse infrastructural elements. This issue will be further explored in the upcoming real-world test case in Section 6.2.3.

Scenario D explores the impact of MATSim's *sample runs* on macroscopic traffic dynamics. To enable meaningful comparisons, traffic flow and density have been rescaled to represent 100% samples. Generally, simulations with lower sample proportions, such as MAT1/10/50%, yield results that closely resemble those from SUMO, up to a traffic density of approximately 100 cars per kilometer. Beyond this density threshold, MATSim's traffic dynamics, as shown in panels **D1** to **D3**, increasingly diverge from realistic patterns as sample proportions decrease. Notably, the pronounced "nose-like" shapes at the right end of each curve in **D1** are largely due to the *stucktime* parameter. This parameter distorts both traffic flow and density, with its effects becoming more severe at lower sample rates. For example, at a 1% sample rate, the traffic densities become unrealistically high, reaching approximately 2500 cars per kilometer, as shown in **D2**. For 100% sample simulations, these artifacts are minimal. However, as previously noted, MATSim cannot achieve the same ultimate flow rates as SUMO.

Additionally, it is crucial to acknowledge that under congested traffic conditions, the *stucktime* mechanism significantly impacts traffic flow. In situations of complete gridlock, vehicles with a weight[4] of 100 are advanced to the next link every 30 seconds. This behavior results in an ultimate capacity of 12,000 vehicles per hour for a 1% sample run. This issue is partly visible in panel **D3**, where the violet curve in the upper right corner shows a flow of approximately 3,500 vehicles per hour. Panels **D1** and **D3** have been truncated to exclude even higher flow rates for clarity. The synthetic roundabout test case reveals significant congestion within the MATSim framework. The impact of the *stucktime* parameter on MATSim's flow in more complex and realistic scenarios warrants further investigation and will be addressed in Section 6.2.3.

In summary, MATSim and SUMO traffic dynamics are comparable under free-flow and moderately congested conditions when MATSim uses 100%

[4] In a 1% sample run in MATSim, each vehicle represents 100 vehicles, leading to an effective net space occupancy of 750 meters.

sample runs. However, achieving this similarity requires adjusting MATSim's link capacities to align with those in SUMO. One approach to achieve this adjustment, for instance, is detailed in [180]. Moreover, employing a 100% sample in MATSim is impractical due to the loss of computational efficiency and the substantial effort needed to model such traffic demands. Conversely, in heavily congested conditions, MATSim and SUMO show significant differences. Here, the *stucktime* parameter in MATSim dominates traffic flow, leading to unrealistically high traffic densities on downstream links, especially with lower sample proportions.

6.2.3 Numerical analysis of non-stationary states in an urban real-world test case

In our real-world test case, we begin with the 10% MATSim *Open Berlin Scenario* [234] as the basis for our analysis. A medium-sized inner-city area is isolated from this scenario, as shown in Figure 6.4, and the corresponding road-based travel demand is transferred to a SUMO model, utilizing a network imported from OSM data. For simplicity, we chose not to include SAEV fleets in our current analysis, as our investigation focuses specifically on network-wide traffic dynamics. Comprehensive simulations involving SAEV operations have been conducted, among others, through collaborative efforts in the publicly-funded research project *PAVE – Potentiale Automatisierter Verkehrssysteme*[5] [120].

The *Open Berlin Scenario* used in this study underwent a calibration process comprising two main steps. First, calibration was based on traffic counts by adjusting activity locations with the CaDyTS framework, as detailed in [77]. Second, mode choice calibration involved refining departure times, vehicle routes, and mode-specific utility constants through a manual calibration procedure. This process included several hundred iterations to achieve a quasi-stable user equilibrium, similar to a Nash equilibrium [165, 164, 61]. To improve travel time realism, `VSPAdjustments` were applied to account for additional delays from factors such as traffic lights, lane changes, and right-of-

[5] PAVE – Potentiale Automatisierter Verkehrssysteme https://pave-your-way.de [120]. Accessed: Sep. 15, 2022.

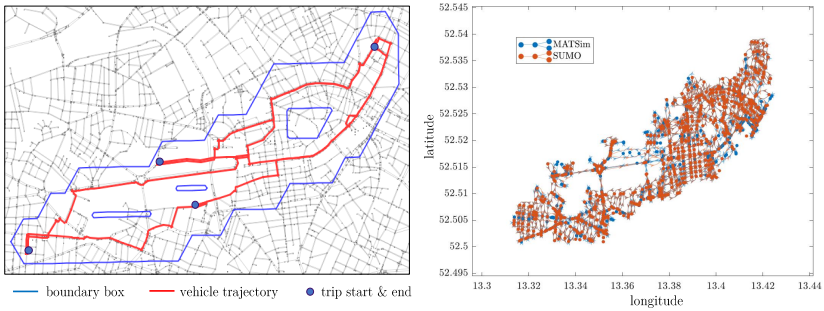

Figure 6.4: The *left* panel shows the configuration of our Berlin test case, with the red line indicating a sample trajectory of an selected ego-vehicle. This trajectory defines the network boundary, marked by the blue outer contour. The *right* panel presents MATLAB digraph representations of the networks in MATSim and SUMO, adapted from [208]. In total, the MATSim network comprises 2318 links and 1204 nodes, while the SUMO network includes 1538 edges and 699 nodes.

way rules not explicitly modeled in MATSim. A more detailed description of the calibration process is provided in [234].

The SUMO model underwent a different calibration procedure, as detailed in Section 7 and pre-published in [208]. This calibration focused on adjusting factors affecting travel demand, routing decisions, and congestion to align with the MATSim model, with a particular emphasis on travel times. Our objective was to ensure that SAEV trips from the MATSim simulation are temporally consistent with the SUMO simulation, where fleet vehicles are expected to reach their designated customers and charging stations on schedule.

We like to stress, that we did not apply identical vehicle routing strategies in both MATSim and SUMO, as initially claimed in Section 5.3. This approach led to significant gridlock issues in the more congestion-prone SUMO simulation, as will be demonstrated in Chapter 7. Instead, we permitted ad-hoc re-routing to achieve a state approximating user equilibrium in SUMO. After calibration, any discrepancies in individual trip travel times and distances between MATSim and SUMO were found to be negligible for our specific use case.

In the subsequent analysis, we focus exclusively on the network-wide (global) density-flow-speed relationships, as illustrated in Figure 6.5. Compared to the

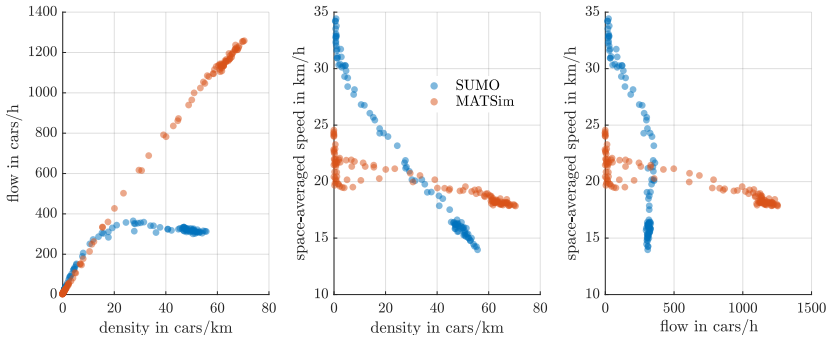

Figure 6.5: The panels, arranged from left to right, display the network-wide relationships between flow and density, speed and density, and speed and flow over a complete day in the Berlin test case. Each data point represents a 15-minute interval. MATSim density is calculated using time-mean flow and space-mean velocity to mitigate visual distortions caused by the stucktime artifacts.

simplified roundabout test case, the traffic dynamics in the real-world scenario reveal more pronounced disparities.

The most significant difference is observed in macroscopic traffic flow, as shown in the left panel of Figure 6.5. Despite accounting for the capacity-reducing effects of the `VSPAdjustments`, MATSim's maximum observed traffic flow exceeds the corresponding value in SUMO by a factor of three. MATSim remains within the free-flow regime throughout the day, while SUMO transitions to a congested regime at a traffic density of approximately 30 vehicles per kilometer. Specifically, around 15% of the SUMO agents are removed from the simulation due to heavy congestion on major roads, which limits available space for vehicle insertion. This discrepancy suggests that MATSim's network elements have higher flow capacities and greater resilience to congestion – a phenomenon examined in more detail in Section 6.3. Additionally, MATSim's traffic densities appear more realistic when observed at the network-wide level compared to a link-wise representation, as the *stucktime* artifacts are smoothed across space.

However, it is important to note that MATSim's storage capacity constraint is frequently violated, as illustrated in Figure 6.6. Each time a vehicle enters a link in the Berlin test case, we record the link length and the number of queued

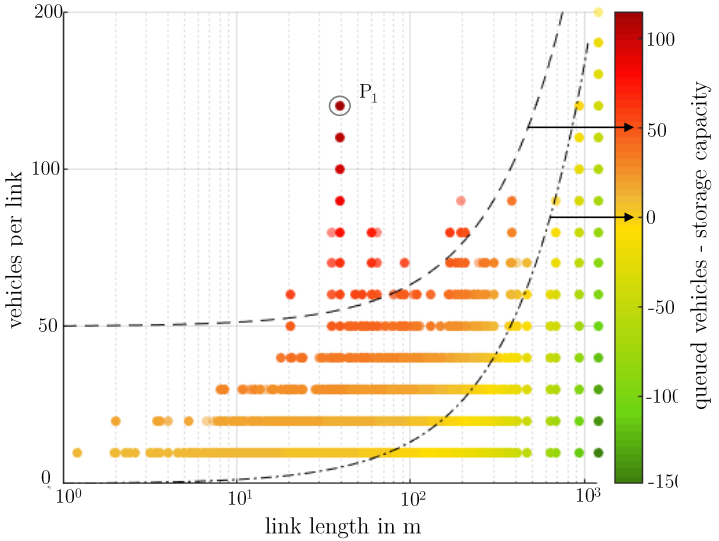

Figure 6.6: Density-related *short-link artifacts* encountered in MATSim simulation, illustrated by the 10% Berlin test case (rescaled to 100%). The figure shows storage capacity violations as a function of link length, with each data point representing a link-entering event. In total, over 1.285 million link-entering events were recorded. For each event, the number of queued vehicles upon link entry is compared to the link's actual storage capacity. A dot-dashed line distinguishes between realistic queue lengths (yellowish-green) and dubious ones (orange-reddish) where queue lengths exceed the link's physical length. A dashed line indicates an excess of 50 vehicles. In a 10% sample run, only links longer than 75 meters can reasonably accommodate vehicles (one vehicle of weight ten occupying a net space of 10×7.5 m). The artifacts become more pronounced with lower sample rates. The most critical event, P_1, shows 120 excess cars (a 900 m vehicle queue bumper-to-bumper) on a 35-meter link.

vehicles. In the figure, each data point represents a specific link-entering event, with color indicating the discrepancy between the actual number of vehicles on the link and its theoretical storage capacity. Data points highlighted in red or orange indicate vehicle excess, while those in yellow or green reflect available capacities. A dashed-dotted line represents a fully occupied link, and a dashed line marks an excess of 50 vehicles. Particularly noteworthy is event P_1, where 120 excess cars – equivalent to a 900 m queue bumper-to-bumper – accumulate on a single link just 35 meters long. Therefore, the *stucktime* parameter is a key

factor in MATSim's inability to accurately model spatial spillback effects on upstream links. On shorter links, this parameter acts as a synthetic vehicle sink, intensifying the observed discrepancies in MATSim and SUMO.

The speed-density relationship shown in the middle panel of Figure 6.5 supports the findings from the roundabout test case. Specifically, the `VSPAdjustments` in MATSim lead to significantly lower speeds in low-density areas compared to SUMO. However, MATSim's space-averaged speeds decline only slightly as density increases, whereas SUMO shows a sharp decrease. Beyond a density of approximately 33 cars per kilometer, MATSim agents generally travel faster than their SUMO counterparts. This discrepancy occurs because, under high congestion (likely due to signalized intersections), time delays become excessively high in the microscopic traffic simulation. Despite these differences, travel times over the course of the day in SUMO are consistent with those in MATSim, as both frameworks have been calibrated to align in this respect (see Chapter 7).

However, similar to the density-related artifacts (see Figure 6.6), travel times (and by extension, velocities) are also distorted in MATSim at the link-wise level, as shown in Figure 6.7. This plot displays link travel times in relation to link length and link occupancy (number of queued vehicles). Generally, it is observed that shorter links with longer vehicle queues experience more pronounced restrictions due to MATSim's *flow capacity* constraint[6], leading to a less favorable ratio of actual travel time to free-flow travel time.

Consider a link with a nominal *flow capacity* of 900 vehicles per hour. The minimum time headway between successive vehicles at the link exit is 4 seconds, 40 seconds, and 400 seconds for 100%, 10%, and 1% sample runs, respectively. Consequently, in MATSim, potential time losses accumulate predominantly on shorter links, particularly at lower sampling rates. The shorter the link, the more pronounced the impact of the imposed waiting times due to flow capacity constraints relative to the link's free-flow travel time. An extreme example is the event (P_2), where a vehicle enters a 3.5-meter-long link and experiences a link travel time approximately 112 times longer than its free-flow passing time.

[6] To recall, the flow capacity constraint operates analogously to a queueing system, permitting vehicles to exit the link (one at a time) at intervals of $\frac{3600}{f_f}$ seconds only, where f_f denotes the link's flow capacity in vehicles per hour; see Section 2.5.1.

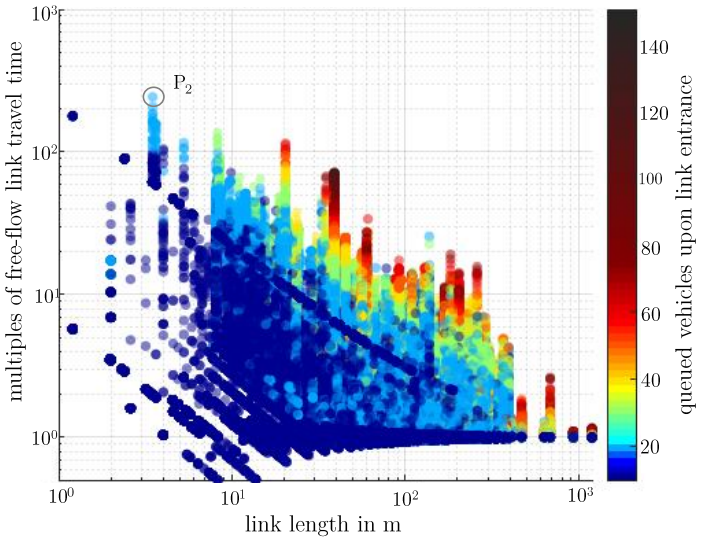

Figure 6.7: Flow-related *short-link artifacts* encountered in the MATSim simulation, exemplified by our 10% Berlin test case (rescaled to 100%): The graph illustrates link travel times, normalized relative to their free-flow travel times, as a function of both link length and occupancy (number of queued vehicles). Notably, normalized travel times increase as link length decreases. This effect is caused by the *flow capacity* constraint, which only allows vehicles to exit a link when the capacity has not yet been fully utilized. Shorter links are more prone to congestion due to frequent violations of the storage capacity constraint, resulting in significant time delays as MATSim attempts to process all queued vehicles. An extreme example is the event (P_2), where a vehicle entering a 3.5 m link experiences a travel time 112 times longer than its free-flow passing time.

In contrast, longer links generally maintain free-flow conditions, as vehicle queues do not propagate effectively upstream due to the density-related artifacts caused by the stucktime mechanism, as previously discussed.

6.3 Comparative analysis of junction flow rates in Berlin – a numerical examination

6.3.1 Comparison of theoretical and observed junction inflows in MATSim and SUMO

As demonstrated in Section 6.2.3, MATSim consistently exhibits significantly higher network-wide traffic flows compared to SUMO. To investigate whether these discrepancies arise from the absence of appropriate network resistances in MATSim, we examine key contributing factors. Notably, MATSim neglects important flow-reducing elements such as right-of-way rules, lane-change maneuvers, deceleration due to network geometry, left turns in the presence of multi-lane oncoming traffic, and the impact of low-power vehicles within First-In-First-Out (FIFO) queues. While the general link capacity reduction of 20–33%, implemented via the `VSPAdjustments`, addresses some of these issues, it may not be sufficient to compensate for all the omitted flow bottlenecks.

For a quantitative assessment, we focus on junction flow rates, as deviations are anticipated to be more pronounced at this level compared to individual links. We assume that SUMO's junction flow rates primarily determine the overall network flow, with right-of-way rules and traffic signals serving as limiting factors. In total, we examine 680 distinct junctions within the Berlin test case, as depicted in Figure 6.4, and conduct a comparative evaluation of their inflow[7] rates in both MATSim and SUMO. The junctions vary considerably in geometric complexity, ranging from simple four-armed intersections to intricate, asymmetrical configurations with multiple lanes. Additionally, the nodes are subject to individual right-of-way rules, which contribute to the diversity and comprehensiveness of our analysis.

Figure 6.8 provides a visual representation of the outcomes from this examination. In the figure, each blue dot represents an individual junction and denotes the maximum observed inflow rate recorded across all 15-minute intervals of the day in both SUMO and MATSim. It is important to note that temporal

[7] The choice to focus on inflow rates rather than outflow rates is a personal preference, as both metrics are quantitatively identical aside from the temporal delay.

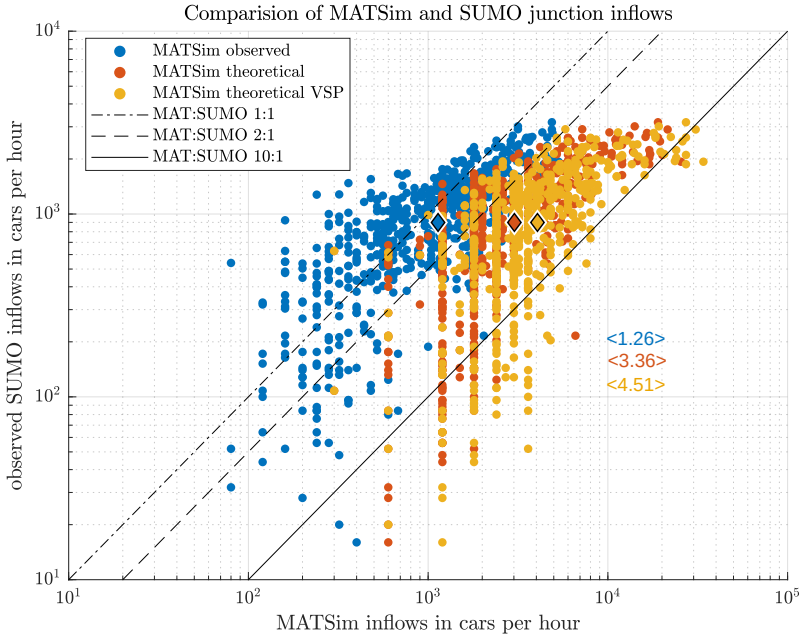

Figure 6.8: Numerical observations of the maximum junction inflow rates in SUMO, compared with those obtained from MATSim simulation, depicted as blue dots. Additionally, theoretical junction flow rates (extracted from the network file) are presented as yellow and red dots, representing potential scenarios with MATOSM and MATVSP network properties, respectively. Each dot on the graph corresponds to an individual junction within the Berlin test case. Diamond symbols represent the mean values of the respective datasets, providing a summary of the central tendencies. The values enclosed within angle brackets ("< >") denote the ratio of the mean MATSim flow rates to those observed in SUMO. The graph is further supplemented with dashed-dotted, dashed, and solid lines, which delineate the points at which the flow ratios between MATSim and SUMO equal 1, 2, and 10, respectively. These lines serve as reference indicators for the comparison of network performance between the two simulation frameworks.

alignment of the maximum flow rates between SUMO and MATSim is not a required condition for the data points.

On average, the junction throughputs in MATSim exceed those in SUMO by a factor of 1.26, as indicated by the bluish diamond symbol, which is positioned on the lower right side of the 1:1 auxiliary line. Junctions located on this line

exhibit identical flow rates in both MATSim and SUMO. Data points located to the left of the line indicate higher capacities in SUMO, while data points to the right of the line signify higher capacities in MATSim.

Furthermore, the yellowish and reddish datasets offer insights into MATSim's theoretical junction flow capacities with MATOSM and MATVSP network properties, respectively. These values are not derived from simulations but are instead obtained from the flow capacity attribute within the network file. Specifically, only the capacities of the links contributing to the inflow at a junction are aggregated.

It becomes apparent that the observed junction flow rates in MATSim do not fully utilize their maximum potential, which aligns with the overall free-flow driving conditions observed within the mesoscopic framework. Theoretically, MATSim's junction flow rates, depending on the network import settings (MATOSM or MATVSP), exceed SUMO's capabilities by a factor of 3.4 or 4.5 on average. Interestingly, the MATVSP run demonstrates the highest junction capacities, despite the initial intent of `VSPAdjustments` being to slightly reduce mesoscopic traffic flow to compensate for the absence of right-of-way rules and traffic signals (which are not accounted for by MATSim). Notably, the link capacities for the MATVSP scenario initially appear to decrease by 33% and 20% for primary and secondary roads, respectively. However, this reduction is offset by an exception where capacities are subsequently doubled for links shorter than 100 meters. Consequently, this results in a capacity increase of 33% for short primary roads and a remarkable 60% for short secondary roads compared to MATOSM settings. Given the prevalence of short links in the Berlin test case, it is plausible that the `VSPAdjustments` contribute to the higher junction flow rates observed.

However, it is important to note that visual inspections during SUMO simulations indicate that not all SUMO edges reached their maximum capacity either, despite the network experiencing significant congestion during daytime hours. Therefore, accurately deriving the maximum flow rates in SUMO from this analysis is challenging, as low junction capacities often result from upstream bottlenecks and are not directly related to the junction's intrinsic flow rate.

6.3.2 Analysis of simulated junction inflows by junction type

In this section, we delve deeper into the examination of maximum flow rates for various junction types. While network-wide traffic dynamics indicate that MATSim generally has higher junction capacities, the presence of blue data points in the upper left corner of the 1:1 auxiliary line in Figure 6.8 suggests that there are still many junctions where SUMO's flow rates exceed those observed in MATSim. This raises the question of whether certain junction types are more favorable to either MATSim or SUMO.

To address this question, we have enhanced our analysis by incorporating additional information about the right-of-way rules associated with each node. The updated scatter plot in Figure 6.9 a) now reflects these details. Each data point represents an individual junction, with the color denoting the specific right-of-way rules governing that junction. A preliminary examination reveals that junctions controlled by traffic lights, indicated by yellow dots, appear to have higher capacities in MATSim. For other junction types, the balance shifts toward MATSim for major, multi-lane junctions (of high capacity), while minor junctions are more evenly distributed between the two simulations.

These initial observations are supported by the histograms presented in panels b) to e) of Figure 6.9. Panel b) illustrates the distribution of all 684 junction inflow rates, irrespective of their right-of-way rules. It shows that SUMO has a larger number of junctions with flow rates below 1500 cars/h, while MATSim features more nodes with higher capacities. On average, MATSim's flow rates exceed those of SUMO by approximately 240 cars/h.

Panels c) to e) display kernel-smoothed distributions (KSD) for each specific junction type. For minor junctions governed by right-before-left rules, SUMO's flow rates tend to be higher, likely due to MATSim's reduced free-flow velocity and capacities resulting from the MATVSP network settings. A different pattern emerges for more heavily trafficked priority and traffic light-controlled junctions. In these cases, SUMO vehicles frequently experience delays, whether from yielding to high-priority edges or waiting at traffic lights. Conversely, MATSim appears to have fewer constraints on velocity and capacity in these scenarios and seems to perform better under denser traffic conditions.

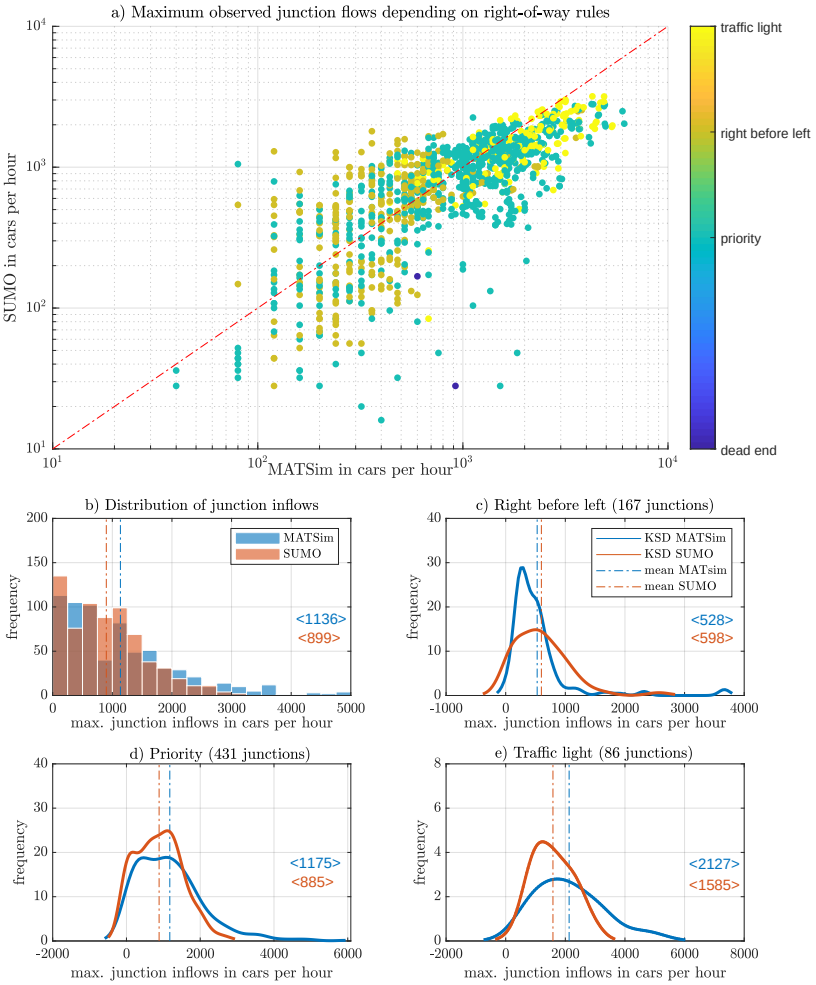

Figure 6.9: Panel *a)* presents a comparison of the maximum observed daily junction flow rates in the Berlin test case between MATSim and SUMO, categorized by different junction types as indicated by the color coding. Panel *b)* provides an overview of the flow distribution across junctions, regardless of specific right-of-way rules. Panels *c)* to *e)* display kernel-smoothed flow distributions for right-before-left, priority, and traffic-signaled junctions, respectively. The mean values, enclosed within angle brackets ("< >"), are visualized by dash-dotted lines.

Based on these findings, we assume that right-before-left junctions in SUMO are likely to experience higher vehicular traffic throughout the day compared to their MATSim counterparts. This expectation arises from two factors: (i) SUMO agents have the flexibility to re-route and bypass potential artificial network bottlenecks, which are often present in imperfect (uncleansed) networks, and (ii) SUMO is required to handle the demand generated by MATSim, and directing more traffic through these minor junctions may help SUMO manage this demand more effectively, thereby relaxing congestion at other hotspots.

6.4 Implications for further toolchain design

Our analysis has shown that traffic dynamics in MATSim and SUMO are consistent within a narrow range, corresponding to steady-state equilibrium conditions on homogeneous road segments. However, a more nuanced picture emerges when examining the idealized transient case, such as the roundabout scenario. In this case, traffic dynamics in MATSim align closely with SUMO for 100% sample runs under free-flow or moderately congested conditions. Yet, under heavily congested conditions, MATSim's traffic dynamics diverge significantly from SUMO. In these situations, the remaining traffic flow is primarily influenced by MATSim's *stucktime* parameter, which leads to unrealistically high densities on downstream links.

We further investigated the impact of various MATSim parameter adjustments, including the introduction of `VSPAdjustments` for network properties and different sample shares. The results provided a mixed assessment. While the MATVSP network properties were ineffective in harmonizing traffic dynamics within the roundabout test case, they did help reduce discrepancies, particularly in terms of average velocity, in the more realistic Berlin test case. For low sample shares, the *stucktime* mechanism led to significant shortcomings in MATSim's traffic dynamics, affecting both the idealized roundabout scenario and the larger Berlin case.

In essence, network-wide traffic flow in MATSim tends to be excessively high, and link-level densities are often unrealistically large. The inflated traffic flow is due to MATSim's limited network resistances, stemming from the lack of right-of-way and traffic signal modeling. The excessive densities result directly from the *stucktime* parameter, which frequently violates the storage capacity

constraint, particularly on shorter links. Consequently, this limitation hampers MATSim's ability to accurately model spatio-temporal congestion patterns.

Furthermore, MATSim's driving conditions display an unrealistic level of stability in the free-flow regime, particularly for 24-hour simulations of real-world urban scenarios, where much more congestion would be expected during peak hours. This issue has also been highlighted by [9]. The authors argue that MATSim agents tend to become increasingly efficient during the iterative process of optimizing their daily activity schedules, striving to reach a user-equilibrium state. In large-scale scenarios, this co-evolutionary traffic assignment process can span several hundred iterations, during which agents gain global knowledge and become adept at avoiding congestion – an advantage that real-world drivers typically lack. As a result, severe congestion is effectively eliminated as the iterations progress.

From these findings, several implications can be drawn for our specific use case of deducing SAEV drive cycles:

(i) Co-simulation of MATSim and SUMO is not a feasible option due to inconsistencies in traffic dynamics and network representation. SUMO would struggle to handle the demand generated by MATSim, as MATSim's mesoscopic road and junction capacities are overly high, and its link-level traffic dynamics are too unrealistic to be meaningfully translated into microscopic traffic simulation.

(ii) Rather than attempting to align network capacities or make MATSim more microscopic in terms of traffic dynamics, it is more pragmatic to accept these non-conformities and structure our final tool-coupling approach around these boundary conditions. Efforts to achieve greater conformity would be resource-intensive and could undermine the framework's inherent strengths in computational performance and scalability.[8]

[8] Although we devised several optimization measures for the MATSim simulation, their implementation was ultimately discarded. Applying corrective measures would require deviating from the calibrated domain of the *Open Berlin Scenario*, and the lack of necessary data precludes any attempt at re-calibration. Thus, adhering to MATSim's default settings is essential. For completeness, we briefly outline the enhancements considered. First, we explored introducing a more realistic *stucktime* period to mitigate simulation artifacts. Additionally, we considered methodologies for tracking sample-share vehicles of varying weights across multiple edges (allowing a single vehicle to occupy multiple edges simultaneously), with the goal of reducing short-link artifacts and supporting spatio-temporal spillback effects on upstream links in MAT-

(iii) As a direct consequence, we should explore additional degrees of freedom tailored to our vehicle-centered use case, even if this restricts the overall applicability of our toolchain. A more flexible sequential tool-coupling approach could be considered, where computationally intensive tasks are automatically assigned to mesoscopic modules (e.g., multi-modal, large-scale fleet analysis), while only relevant, simplified map sections are re-simulated at the microscopic level (e.g., SAEV trajectories). The success of this approach hinges on the condition that our ego-vehicles experience delays similar to those in the travel-time-calibrated MATSim model. Ensuring consistency in traffic performance across models is therefore essential.

(iv) Achieving this requires moving away from strict routing conformity between MATSim and SUMO. Different routing strategies are necessary to counterbalance inconsistencies in traffic dynamics and network representation in order to achieve similar traffic performance. This flexibility would also allow SUMO agents to bypass artificial bottlenecks that may arise from automated network generation. The success of a large-scale, multi-level simulation depends on the model's robustness in handling such imperfections; otherwise, the coupling approach would remain constrained by the limitations of microscopic simulation, where large-scale applicability is hindered by deficient network data quality, necessitating extensive network cleansing efforts.

(v) The primary goal is to ensure consistency in traffic performance, even if it requires relaxing traffic conditions in the more congestion-prone SUMO simulation by simulating less than a full 100% population.[9] In this context, raw vehicle counts are of secondary importance; the key requirement is that the average link speeds match across different hours of the day in both MATSim and SUMO. Consequently, the primary purpose of ambient traffic in SUMO is to introduce delays for our SAEV ego-vehicles that are comparable to those in the travel-time calibrated MATSim model.

Although this chapter did not model SAEV fleet characteristics, our proposed sequential tool-coupling approach can easily be extended to a fleet context. A

Sim. Lastly, we evaluated the approach proposed by [180] to adjust MATSim's link capacities to better align with those observed in SUMO, which more accurately reflect real-world conditions.

[9] As will be shown in Chapter 7, simply applying re-routing strategies cannot resolve this issue. With imperfect networks, it is necessary to reduce the sample size for meaningful SUMO simulation.

significant advantage of our approach is its foundation on MATSim's ability to simulate demand-responsive transport systems, as outlined in Section 2.1.2. This includes various SAEV-related aspects, such as fleet dispatching and rebalancing strategies [140], as well as considerations related to EV range and charging constraints [49, 134]. By mapping MATSim's travel demand to SUMO, SUMO inherently inherits all fleet-related operational characteristics. We can then track the trajectories of fleet vehicles in SUMO to obtain more realistic velocity profiles. Additionally, the behavior of autonomous vehicles can be parameterized using SUMO's car-following model, as proposed in Section 3.2.2, and our calibration procedure detailed in Chapter 7 remains sensitive to different driving styles.

In the following chapter, we will outline the calibration methodology used to synchronize SUMO-generated travel times with those observed in MATSim simulations, ensuring that DRT vehicles transferred across meso-micro boundaries exhibit similar spatio-temporal trajectories. It is important to note that travel times within the MATSim *Open Berlin Scenario* have already been adjusted to align with real-world observations (see Section 6.2.3). Technically, this MATSim calibration applies strictly to the base scenario (see Section 2.2.1) without further modifications. With the introduction of the Demand-Responsive Transport (DRT) travel mode, we enter an uncalibrated domain where recalibration is not feasible due to the lack of real-world data. However, visual inspections have shown that, particularly during daytime hours, the travel times of individual vehicles in MATSim DRT scenarios closely align with the travel time estimates provided by Google Maps[10] (see Figure 7.14). In the absence of better alternatives, we must acknowledge these uncertainties and accept the resulting errors in subsequent SUMO simulations.

[10] Google Maps, https://www.google.com/maps. Accessed: Sep. 29, 2024.

7 Calibrating spatio-temporal network states in SUMO with MATSim observations

In the previous section, we observed significant discrepancies in traffic dynamics and overall network capacity between MATSim and SUMO, highlighting the need for alternative methods to achieve comparable spatio-temporal network states in both frameworks. Our calibration approach is guided by the following principles: (i) MATSim's global travel times have already been calibrated to real-world data. (ii) As a result, we assume that the travel times of mesoscopically simulated DRT vehicles in MATSim are reasonably accurate. (iii) Our objective is to ensure that these vehicles experience similar delays in the subsequent SUMO simulation. (iv) The key challenge is developing effective strategies to replicate these conditions in SUMO, given its imperfect (uncleansed) network.

In Section 7.1, we present a concise overview of various calibration techniques within traffic science, highlighting how target values differ depending on the objectives of the traffic simulation. In Section 7.2, we introduce the core concept of our two-stage calibration approach and describe the test case used to evaluate this method. In the first stage of our calibration process, Section 7.3 explores several strategies to enhance SUMO's global network capacity without the need for manual network adjustments. A comprehensive parameter study is conducted to assess the robustness and sensitivity of these strategies. Building on insights from this study, we formulate a global optimization problem in Section 7.4, aiming to align average edge speeds throughout the day between microscopic and mesoscopic simulations. This optimization problem is solved numerically using a genetic algorithm, and the results are discussed. The contents of these sections have been previously published in [208].

In Section 7.5, we address the remaining discrepancies in traffic performance between MATSim and SUMO, particularly during nighttime. In line with our commitment to report both successful and unsuccessful outcomes, we discuss

the futility of the second phase of our calibration approach in Section 7.6, followed by an alternative solution in Section 7.7 to further align global travel times in MATSim and SUMO. By the conclusion of this chapter, we have developed a functional toolchain to automatically derive microscopic SAEV driving cycles from large-scale transport models in MATSim. The complete toolchain, along with its key components, is detailed in Section 7.9.

7.1 State of the art calibration techniques for traffic simulation

Calibrated traffic simulation models are essential tools for supporting local authorities in traffic management and infrastructure-related decision-making. In recent years, the benefits of traffic simulation have attracted a wider range of users beyond traditional traffic planners and engineers, driving the adaptation of simulation tools for new applications. Notably, the automotive industry has embraced microscopic traffic simulation as a cost-effective method for deriving driving patterns in emerging mobility concepts, enabling virtual requirement engineering and prototyping.

In parallel with this evolution, the calibration requirements for traffic simulation models have expanded. Traditionally, calibration focused on macroscopic parameters such as traffic flow, density, and average speeds. However, emerging (vehicle-centric) use cases now require a more detailed calibration process that incorporates realistic driving behaviors, time-varying congestion, and precise travel time estimations. While traffic simulation was once primarily the domain of traffic engineers, it is now being utilized by a broader range of users, including companies without expertise in traffic or transport engineering. As a result, there is growing demand for highly automated, data-driven, and AI-enhanced methods for model construction and calibration.

Recent advancements in data collection methodologies, particularly the utilization of floating car data and mobile phone data, have emerged as transformative tools in the domain of traffic modeling. These innovations hold the potential to significantly reduce, or eventually eliminate, the reliance on labor-intensive, survey-based travel demand modeling and extensive manual network refinement.

This shift represents a significant leap in traffic simulation, meeting the growing complexity and demands of a broader range of stakeholders.

The calibration of traffic models is crucial for improving the predictive accuracy of traffic simulation tools, as emphasized in [58]. This process is especially important when these models are employed to support local authorities in traffic management decision-making.

SUMO's default calibration procedure is based on SUMO-Cadyts, as discussed in [79]. This method involves an iterative adjustment of route choices and departure times for individual trips to align with measured flow values, typically sourced from detector data. However, several limitations must be acknowledged in this procedure: (i) Like all calibration techniques, the reliability of SUMO-Cadyts is inherently tied to the quality of the underlying data, which is often incomplete. (ii) Within the SUMO simulation, agents generally exhibit less flexibility in responding to congestion compared to real-world drivers. As a result, short teleportation intervals are sometimes utilized as a workaround to mitigate severe gridlocks. Excessively congested networks can lead to an underestimation of measured flows in rush hour simulations, as vehicles may be unable to complete their trips. (iii) Cadyts requires the availability of route choice alternatives; if the simulated demand is too low, it may be difficult to identify alternative routes, resulting in less meaningful calibration outcomes.

A wealth of studies address the calibration of microscopic traffic simulation models, typically involving the adjustment of parameters for car-following and lane-change models to align with desired target values. However, it is important to note that the calibration techniques employed vary significantly, and standardization poses considerable challenges, as outlined in [113]. Several factors contribute to this lack of standardization: (i) Calibration techniques must accommodate a wide range of diverse simulation tools, each with its unique characteristics and requirements. (ii) The target values for calibration are heavily dependent on the specific use case, complicating the establishment of a one-size-fits-all approach. (iii) Some studies advocate for the use of multi-objective or dynamic calibration methods, which consider multiple criteria or adapt to changing conditions, such as time-dependent variations in traffic behavior.

Among the commonly employed calibration techniques, sensitivity analysis and trial-and-error methods, as shown in [175], are frequently utilized, although

they may not always yield optimal results. Alternative approaches include the use of multi-start algorithms, such as in [53], and the application of neural networks, as explored in [105]. Furthermore, genetic algorithms have been applied to identify optimized model parameters, as demonstrated in [51, 115, 148].

The calibration procedure is often divided into distinct phases. For example, in [103], the process starts by comparing simulated and observed global parameters, such as maximum acceleration and other vehicle characteristics, before moving on to calibrating local parameters, like observed speeds. In contrast, [68] begins with the calibration of network-wide capacity to accurately replicate field conditions, followed by refining link-specific capacities to better match capacity measurements at bottlenecks.

Calibrating models based on speed, density, and flow relationships, as shown in [51, 148, 75], provides the advantage of covering a broad spectrum of traffic conditions, from free-flow to congestion. However, despite this potential benefit, the use of fundamental traffic flow relationships in microscopic traffic calibration remains relatively limited [51].

In recent years, there has been growing interest in dynamic microscopic calibration, as it has become clear that static parameters are insufficient to capture the variability in traffic conditions, the diversity of driving behaviors, and the spatio-temporal evolution of network states. Dynamic calibration methods often rely on Kalman filters [174].

Beyond dynamic calibration, the literature also explores multi-objective approaches. A notable study in this area [55] examines computer science and mathematical optimization, offering insights into the performance and efficiency of 17 state-of-the-art multi-objective algorithms for traffic simulation calibration. These algorithms are applied to three test cases of varying complexity.

It is important to note that most calibration studies in the literature focus on small-scale freeway networks, particularly during rush-hour conditions. In contrast, there is a noticeable lack of large-scale studies, especially in urban areas with numerous signalized intersections operating over a full 24-hour cycle. Calibrating such extensive networks poses significant challenges due to the need for diverse data sources, which are often unavailable or lack the required spatial and temporal precision [11]. As a result, large-scale models frequently rely on calibration using traffic flows derived solely from detector data. Obtaining

network-wide aerial imagery to estimate queue lengths or deploying extensive probe vehicles for travel time data collection would be both cost-prohibitive and time-consuming.

As discussed in [11], the calibration of a medium-scale network in New Jersey was successfully achieved by incorporating traffic volumes, travel times, and queue lengths. However, the authors emphasize that this success came with significant resource demands. These were required for (i) network construction, (ii) integrating essential control signs and turning priorities, and (iii) collecting and processing the data needed for calibration and validation. Given these extensive requirements, scaling this procedure to metropolitan networks is impractical. This point is further supported by [113], which highlights the prohibitive costs of data acquisition for calibrating large networks.

Calibrating traffic models based on travel times is not a new concept, but it often overlooks the spatio-temporal evolution over a full 24-hour cycle. Travel time calibration is commonly applied in motorway models due to the availability of various detection methods, such as Bluetooth detectors, automatic plate recognition systems, and floating vehicle data [113].

In traditional traffic planning, congestion and queue-length-based calibration played a relatively minor role, focusing primarily on preventing peak-hour congestion. However, in automotive requirements engineering, especially for scenarios involving component-stressing maneuvers like stop-and-go patterns, accurately representing realistic congestion is crucial. As a result, a faithful representation of congestion patterns throughout the day becomes essential.

In [206], an approach is introduced for calibrating and validating key characteristics of traffic instabilities using velocity time series data from aggregated stationary detector data. This method serves a dual purpose: validating traffic flow models in their ability to simulate the spatio-temporal evolution of congestion, and predicting actual traffic breakdowns.

Some studies explore the characteristics of stop-and-go patterns, such as wavelength and propagation velocity, as shown in [6, 30, 130, 191]. However, only in [206, 231, 232] is the growth rate of these oscillations quantified. While the approach in [206] validates traffic flow models in terms of their ability to accurately simulate congestion dynamics, it does not involve calibrating entire networks or individual roads to specific congestion levels.

Despite the extensive body of prior research on microscopic traffic model calibration, its applicability to our specific use case is limited. Our scenario necessitates dynamic calibration of individual vehicle travel times within large-scale, imperfect networks. To address this challenge, we propose an innovative approach for calibrating spatio-temporal network states in microscopic traffic simulation at a global level (see Section 7.4), and later at a local level (see Section 7.6).

This approach, inspired by automotive considerations, departs from traditional calibration methods that focus primarily on traffic counts. Instead, it emphasizes the spatio-temporal alignment of network states. Achieving accurate travel time matching is critical to ensure that our SAEV ego-vehicles, guided by mesoscopic traffic planning, can efficiently serve their assigned customers on time within the microscopic traffic simulation. The ambient traffic in SUMO serves solely to introduce delays to our ego vehicles as they follow predefined routes, replicating the travel-time patterns established in the calibrated MATSim model. Therefore, the alignment of edge-based vehicle counts is less crucial.

7.2 Methodological approach

We propose a two-stage calibration technique aimed at achieving spatio-temporal alignment between MATSim and SUMO network states. In the first stage, we globally adjust the time-dependent relative mean speeds of all active edges in SUMO, as illustrated on the left side of Figure 7.1. This involves the static adjustment of demand-, routing-, and congestion-related factors in SUMO to align travel delays with those observed in the MATSim simulation. To automate this process, we utilize a genetic algorithm (GA) that modifies configuration parameters for SUMO and DUAROUTER, without altering the settings of the car-following or lane-change models.

Following the global optimization, the study advances to the second stage, which focuses on local optimization. In this phase, the emphasis shifts to refining SUMO's network states in close proximity to our ego-vehicles. Utilizing the globally adjusted network states from the first stage, we develop methodologies for strategically introducing and removing ambient traffic vehicles, ensuring controlled adjustments to achieve similar spatio-temporal trajectories in both frameworks. The primary KPI for this assessment is the conformity of

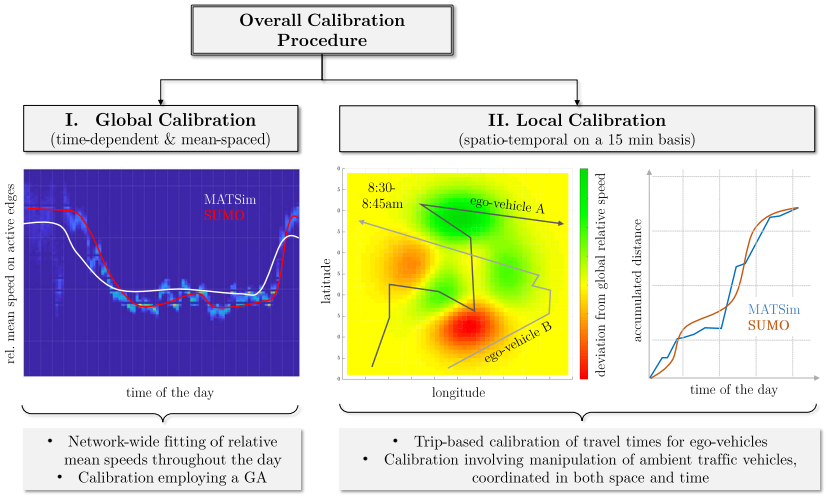

Figure 7.1: Overview of our two-stage calibration procedure. The figure illustrates our approach, which consists of global and local stages aimed at synchronizing spatio-temporal network states in both MATSim and SUMO. In the first stage, global calibration is conducted by fitting relative mean speeds across all active edges throughout the day. In the second stage, local alignment is achieved by adjusting the spatio-temporal trajectories of ego-vehicles in close proximity, involving controlled additions and removals of ambient vehicles.

accumulated distances over time for all ego-vehicles in MATSim and SUMO within 15-minute intervals (see the right side of Figure 7.1). Further details on the local optimization process are provided in Section 7.6.

To evaluate our approach, we use the medium-sized test case introduced in the previous chapter (see Figure 6.4), which focuses on the daily trajectory of a selected ego-vehicle. In a revised version of this illustration (refer to Figure 7.2), we have included additional information about the individual trips made by the ego-vehicle over a 24-hour period. This data is essential for assessing the quality of the calibration routine in Section 7.4.3.

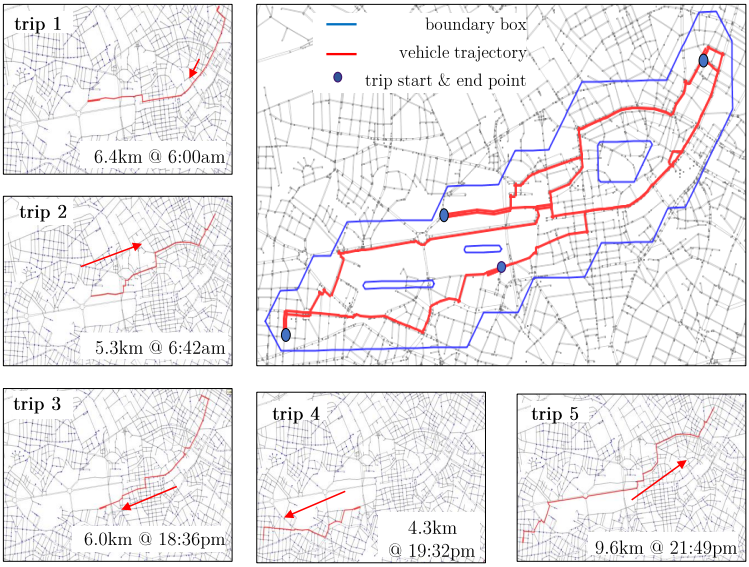

Figure 7.2: Revised illustration of our Berlin test case, highlighting the daily trajectory of the selected ego-vehicle (shown by red routes in the main panel). Additional details regarding the individual trips of the ego-vehicle are provided in the smaller panels, including trip start times and lengths.

7.3 Exploration of parameter space

In this section, we aim to deepen our understanding of how various parameters within the SUMO framework affect network states, including edge-wise travel times and congestion ratios. Our goal is to identify the most impactful factors and evaluate their reliability in stochastic terms. Additionally, we investigate how optimal parameter choices can maximize flow rates within the SUMO network. We have categorized four distinct groups of measures, as illustrated in Figure 7.3, and will focus on a detailed analysis of three of them.

a) Strategies for alleviating local traffic hotspots. The most relevant SUMO parameters for alleviating congestion at hotspots are `time-to-teleport` and `max-depart-delay`. The former defines the maxi-

Category	(A) Hotspot relaxation	(B) Travel demand tweaking	(C) User equil. & routing	(D) Network refinement
Influencing factors	• optimize *teleport* functionality to mitigate congestion • optimize *depart delay* function to facilitate network insertion • ignore junction blockers • suppress u-turns in vehicle routes	• increase MATSim's demand sample • reduce urban link capacities in MATSim • transfer less than 100% of travel demand to SUMO • define and inject artificial demand	• Establish a new user equilibrium with DUAROUTER • allow for adhoc re-routing with varying *rerouting probabilities* and *periods*	• optimize automated network generation • incorporate MATSim travel demand knowledge to enhance SUMO network capacity • create an artificial network to resolve bottlenecks
Assessment	• identify key levers with high expected impact	• MATSim 1% runs are prone to artifacts • determine the feasible demand share SUMO can effectively absorb	• alternative vehicle routing can compensate for differences in traffic dynamics & network repr.	• addressing individual bottlenecks is challenging

Figure 7.3: Main categories outlining various measures to enhance network capacity in SUMO. The following Sections 7.3.1 to 7.3.4 will identify the most impactful and reliable strategies.

mum duration a leading vehicle can remain trapped in front of an intersection before being teleported to the next available edge on its route[1]. The latter parameter specifies the time interval within which a vehicle ready for insertion must reattempt to enter a fully occupied edge before being permanently excluded from the simulation.

The `max-depart-delay` functionality, in particular, is valuable for our specific use case, as it helps mitigate issues associated with agent cloning (the replication of MATSim agents in sample runs to achieve a 100% demand representation in SUMO). Even though we assign normally distributed time offsets to the departure times of these clones, SUMO often struggles to accommodate these new vehicles due to the temporal clustering of multiple insertion-ready vehicles on the same starting edge. Attempts to spatially distribute insertion-ready vehicles more evenly were also unsuccessful. First, particularly in trajectory-

[1] During teleportation, the vehicle's travel time remains representative, as it virtually moves at average edge speed. Similar to MATSim, this process does not capture detailed driving dynamics.

clipped SUMO networks (e.g. , Figure 5.6, panel C4), there were few alternative insertion points available. Second, this often resulted in congestion on minor roads, as vehicles had to take significantly longer detours to rejoin their originally designated routes. With such measures proving futile, the `departure-delay` parameter effectively limits the undesirable side effect of spatio-temporal vehicle clustering during insertion.

b) Strategies for modifying travel demand. In addition to addressing artificial network bottlenecks, SUMO faces significant challenges in accommodating travel demand generated by MATSim. While MATSim's simplified queue model is effective for typical transport planning applications, its limited representation of traffic dynamics (cf. Section 6.2.3) poses difficulties when integrated with microscopic traffic simulations. To mitigate these deficiencies, we explore three potential measures.

First, we analyze the impact of varying MATSim population samples (1% and 10%)[2]. Larger MATSim samples reduce the need for clones, leading to fewer agents with identical routes in SUMO and alleviating potential network load imbalances. Additionally, larger samples improve spatial congestion modeling by being less susceptible to the short-link artifacts mentioned in Section 6.2.3.

Second, we investigate MATSim's `VSPAdjustments`, which are automatically applied during OSM network import to account for reduced inner-city flows and lower speeds caused by signalized intersections. These adjustments significantly change link capacities (see Figure 7.4) and speed limits within the network.

Third, we consider transferring only a specific share of MATSim agents to the SUMO simulation to balance network load. Although this approach may seem unconventional, it is well-suited to our needs, as we aim to achieve comparable traffic conditions and travel times, prioritizing equivalence in these aspects over identical traffic counts.

c) Strategies anchored in routing for efficient traffic management. The third aspect of our analysis focuses on employing various routing strategies in SUMO to optimize network workload distribution. Due to differences in network representation between MATSim and SUMO, as well as additional

[2] While larger population samples (> 10%) are preferable, constraints in travel demand data and computational resources limit this approach.

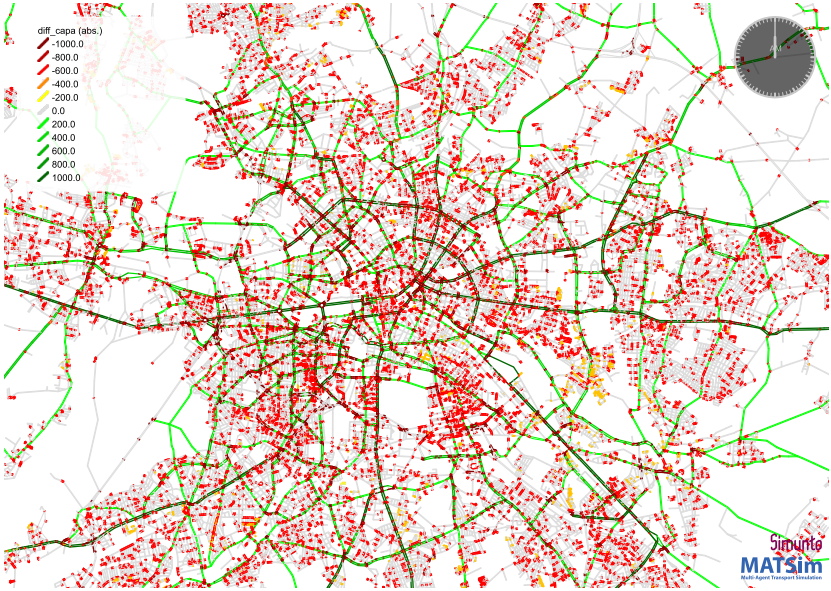

Figure 7.4: Capacity-based outcomes resulting from reversing (!) the default `VSPAdjustments` in the network file of the Open Berlin Scenario [234], visualized using the tool `VIA` (https://www.simunto.com/via/, accessed: Sep. 29, 2024). This visualization shows that road capacities in the original Berlin network (without `VSPAdjustments`) were increased (in green) for primary and secondary roads, but were significantly lower (in red) for short links measuring less than 100 meters. Consequently, we assert that `VSPAdjustments` may actually exacerbate the short-link artifacts discussed in Section 6.2.3 in real-world test cases.

artificial bottlenecks arising from data quality limitations during OSM import, edge-wise network capacities vary. Consequently, MATSim's dynamic user equilibrium does not match SUMO's user assignment equilibrium.

To address this, we explore several strategies: First, we aim to establish a new user equilibrium in SUMO by allowing vehicles to select routes different from those in MATSim, although this option is disabled for ego-vehicles. Second, we implement in-time rerouting in SUMO and analyze its effects on network capacity, varying both the `rerouting-probability` (which changes the likelihood of a vehicle using a rerouting device) and the `rerouting-period` (which sets the time intervals for route reevaluation). Finally, we investigate the

impact of suppressing U-turns at vehicles' starting and ending edges by using the `remove-loops` parameter, aiming to reduce network disturbances.

d) Strategies for refining the SUMO network. Meticulous microscopic network refinement is the most effective method for mitigating artificial network bottlenecks in SUMO. However, as discussed in Section 3.3.5, manual network refinement is not practical for large-scale scenarios such as ours, and improving import heuristics proves to be more complex than initially anticipated. Based on our experience, optimizing network heuristics for specific junction geometries can enhance traffic flow locally, but it frequently leads to performance setbacks in other areas of the network. Additionally, attempts to increase flow rates in SUMO – such as incorporating MATSim's junction flow rates to heuristically enhance SUMO's turning capacities – proved intricate and yielded limited results.

Ideas for creating artificial network elements to bypass major hotspots – such as automatically inserting road segments that do not exist in real life – were also rejected due to (i) the extensive effort required to develop and implement such heuristics and (ii) the unintended consequences of these intrusive measures (e.g., ego-vehicles using these routes). Consequently, we exclude measures related to network refinement in SUMO (see Figure 7.3, column D) and focus instead on the more feasible strategies outlined in paragraphs a) to c).

Overall, we conduct 1,316 SUMO simulations of the Berlin test case to assess the effects of these strategies on the network's performance and their robustness to stochastic variability. Table 7.1 provides a comprehensive overview of all simulations, categorized into nine scenarios with 69 distinct lever values across these scenarios.

7.3.1 Assessing SUMO's maximum travel demand capacity through dynamic user assignment (DUA) sensitivity analysis

Given the significant disparities in network capacity between MATSim and SUMO, expecting similar traffic assignments from these two simulation frameworks is unrealistic. Enforcing MATSim's vehicle routes within SUMO would inevitably lead to severe congestion issues. Therefore, our approach

Table 7.1: Summary of conducted simulations. Scenarios A* and C* focus on the impact of individual levers, whereas Scenarios B* evaluate the synergistic effects of combined levers. Default parameter settings *individual (i)* scenarios: 50% demand share, 300 s `time-to-teleport`, no depart-delay, no rerouting and with route loops. Settings *combined (c)* scenarios (derived from the optimal solutions of individual levers): 60 s `time-to-teleport`, 60 s `max-depart-delay`, `ignore-junction-blocker`, 80% `rerouting-probability`, 60 s `rerouting-period` and `remove-loops`. Except for scenario B3, all SUMO simulations are based on a 1% MATSim sample run. For detailed explanations of specific options, please refer to the glossary provided in the front matter.

Scenario	individual/ combined	no. of levers	detailed lever settings (††)	seed trials	simu-lations	see Sec.
C1 DUA demand sensitivity (†)	i	10	[10:10:100%]	50	326	7.3.1
C2 `rerouting-period`	i	5	[60:60:300s]	30	150	7.3.3
C3 `rerouting-probability`	i	10	[5,10:10:90%]	30	300	7.3.3
C4 `remove-loops`	i	2	[on/off]	30	60	7.3.3
A1 `time-to-teleport`	i	5	[60:60:300s]	30	150	7.3.3
A2 `max-depart-delay`	i	7	[60,300,600,900:900:4500s]	30	210	7.3.3
B1 SUMO demand share	c	10	[10:10:100%]	4	40	7.3.4
B2 `VSPAdjustments` undone	c	10	[10:10:100%]	4	40	7.3.4
B3 MATSim 10% sample run	c	10	[10:10:100%]	4	40	7.3.4
total number	$\sum$	69		$\sum$	1316	

(†) Intentionally, the dynamic user assignment (DUA) scenario C1 included a total of 500 simulations (10 distinct levers with 50 seed trials each). However, due to SUMO's difficulties in accommodating the demand generated by MATSim, the simulations for the 70% demand share had to be terminated prematurely at DUA iteration 27, instead of the intended 50. Simulations for demand shares exceeding 70% were deemed infeasible.

(††) Lever settings are provided in MATLAB notation $[x_1 : i : x_k]$, creating a regularly-spaced vector x using i as the increment between elements.

begins by iteratively establishing a new user equilibrium in SUMO using the `duaIterate.py`[3] script. This script aims to determine optimal routes for all vehicles, ensuring that alternative paths do not result in increased travel costs. The process consists of two main iterative steps: (i) using `DUAROUTER` to identify low-cost routes through shortest path computation and (ii) conducting SUMO simulations to evaluate the actual travel times on these routes. The recalculated edge costs are then applied in the subsequent iteration. A total

[3] "Assignment Tools," Sumo, https://sumo.dlr.de/docs/Tools/Assign.html. Accessed: Sep. 24, 2024.

of 50 dynamic user assignment (DUA) iterations are performed utilizing the *Gawron* route-choice algorithm [85].

Figure 7.5 presents the results of the DUA iterations for various travel demand shares transferred from MATSim to SUMO. This transfer is accomplished by creating an increasing number of copies of the original MATSim agents. The evaluation criterion used is the relative mean speed[4], averaged over a 24-hour period across all active network edges, as illustrated in the main panel.

The results reveal that stable user equilibria are achieved only for travel demand shares below 40%. Demand shares exceeding this threshold do not produce stable outcomes, as demonstrated by the observations in iteration 49 for the yellowish 40% demand and in iterations 33-34 for the orange 50% demand. To enhance convergence, it may be beneficial to explore strategies such as allowing only a fraction of vehicles (approximately $10 - 20\%$) to reroute after each DUA iteration, potentially using parameters like `keep-route-probability`. Additionally, applying smoothing techniques to edge weights across iterations, such as employing the `weight-memory` parameter, could help reduce frequent route changes. However, these specific adjustments were not examined in this study.

As expected, iteration 0, which is based on travel times in an initially empty network, consistently yields the lowest relative mean speed for a given demand, with values ranging between 0.2 and 0.3 for demand shares up to 50%.

Each data point in the main panel of Figure 7.5 can be further dissected into a temporally resolved representation, illustrating the relative mean speed across all active edges throughout the day. Exemplary instances of these temporally resolved representations are provided in the subpanels of Figure 7.5, showcasing distinct network state scenarios.

Subpanel **d** illustrates the ideal scenario, characterized by smooth traffic flow that consistently maintains relative mean speeds near 0.75 during day. During the morning hours, only a few streets are occupied, resulting in a wide distribution of relative mean speeds. Some roads experience free-flow driving conditions approaching a value of 1, while others exhibit moderate speed ratios

[4] In this context, relative mean speed refers to the ratio of the actual mean speed to the speed limit of the road segment.

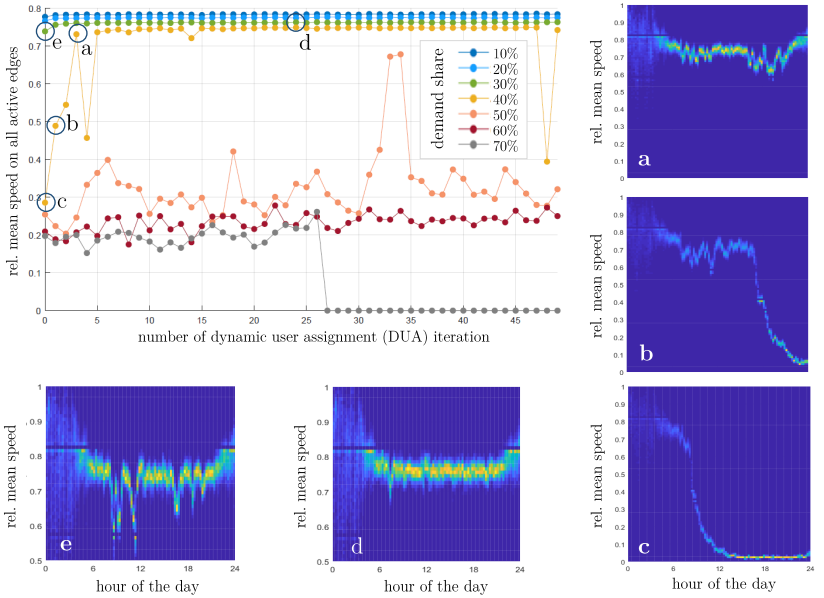

Figure 7.5: The main panel displays the average relative mean speeds in SUMO, calculated over 24 hours across all active edges, for 50 DUA iterations and varying travel demand shares (10% to 70%) in scenario C1. Notably, the simulation for the 70% demand share was halted after the 26th iteration, as further improvements were considered unlikely. The subpanels, labeled from **a** to **e**, illustrate the relative mean speed along all active edges throughout the day for specific simulations, showcasing diverse network states that range from free-flow conditions (as observed in **d**) to slightly congested scenarios (depicted in **a** and **e**) and heavily congested driving conditions (shown in **b** and **c**). The employed color scheme reflects the frequency of specific combinations of daytime and relative mean speed, transitioning from blue to yellow.

in the range of 0.5 to 0.6. This variability may arise from two main factors: vehicles could be delayed by actuated traffic signals during the night, or such delays may result from the cloning process, where a relatively large number of vehicles initiate their journeys on the same edge within a short time frame. In contrast, the simulations depicted in subpanels **a** and **e** demonstrate less stable traffic conditions, with noticeable congestion during the morning and evening rush hours. These network performances align more closely with real-world observations. Subpanels **b** and **c**, on the other hand, showcase scenarios where

the network experiences heavy, and at times unrealistically severe, congestion. These situations are marked by significant traffic breakdowns, with congestion persisting well into the early hours of the next morning (and dissipating only thanks to the teleport option), as agents are not allowed to reroute during these periods.

7.3.2 Evaluation methodology for scenario analysis

SUMO exhibits stability in handling up to 40% of the travel demand transferred from MATSim. Building on this baseline, we systematically investigate the impact of specific parameters aimed at improving SUMO's traffic flow. This is carried out through five distinct scenarios, denoted as $s \in S = \{\text{A1}, \text{A2}, \text{C2}, \text{C3}, \text{C4}\}$, as detailed in Table 7.1. In each scenario, a key parameter is varied across a set of values M_s. For instance, in scenario C2, the parameter of interest is the `rerouting-period`, with multiple lever values ($m \in M_{C2}$ with $M_{C2} = \{60, 120, 180, 240, 300\}$) under investigation.

To thoroughly evaluate the robustness of each scenario (s) and lever (m), we run simulations with 30 different random seeds (r) for each lever. This approach takes into account the natural variability in simulation results, which is influenced by factors like car-following and lane-changing models, as well as individual vehicle behaviors (e.g., whether a vehicle reroutes or exceeds speed limits).

To assess the performance and robustness of each scenario lever (m), we define a scoring metric, θ^m, where higher values correspond to improved traffic performance, indicating enhanced traffic states in SUMO. This scoring metric is defined as:

$$\theta^m = \beta^m_{\text{velo}} + \beta^m_{\text{conv}} + \beta^m_{\text{spread}} + \beta^m_{\text{tele}} + \beta^m_{\text{noin}}, \qquad 0 \leq \theta^m \leq 5. \tag{7.1}$$

Here, each β^m_{ind} represents a sub-score corresponding to one of five different performance indicators, each carrying equal weight. The first term, β^m_{velo}, rewards levers that lead to smoother traffic flows, reflected by higher relative mean speeds on active edges. The next two terms, β^m_{conv} and β^m_{spread}, provide rewards for levers that exhibit faster convergence of simulation results across different seed trials and lower output variability, respectively. The final two

terms, β^m_{tele} and β^m_{noin}, reward simulations with fewer teleported and discarded vehicles.

Let the barred variables $\bar{x}^m$ represent the lever-specific arithmetic mean over all seed trials (r), and hatted variables $\hat{x}$ denote the maximum value of a specific indicator across all lever- and seed-simulations (m and r):

$$\bar{x}^m = \frac{1}{R} \sum_{r \in R} x^m_r, \qquad \hat{x}_{\text{ind}} = \max_{m,r} \left(x^{m,r}_{\text{ind}} \right), \tag{7.2}$$

where R is the total number of seed trials. The velocity-based score β^m_{velo} is then defined as:

$$\beta^m_{\text{velo}} = \frac{1}{\hat{\alpha}_{\text{velo}}} \sum_{t \in T} \sum_{e \in E} \frac{\bar{v}^{m,e}_{\text{act}}}{v^e_{\text{des}}}, \tag{7.3}$$

Here, Equation 7.3 evaluates each simulation based on the relative speed, averaged over 24 hours across all active edges ($e \in E$)[5]. The data for this calculation is extracted from SUMO's `summary-output-file`, which records the average ratio of the actual speed ($\bar{v}^{m,e}_{\text{act}}$) to the desired speed ($v^e_{\text{des}}$) for each 15-minute interval over a day ($t \in \{1, \ldots, T\}$). In this case, the desired speed corresponds to the legal speed limit.

The four remaining sub-scores, β^m_{ind}, can be expressed as follows:

$$\beta^m_{\text{ind}} = 1 - \frac{\alpha^m_{\text{ind}}}{\hat{\alpha}_{\text{ind}}}, \tag{7.4}$$

where the specific expressions for α^m_{ind} are given by:

$$\alpha^m_{\text{conv}} = \sum_{p \in P} \sum_{r=19}^{R-1} \left| F_{r+1}(p) - F_r(p) \right|, \quad P = \{2.5, 25, 50, 75, 97.5\} \tag{7.5}$$

$$\alpha^m_{\text{spread}} = \max_{1 \le r \le 30} \left(\alpha^{m,r}_{\text{velo}} \right) - \min_{1 \le r \le 30} \left(\alpha^{m,r}_{\text{velo}} \right) \tag{7.6}$$

[5] Note that the set of active edges (E) varies over time. For simplicity, we omit the explicit time dependency in the notation.

$$\alpha^{m}_{\text{tele}} = \bar{n}^{m}_{\text{tele}} \tag{7.7}$$

$$\alpha^{m}_{\text{noin}} = \bar{n}^{m}_{\text{noin}}. \tag{7.8}$$

Equation 7.5 assesses the convergence of simulation results with an increasing number of of seed trials. To capture this, percentile functions $F_r(p)$, for $p \in P = \{2.5, 25, 50, 75, 97.5\}$, are calculated based on the daily relative mean speeds for each seed trial 1 up to r. The convergence is quantified by summing the absolute differences between consecutive percentile functions, $|F_{r+1}(p) - F_r(p)|$, for the last 10 seed trials ($19 \leq r \leq 29$) across all percentiles p. A robust lever is expected to produce more stable percentile functions $F_r(p)$ as the number of seed trials increases.

Equation 7.6, used in conjunction with Equation 7.4, rewards levers that demonstrate minimal variability in daily relative mean speeds across all seed trials. The spread is quantified as the difference between the maximum and minimum observed values of $\alpha^{m,r}_{\text{velo}}$ across all seed-simulations (r) for a given lever (m).

As previously mentioned, Equations 7.7 and 7.8, combined with Equation 7.4, reward simulation runs that result in fewer teleported or discarded vehicles. The terms $\bar{n}^{m}_{\text{tele}}$ and $\bar{n}^{m}_{\text{noin}}$ represent the average number of teleported and non-inserted vehicles for each lever-specific simulation, respectively.

To allow direct comparison, all indicators are normalized by their scenario-specific maximum values, denoted as $\hat{\alpha}_{\text{ind}}$. This normalization ensures that the impact of each lever (m) is directly comparable across scenarios. The lever with the highest score, θ^m, is chosen for further analysis in the combined scenarios (referred to as B* in Table 7.1) to assess the upper limit of SUMO's network capacity.

7.3.3 Impact of individual measures on traffic flow performance

This section discusses the impact of the individual measures analyzed in scenarios A* and C* (see Table 7.1), with findings illustrated in Figure 7.6 and Table 7.2. Figure 7.6 presents the robustness of each lever *m* and its effects on three KPIs: (i) relative mean speed (upper panel), (ii) the number of teleported vehicles (middle panel), and (iii) the number of not-inserted vehicles (lower panel), using boxplots. The boxes represent the range of simulated KPI values across 30 distinct random number seeds.[6] In contrast, Table 7.2 utilizes the scoring function (Eq. 7.1) to rank all levers based on their effectiveness and seed robustness, thereby facilitating a clearer comparison among the levers.

Upon initial observation, Figure 7.6 indicates that many simulation results are notably sensitive to variations in random seed. Only scenario **A1**, with a `time-to-teleport` setting of 60 seconds, demonstrates stable results across all three KPIs despite changes in random seeds. In contrast, other scenarios display fluctuations of up to 0.4 in relative mean speed. Smaller fluctuations are evident in scenarios **C2** (300 seconds), **C3** (5%), and **C4** (70%), though it is important to recognize that these may be influenced by the relatively small sample size.

In fact, we did not expect this level of instability in the stochastic parameters. Relying on such unstable settings during our calibration procedure is not feasible. Furthermore, the DUA simulations indicate that the 50% travel demand share, used as input for all simulations in this section, serves as a tipping point for our test case, where results can change rapidly due to the occurrence of a single event (e.g., a stalled vehicle).

A distinctive characteristic of our test case is the limited availability of routing alternatives, particularly in the left and middle sections of our network (see Figure 7.2). Consequently, we assume that overall network performance is primarily influenced by a few isolated bottlenecks, which are highly sensitive

[6] We acknowledge that the limited number of seed trials (30) may not yield statistically robust conclusions regarding random seed sensitivity. However, despite noticeable fluctuations in some seed-accumulated KPIs (see the red outlier in the upper panel of Figure 7.6), the monotonic trends in median values across each scenario suggest qualitative reliability. Given computational constraints, a substantial increase in seed trials was deemed impractical.

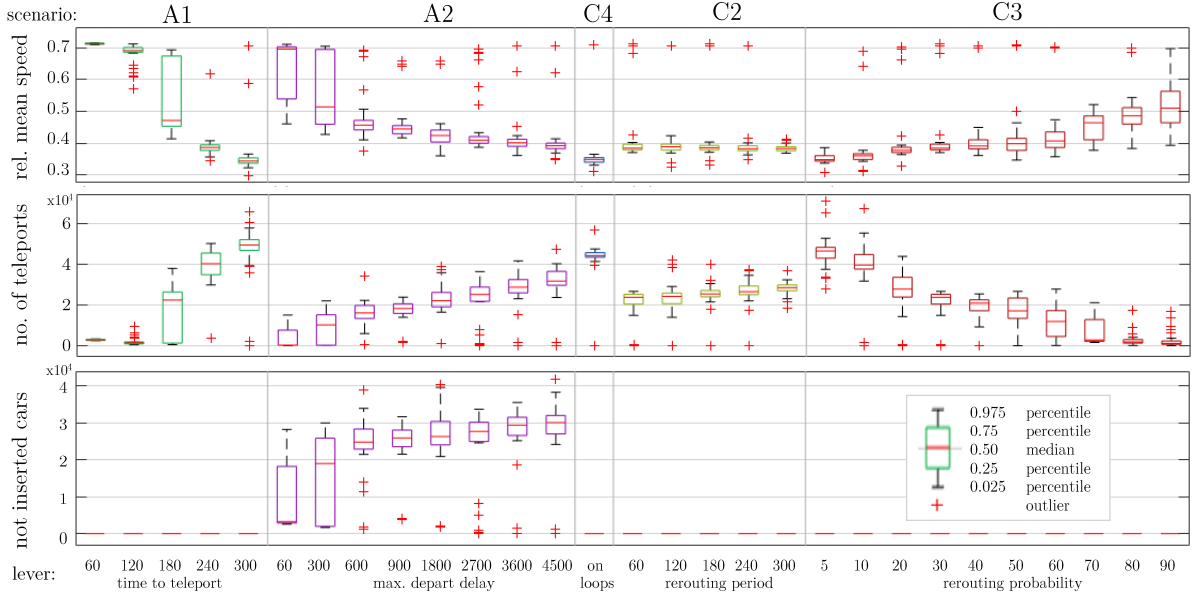

Figure 7.6: Scenario-based visualization of lever seed robustness and their effects on (i) relative mean speed across all active edges (upper panel), (ii) count of teleported vehicles (middle panel), and (iii) number of vehicles not inserted (lower panel). Detailed scenario descriptions can be found in Table 7.1. Each lever undergoes simulation with 30 distinct random seeds.

Table 7.2: Tabulated results of the sensitivity analyses. The ranking of each lever (m) is based on the total score θ^m, comprising five different indicator terms β^m_{ind} as defined in equations 7.2 to 7.8. Levers with the highest scores in each scenario (highlighted in gray) serve as input data for the scenario B* simulations in Section 7.3.4.

scenario	A1					A2								C4
lever m	60	120	180	240	300	60	300	600	900	1800	2700	3600	4500	on
β^m_{velo}	1.00	0.95	0.73	0.55	0.51	0.90	0.79	0.67	0.65	0.60	0.63	0.59	0.57	0.50
β^m_{conv}	0.99	0.81	0.00	0.95	0.97	0.59	0.04	0.60	0.92	0.92	0.42	0.92	0.93	0.96
β^m_{spread}	0.99	0.79	0.23	0.93	0.92	0.34	0.20	0.78	0.88	0.00	0.49	0.89	0.91	0.94
β^m_{tele}	0.94	0.95	0.60	0.15	0.00	0.93	0.81	0.67	0.63	0.51	0.52	0.41	0.33	0.06
β^m_{noin}	1.00	1.00	1.00	1.00	1.00	0.70	0.47	0.19	0.16	0.06	0.18	0.04	0.00	1.00
score θ^m	4.91	4.50	2.56	3.58	3.40	3.46	2.32	2.91	3.23	2.09	2.24	2.84	2.75	3.46
rank	1	2	26	14	18	17	27	23	22	29	28	24	25	16

scenario	C2					C3								
lever m	60	120	180	240	300	10	20	30	40	50	60	70	80	90
β^m_{velo}	0.55	0.57	0.55	0.54	0.49	0.53	0.59	0.59	0.58	0.59	0.60	0.63	0.69	0.74
β^m_{conv}	0.90	0.91	0.94	0.97	0.95	0.94	0.45	0.88	0.87	0.77	0.78	0.68	0.79	0.49
β^m_{spread}	0.90	0.93	0.93	0.94	0.92	0.90	0.86	0.93	0.81	0.73	0.75	0.67	0.73	0.15
β^m_{tele}	0.49	0.47	0.43	0.39	0.01	0.15	0.45	0.55	0.60	0.65	0.75	0.85	0.93	0.94
β^m_{noin}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
score θ^m	3.84	3.87	3.85	3.84	3.36	3.52	3.35	3.94	3.86	3.74	3.88	3.83	4.14	3.32
rank	4	11	7	9	10	15	20	5	8	13	6	12	3	21

to stochastic factors. We expect that seed robustness likely improves with larger networks and higher travel demand shares. Additionally, in Section 7.3.4, we observe indications that random seed stability increases in simulations involving combined levers (scenarios B* in Table 7.1). Notably, simulations with relatively low relative mean speeds of approximately 0.35 to 0.45 generally exhibit the lowest output fluctuations across all seed runs, suggesting that congested traffic states tend to be more stable regarding seed-related impacts.

It is not surprising that lower relative mean speeds of a lever correspond to a higher number of vehicles requiring teleportation during the simulation. Interestingly, however, the number of vehicles that remain uninserted does not increase with shorter accepted `max-depart-delay` times. This seemingly counterintuitive finding suggests that by quickly alleviating local traffic hotspots, many severe gridlocks can be prevented from manifesting in the first place,

ultimately leading to fewer teleported and uninserted vehicles over time. A similar effect is observed for short teleport times of 60 seconds: critical vehicles are removed more rapidly without triggering secondary queuing effects. The rationale behind our penalization of teleported and uninserted vehicles is their potential impact on our ego-vehicles, which undermines our ability to microscopically track their trajectories. Thus, we should prioritize shorter `time-to-teleport` and `max-depart-delay` periods.

Additionally, it is noteworthy that the `remove-loops` scenario **C4** and the `rerouting-period` scenario **C2** have the least impact on the network's relative mean speed. Scenario **C4** benchmarks against the 300 s `time-to-teleport` lever from scenario **A1**. As evidenced in Table 7.2, eliminating u-turns at the start and end edges of a trajectory does improve network performance, albeit to a limited extent.

Another important point is the double penalization of all levers in the `max-depart-delay` scenario **A2**. This scenario is unique in that vehicles are expelled from the simulation if they cannot be inserted within the specified time frame, resulting in an additional penalty for all levers, on top of the standard penalty for teleported vehicles. As discussed in Section 7.3,however, this mechanism is essential for mitigating cloning artifacts arising from the transfer of travel demand from MATSim to SUMO.

7.3.4 Evaluating the cumulative impact of multiple measures on network performance

For the combined-measure scenarios denoted as B* (see Table 7.1), we employ the top-performing lever settings highlighted in gray in Table 7.2. In these simulations, we move beyond the fixed 50% travel demand, instead testing ten different travel demand levels, ranging from 10% to 100% in 10% increments. This methodology allows us to determine how many agents the SUMO network can handle efficiently under these *optimal* conditions.

Figure 7.7 presents the results of these simulations, each conducted with four different random seeds. The top panel illustrates the network's relative mean speed as demand increases. Each line represents the mean across seeds for one of the three scenarios: **B1** MATSim 1% with `VSP`-adjusted network, **B2**

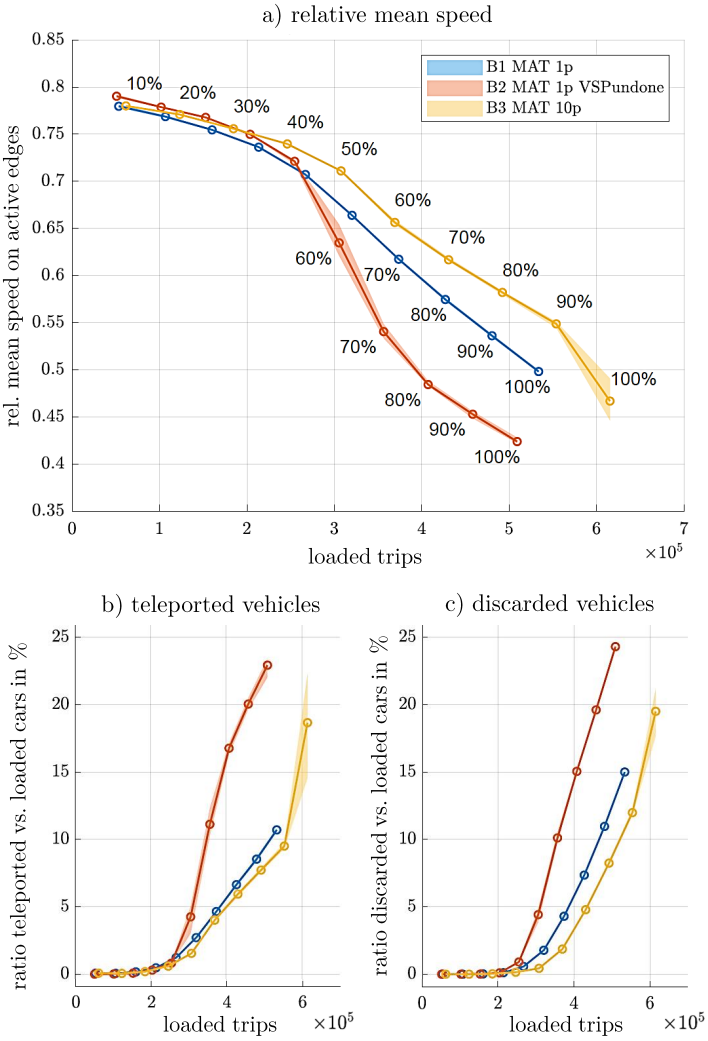

Figure 7.7: Comprehensive analysis of combined scenarios: Network relative mean speed (a), number of teleported vehicles (b), and number of not-inserted vehicles (c) as a function of loaded trips. Each data point corresponds to a specific demand share, as indicated. The solid lines represent the mean values across four random seeds, while the shaded regions illustrate the variability across seeds.

MATSim 1% without `VSPAdjustments`, and **B3** MATSim 10% with `VSP`-adjusted network (see Table 7.1). The shaded regions indicate the variability across seeds. The bottom left panel shows the ratio of teleported to loaded vehicles, while the bottom right panel depicts the ratio of discarded (not-inserted) to loaded vehicles.

While in Section 7.3.1 we struggled to accommodate a 50% travel demand in SUMO (see Figure 7.5), the optimized parameters have now significantly improved overall network capacity. The daily relative mean speed on all active edges has more than doubled across all three scenarios: previously, the relative mean speed for a 50% demand was approximately 0.35. Now, with the optimized settings, the relative mean speed ranges between 0.7 and 0.73 at the same demand level. Even at 100% demand, the relative mean speed remains between 0.42 and 0.5, indicating a far less congested network compared to the 50% demand simulations from Section 7.3.1.

As shown in the top panel of Figure 7.7, the results reveal that SUMO simulations based on a MATSim run without `VSPAdjustments` (scenario **B2**) consistently underperform compared to its `VSP`-adjusted counterpart (scenario **B1**). This discrepancy arises because, on average, the `VSP`-adjusted network capacities in urban areas are effectively higher, even though the adjustments were originally intended to introduce additional driving delays. As explained in Section 6.3.1 and illustrated in Figure 7.4, attempts to reduce capacities on primary and secondary roads through `VSPAdjustments` are counteracted – and even surpassed – by simultaneous capacity increases on short links (< 100 m).

Moreover, in addition to the lower relative mean speeds on active edges in scenario **B2**, the number of loaded trips[7] is slightly lower compared to the corresponding demand shares in scenario **B1**. The reduction in loaded trips observed for the `VSP`-undone scenario **B2** can be primarily attributed to secondary effects stemming from the reduced network capacity. In **B2**, the diminished attractiveness of the road network, caused by longer travel times and increased congestion, leads to fewer agents selecting road traffic as their mode choice in MATSim. This, in turn, reduces the total number of trips generated by MATSim (and subsequently in SUMO). MATSim's co-evolutionary algorithm,

[7] Loaded trips include both successfully inserted trips and those discarded due to congestion at entry links.

which iterates over numerous cycles (typically around 500 in this case), aims to generate a dynamic traffic assignment in a quasi-equilibrium state.

As expected, SUMO simulations using a travel demand generated from a 10% sample run of MATSim, represented by scenario **B3**, show a significantly improved network state compared to the baseline scenario **B1**. This improvement is due to the larger sample size in MATSim, which produces a wider variety of trip types and a more diverse distribution of start and end links. Additionally, during the MATSim-to-SUMO demand transfer, agent cloning is reduced by a factor of 10, significantly minimizing the number of agents trying to access the simulation via the same link within a short time window. However, we avoid scaling the travel demand to 100% in SUMO, as the relative mean speed drops sharply beyond a 90% demand share, and the number of teleported and not-inserted vehicles rises dramatically, as shown in the bottom panels of Figure 7.7.

For all subsequent simulations required for the calibration of meso-micro traffic states, we make two critical decisions. First, we select the 80% travel demand scenario based on the 10% MATSim simulation with `VSP`-adjusted network. Second, we discontinue varying the `remove-loops` setting (**C4**) and the `rerouting-period` value (**C2**), as these parameters have minimal impact on traffic conditions. Instead, we set them to 'on' and 60 seconds, respectively. By making these decisions, we reduce the remaining parameter space for our calibration procedure, which helps accelerate convergence and decrease computation time.

7.4 Genetic algorithm for calibrating global network states

7.4.1 Design of the framework

In the previous section, we gained valuable insights into the behavior of the SUMO network and its response to various parameter settings. Building on this foundation, we formulate a global optimization problem and employ a genetic algorithm (GA) to solve it numerically. Genetic algorithms are a subset of

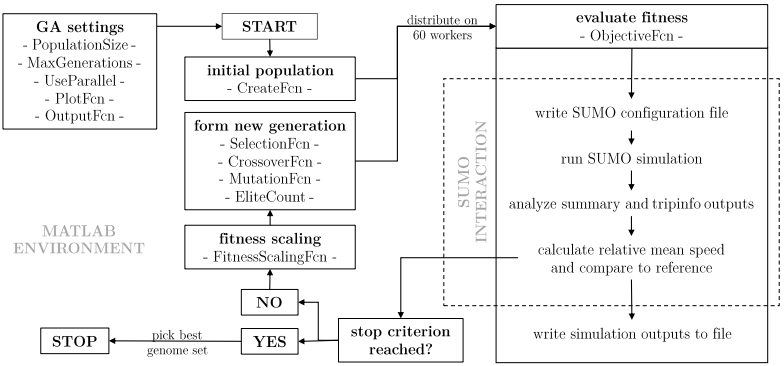

Figure 7.8: Framework design of the genetic algorithm.

evolutionary algorithms used for addressing both constrained and unconstrained optimization problems through mechanisms akin to natural selection.

As illustrated in Figure 7.8, we leverage the GA functionalities available in MATLAB[8] to facilitate the parallel execution of SUMO simulations while tackling the optimization problem. The process begins with an initial population of potential solutions, where each individual in the population has a distinct set of genomes corresponding to various SUMO parameters relevant to our use case. Detailed information about the parameter space is provided in the following section.

Next, we evaluate the fitness of each candidate solution using a fitness function. This function involves executing SUMO simulations with the designated parameter sets, analyzing the relative mean speed on all active edges throughout the day, and comparing these results to reference data. After this evaluation, a new generation of offspring is created by applying evolutionary principles such as mutation, crossover, and selection to the genomes of the parent individuals. This iterative process aims to evolve the genomes within the initial population toward more optimal solution candidates.

[8] For more information, please refer to "GA Documentation," MATLAB, https://de.mathworks.com/help/gads/ga.html?searchHighlight=ga&s_tid=srchtitle. Accessed: Sep. 25, 2024.

MATLAB provides a wide range of evolutionary strategies that can be customized to meet specific requirements through user-defined algorithms. For our purposes, we have chosen the following options for the genetic algorithm:[9]

```
options = optimoptions('ga',...
'PopulationSize', 60, ...
'MaxGenerations', 150, ...
'CreationFcn',@gacreationuniform,...
'SelectionFcn',@selectionstochunif,...
'CrossoverFcn',@crossoverscattered,...
'CrossoverFraction',0.8,...
'MutationFcn',@mutationgaussian,...
'EliteCount', '0.05*PopulationSize', ...
'FitnessScalingFcn', @fitscalingrank, ...
'MaxStallGenerations', 5,...
'FunctionTolerance', 1e-06, ...
'UseVectorized', false, ...
'UseParallel', true, ...);
```

To enhance the speed of the simulation, the fitness evaluation process is executed in parallel using 60 worker threads on a Lenovo P620 Workstation, equipped with an AMD Ryzen Threadripper PRO 3995WX processor. This parallelization enables the simultaneous execution of 60 SUMO simulations, allowing for the rapid assessment of the fitness of an entire generation.

7.4.2 Formulating the global optimization problem

The evaluation of each population's fitness relies on an objective function with the aim of minimizing the quarter-hourly differences in relative mean speed between MATSim and SUMO. This global optimization problem can be mathematically defined as follows:

$$\min \epsilon(x), \quad x = (x_1, x_2, x_3) \in \mathbb{R}^3. \tag{7.9}$$

[9] For further details, please refer to "GA Options," MATLAB, https://de.mathworks.com/help/gads/genetic-algorithm-options.html#f9147. Accessed Sep. 25, 2024.

Here, $\epsilon(x)$ represents the objective function, and x comprises the set of independent genomes subject to mutation during the iterative GA procedure to generate new offspring populations. Specifically, x_1 represents the time the foremost vehicle has to wait before being teleported to the next available edge on its path, x_2 denotes the maximum waiting time for vehicle insertion before being skipped, and x_3 describes the probability of a vehicle having a rerouting device in SUMO. To constrain the GA's solution space, the genomes are limited to the following discrete value sets:

$$x_1 \in \{30, 35, 40, ...300\} \tag{7.10}$$

$$x_2 \in \{30, 60, 90, ...1800\} \tag{7.11}$$

$$x_3 \in \{0.6, 0.65, 0.7, ...0.9\}. \tag{7.12}$$

The defined solution space operates without any additional linear, non-linear, or inequality constraints. For our specific use case, we formulate the objective function $\epsilon(x)$ as the normalized sum of square-penalized errors, estimating the differences between observed values $y(x)$ in MATSim and the simulated values $\tilde{y}(x)$ in SUMO throughout a day. This objective function is expressed as:

$$\epsilon(x) = \frac{\sum_{t=1}^{T} \left[y(x,t) - \tilde{y}(x,t) \right]^2}{T}. \tag{7.13}$$

Here, T corresponds to the total number of 96 binning intervals t, each with a duration of $\mathrm{d}t = 15$ min. Within each binning interval, the relative mean speed on all active edges e is utilized as the evaluation criterion:

$$y(x,t) = \sum_{e=1}^{E(t)} \frac{v_{\mathrm{act},x,e}}{v_{\mathrm{lim},x,e}}. \tag{7.14}$$

In this context, $E(t)$ signifies the time-dependent total number of active edges that vehicles traverse during the time interval t. The ratio $\frac{v_{\mathrm{act}}}{v_{\mathrm{lim}}}$ represents the actual time- and space-mean velocity v_{act} of each active edge in relation to its corresponding speed limit v_{lim}. Notably, it is essential to recognize that due to MATSim's `VSPAdjustments` (refer to Section 7.3), the speed limits in MATSim networks are scaled down by a factor of two for links with legal

speed limits less than 51 km/h. Consequently, when evaluating the fitness of populations, these adjustments must be reversed to ensure that the velocity ratio $\frac{v_{\text{act}}}{v_{\text{lim}}}$ aligns with SUMO simulations.

7.4.3 Discussion of the best-fit solution

The GA procedure effectively terminated at generation 12 when the average relative change of the best score remained below 10^{-6} (`FunctionTolerance`) for five consecutive generations (`MaxStallGenerations`). In total, 720 individual populations were simulated, with an average SUMO simulation duration of approximately 2.5 hours. As the best solution to our global optimization problem, the following parameter set was identified: $x_{\text{opt}} = (65\text{s}, 1500\text{s}, 0.7)$, representing the `time-to-teleport`, `max-depart-delay`, and `rerouting-probability`, respectively.

A graphical representation of the best-fit solution is presented in Figure 7.9. The left panel offers an overview of the various error values, $\epsilon(x)$, across all 720

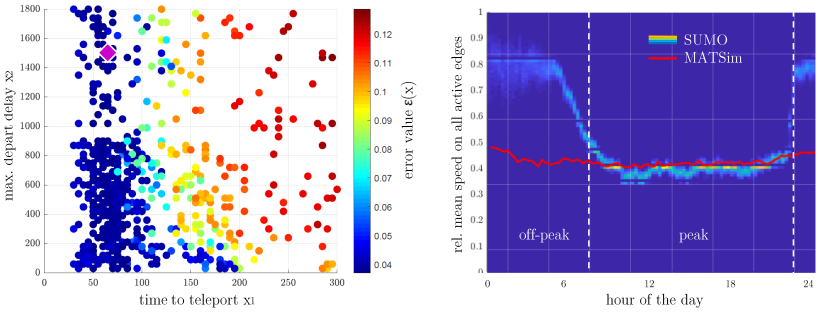

Figure 7.9: Graphical representation of the best-fit solution obtained through our Genetic Algorithm (GA). The left panel depicts the variations in error values $\epsilon(x)$ for 720 distinct populations across the parameters x_1 and x_2 as an example. The magenta diamond highlights the best-fit solution, indicating that the `time-to-teleport` parameter significantly influences the outcomes, except for simulation runs with small `max-depart-delay`. In the right panel, the temporal evolution of the relative mean speed on all active edges for SUMO's best-fit solution is compared to the MATSim reference (red line). The color code represents the frequency distribution of specific combinations of daytime and relative mean speed, transitioning from blue to yellow.

populations concerning parameters x_1 and x_2. The best-fit solution is marked with a magenta diamond. Generally, the `time-to-teleport` parameter, x_1, has the most direct impact on population scores, followed closely by the `rerouting-probability`, x_3. The negative impacts associated with high values of the `max-depart-delay` parameter,x_2, such as an increased number of teleported and not-inserted vehicles, can be mitigated by implementing lower `time-to-teleport` values. As a result, the `max-depart-delay` parameter does not play a critical role in the overall performance of the system.

The right panel of Figure 7.9 illustrates the relative mean speed of SUMO's best-fit solution over the course of a full day. The reddish MATSim line serves as a reference. Two noteworthy observations arise: (i) SUMO reasonably matches MATSim throughout the day, especially during peak hours, but it yields relative mean speeds that are too high during the night-time (off-peak). (ii) Intuitively, one could argue that the MATSim benchmark line during nighttime periods does not align with real-world observations. Due to the `VSPAdjustments`, which halve all speed limits when the legal speed limit is below 51 km/h, the MATSim relative mean speed on all active edges is constrained to a maximum of 0.5 in this context, a constraint that may seem overly pessimistic.

Interestingly, MATSim's travel times for individual vehicles closely align with those observed in Google Maps, regardless of the trip's starting time (cf. Figure 7.14). This similarity suggests a good agreement in relative mean speeds as well. Consequently, we can are inclined to infer that SUMO tends to underestimate travel times during nighttime, resulting in relatively high mean speeds. One potential reason for this discrepancy could be the actuated traffic light control, which results in an excessive amount of green lights during nighttime simulations. In real-world scenarios, traffic lights are often switched off during these hours for economic reasons, causing vehicles to approach intersections more cautiously.

Based on these findings, we anticipate that our SUMO ego-vehicles will exhibit travel times similar to those of corresponding MATSim reference vehicles during daytime hours. However, during off-peak periods, we expect considerably shorter travel times. To validate this hypothesis, we analyze the trip travel times of our selected ego-vehicle, as introduced in Figure 7.2, across all populations. For the purpose of distinguishing off-peak and peak-time trips, we use the first trip at 6:00 am and the third trip at 18:38 pm as reference points.

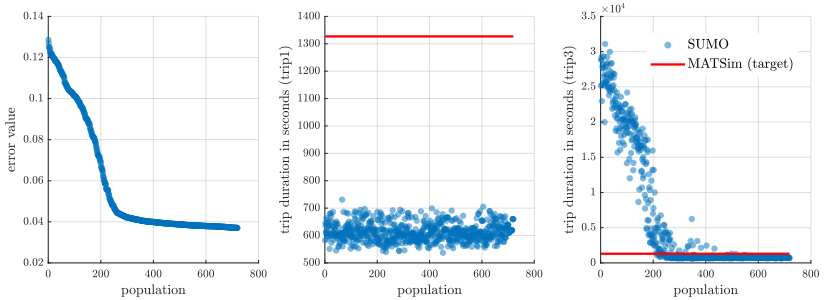

Figure 7.10: Convergence of error values with higher GA populations and their impact on trip travel time in free-flow and congested traffic conditions. Left panel: Error values for all GA populations sorted in descending order. Middle and right panels: Duration of the ego-vehicles' trip 1 (middle) and trip 3 (right) for all populations, sorted in the same order as in the left panel. Notably, for trip 3, the travel times exhibit a comparable trend to the error value $\epsilon(x)$ and converge towards the MATSim travel time.

Initially, we arrange and plot the error values of all populations in descending order, with the rightmost point representing the best-fit solution (as illustrated in the left panel of Figure 7.10). We subsequently apply the same sorting to the duration of trip 1 and trip 3 for all populations as displayed in the middle and right panel respectively. As expected, travel times for trip 1 demonstrate minimal sensitivity to variations among different populations. This is attributed to the network's consistent free-flow driving conditions during off-peak hours. Moreover, the travel time observed in MATSim is approximately twice as long as that simulated in SUMO. Conversely, peak-time trip 3 reveals a different pattern. Here, travel times across different populations converge in a manner similar to the error value $\epsilon(x)$ and stabilize in close proximity to the MATSim travel time. The remaining variations in travel times can primarily be attributed to two factors: (i) the ego-vehicle may encounter slightly different route lengths in the respective MATSim and SUMO simulations, and (ii) variations in individual `SpeedFactor`s can lead to notable differences in driving style, causing the driver to either frequently exceed speed limits or drive more slowly than average.

Given the promising results of the global optimization during peak hours, we remain optimistic that any remaining minor spatial traffic inconsistencies during these periods can be resolved through subsequent local optimization efforts outlined in Section 7.6. This is especially relevant when ego-vehicles are

restricted from further re-routing. However, during off-peak hours, the global optimization reveals potential for enhancement. The next section will focus on addressing these challenges.

7.5 Navigating off-peak hour challenges: strategies and solutions put in perspective

To address the optimistic travel times observed in SUMO during night hours, we introduce additional delays for ego-vehicles by modifying traffic signal control. Presently, gap-based actuated traffic lights are employed throughout the day, adjusting dynamically to prevailing traffic conditions. To prevent a continuous green-wave effect during the night, we deactivate these traffic lights from 22:30 to 7:00, prompting vehicles to approach intersections with increased caution. When traffic lights are deactivated, intersections operate under a priority model, where vehicles on major roads have right-of-way, and those on minor roads must yield. To customize traffic light programs for individual intersections in SUMO, we utilize the *Wochenschaltautomatik* (WAUT) descriptions, which enable the loading of different signal programs for the same traffic light based on defined time intervals[10]. This adjustment aims to introduce additional delays to SUMO vehicles during night hours, ensuring better alignment with MATSim's traffic states.

The results, illustrated in Figure 7.11, are somewhat surprising. Contrary to our hypothesis, turning off traffic lights (red line) leads to even shorter travel times compared to those under actuated traffic light control (blue graph). This challenges the expectation that actuated traffic lights establish continuous green waves for sparsely distributed ego-vehicles at night. Instead, it appears that temporal cool-downs[11] of the actuated control induced some vehicles to wait at red lights. Conversely, when traffic lights were switched off, major roads rarely

[10] For further details on the weekly switch automatism, refer to "Traffic Light Documentation," SUMO, https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html. Accessed Sep. 26, 2024.

[11] Unlike default traffic lights with fixed phase layouts, actuated traffic light control features green phases of varying lengths (typically 5-50 seconds) that dynamically adapt to traffic demand. However, once a direction is granted a green phase, a configurable cooldown period may be enforced before it can receive another green light.

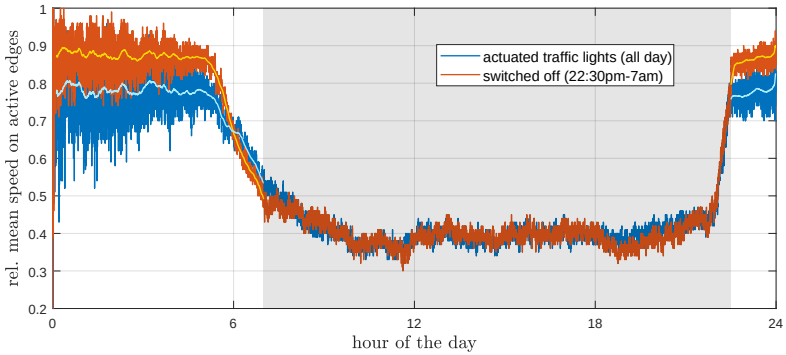

Figure 7.11: Adapted traffic light control at night to address off-peak hour challenges. Surprisingly, deactivating traffic lights from 22:30 pm to 7 am (highlighted in red) did not result in additional delays during nighttime hours compared to actuated traffic light control (depicted in blue). Contrarily, the relative mean speed on all active edges in SUMO even increased when traffic lights were disabled.

experienced additional delays due to constant priority granting. In consideration of these outcomes, our preference leans toward retaining the implementation of actuated traffic light control.

Our next attempt involves the localized induction of additional delays to ego-vehicles during nighttime by introducing supplementary traffic. However, we took a moment to reconsider the viability of this approach. MATSim, despite being calibrated on travel times, exhibits two noteworthy characteristics: (i) it sustains stable driving conditions on the free-flow regime, and (ii) its relative mean speeds on active edges consistently hover between $0.4 - 0.5$ irrespective of the time of day. Both aspects stand in contrast to naive expectations. Thus, instead of compelling SUMO to exhibit unrealistic traffic behavior (adapted from MATSim), we chose to seek validation from an external source, namely Google Navigation, to obtain realistic travel times in Berlin during nighttime. This methodology is employed to assess the degree of deviation in travel times simulated by SUMO from reality.

The upper panel of Figure 7.14 demonstrates that randomly selected MATSim trips (indicated by blue dots) align closely with travel times estimated by Google Navigation (represented by reddish rectangles). During the daytime, MATSim

travel times fall within the midrange of Google estimates, whereas during nighttime, they tend to approach the upper bound. In contrast, SUMO clones (red dots) consistently remain in the lower bound of Google estimates under non-congested conditions with our current parameterization. In cases of severe congestion, SUMO exhibits significantly higher travel times, as evidenced by the outliers for cars 13, 16, 17, and 18, which lie outside the bounds of the figure due to their extreme values. However, it is safe to conclude that SUMO travel times generally remain within an acceptable range of realism during nighttime conditions, which raises the question of whether any adjustments to SUMO are necessary for this time period.

Additionally, we would like to convey the following considerations: Google Navigation derives travel time estimates from real-time data, taking into account factors such as current traffic conditions. While these estimates strive for realism, they are subject to changes caused by unforeseen events, such as accidents or sudden increases in traffic. It is acknowledged that Google may incorporate a margin of error in their estimates, potentially overestimating driving times, especially for longer distances. This is attributed to the strategy of erring on the side of caution, as it is generally considered favorable to arrive early rather than late. Such reasoning reinforces SUMO's inclination toward optimistic travel times, prompting us to maintain them as simulated. The immediate consequence of the discrepancy between MATSim and SUMO traffic states during nighttime is that SUMO DRT vehicles are unlikely to face the risk of tardiness for their designated customers. This outcome is deemed acceptable.

7.6 About the futility of local calibration

After completing our global calibration, we turn our attention to the local calibration procedure. The goal is to refine the spatio-temporal alignment of individual DRT trajectories in both MATSim and SUMO. Despite previous network-wide calibration, individual ego-vehicles still encounter varying delays between both frameworks on their routes. On a more local scale, the presence or absence of vehicles in specific areas leads to fluctuations in traffic conditions, influencing ego-vehicle travel times in the short term. While these localized variations tend to average out over longer trips due to the global calibration,

they present challenges for shorter DRT trips. When individual travel times do not align well between MATSim and SUMO, DRT vehicles may lose their idling times between consecutive trips, disrupting the entire DRT dispatching process. Consequently, DRT vehicles could arrive late for their next designated customer or, in the case of EVs, miss out on charging opportunities.

Unfortunately, our attempt to locally calibrate individual DRT trajectories proved unsuccessful and reached a dead-end due to various reasons. However, acknowledging the importance of sharing unsuccessful research approaches, we present our attempted local calibration method nonetheless.

Our methodology was rooted in the following principle: Leveraging the globally calibrated MATSim and SUMO simulations, we overlay MATSim's spatio-temporal traffic conditions – characterized by the links' relative mean speeds distributed across space and time – onto the conditions generated by SUMO. This process generates a map that pinpoints areas in the SUMO network where additional vehicles are needed to introduce delays or, conversely, where ambient traffic should be reduced to alleviate congestion. This approach requires a subsequent SUMO simulation run with disabled re-routing functionalities to fix vehicle routes to SUMO's globally calibrated state.

We begin by calculating and plotting the relative mean speed for each link in MATSim at 15-minute intervals, as illustrated in the top panel of Figure 7.12 for 4pm. In this figure, greenish colors indicate relative mean speeds around 0.5[12], while transitions to yellow and red indicate worsening congestion levels, with red representing near-complete traffic breakdown. By observing these temporal snapshots in sequence, we can track the emergence, propagation, and dissipation of vehicle queues over time.

Next, we translate these spatially distributed relative mean speeds in into auxiliary surfaces[13] – one for each 15-minute interval and major cardinal direction (N, E, S, W) – and apply a smoothing technique to redistribute

[12] It is noteworthy that a relative mean speed of 1 in MATSim corresponds to approximately 0.5 in SUMO due to the halving of speed limits in urban areas specifically for MATSim `VSP`-adjusted runs (limited to links with speed limits below 51 km/h). This normalization ensures a uniform basis for comparison.

[13] To increase the sampling density of the auxiliary surfaces, each link is represented by a sequence of ten equidistant points indicating the average relative mean speed during the respective time interval.

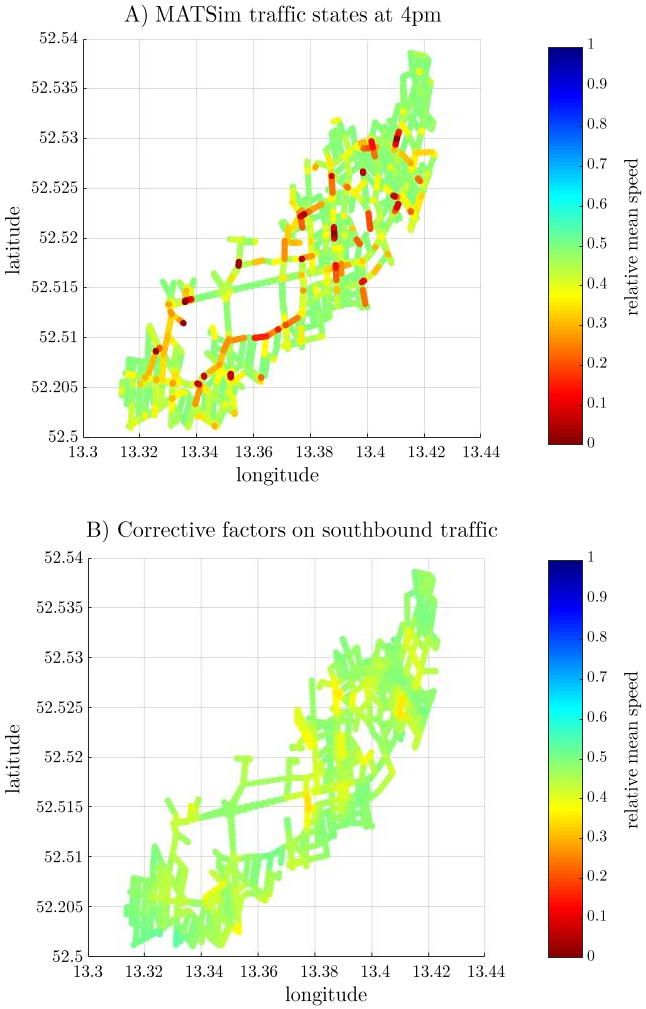

Figure 7.12: Illustrations of the local calibration approach. The *top panel* shows the distribution of relative mean speeds on a link-by-link basis within MATSim simulation at 4 pm. Each link is represented by approximately 10 evenly spaced scatter points. Areas with congestion (highlighted in reddish tones) are emphasized (brought to the foreground) for clarity. In contrast, the *bottom panel* identifies regions in SUMO simulation where southbound traffic must be intensified to introduce additional delays for our ego-vehicles. Highly congested areas (dark reddish, short links) in MATSim are spatially and temporally smoothed to create more moderate traffic congestion.

localized delays to adjacent roads with similar orientations. For instance, when evaluating traffic conditions for southbound roads at a specific time of day, we specifically consider the relative mean speeds for road segments with orientation angles falling within the range of $135° < \theta \leq 225°$. This direction-wise mapping is essential for capturing the directional nature of congestion patterns, such as city-bound congestion during morning rush hours and outbound traffic jams in the afternoon.

Overlaying these auxiliary surfaces from MATSim with their corresponding SUMO counterparts enables us to identify specific areas within the microscopic network that require traffic flow adjustments, such as the insertion or removal of vehicles, as illustrated in the bottom panel of Figure 7.12. Correction factors derived from the combined data are then applied to adjust SUMO's ambient travel demand on the affected edges, effectively guiding the adjustment process.[14]

As previously noted, the effectiveness of this approach is compromised for several reasons: (i) MATSim struggles to accurately replicate spatio-temporal congestion patterns, with bottlenecks predominantly forming on short links rather than at infrastructural choke points. (ii) Agents in MATSim demonstrate a notable ability to bypass obstacles, leading to only minor congestion levels. (iii) Prolonged vehicle queues caused by infrastructural bottlenecks did not develop over time; instead, static or erratic local congestion hotspots emerged. Consequently, imposing such arbitrary congestion patterns on SUMO is unwarranted, given that SUMO typically represents spatio-temporal congestion dynamics more realistically than MATSim.

Unfortunately, this unsuccessful attempt to calibrate trip-based travel times requires us to reevaluate our global calibration procedure. Originally, the global calibration was intended as a foundational step to establish an optimal starting point for the more detailed local calibration process. However, as the local calibration is now deemed unfeasible, we must redefine the objective of our global calibration procedure. Instead of focusing on aligning network traffic states in MATSim and SUMO, we will shift our attention to aligning ego-vehicle travel times. Unfortunately, we question the feasibility of achieving this trip-based alignment through static adjustments of global parameters, which

[14] Please note that this concept has not been implemented or realized.

was a key rationale for adopting a two-staged calibration approach in the first place. Nevertheless, with this shift in focus, our next steps will involve three primary actions. First, we will analyze travel times in both MATSim and SUMO on a trip-based level. Second, we will explore the correlation between the previous target value of the genetic algorithm – which represented relative mean speeds on active edges – and the revised target value reflecting trip-based travel times. Finally, we will reevaluate all SUMO simulations produced by the Genetic Algorithm in Section 7.4, considering our updated target value.

7.7 Alternative GA objective function: trip travel time comparison

To assess the consistency of trip-based travel times between MATSim and SUMO, we assign each SUMO clone its corresponding MATSim vehicle and compare their travel times using a log-log representation, as shown in Figure 7.13. Our analysis focuses on trips occurring between 9 a.m. and 9 p.m. to exclude uncalibrated off-peak hours.

Even after global calibration, significant differences remain between SUMO and MATSim travel times. On average, SUMO vehicles complete trips approximately 41% faster than their MATSim counterparts. However, this mean is skewed by upper-left outliers, with some SUMO cars experiencing travel time delays of up to 1800% compared to MATSim. These extreme values arise mainly from vehicles without rerouting capabilities becoming stuck in infrastructural bottlenecks.[15] Sample percentiles ($p = \{0.05, 0.25, 0.50, 0.75, 0.95\}$) provide more meaningful metrics: $F(p) = \{252.5\%, 42.5\%, -12.5\%, -37.5\%, -57.5\%\}$, highlighting the substantial variability in relative travel time deviations. Positive percentages indicate that SUMO vehicles are delayed, whereas negative percentages signify that SUMO vehicles are ahead of schedule relative to their MATSim counterparts.

[15] We observe that SUMO vehicles without rerouting often fail to reach destinations on time due to congestion. Revising our initial design choice, where DRT ego-vehicles preserved their original MATSim routes, we now recommend enabling rerouting for these vehicles. After all, our selection of DRT vehicles is based on aggregated KPIs rather than individual route choices.

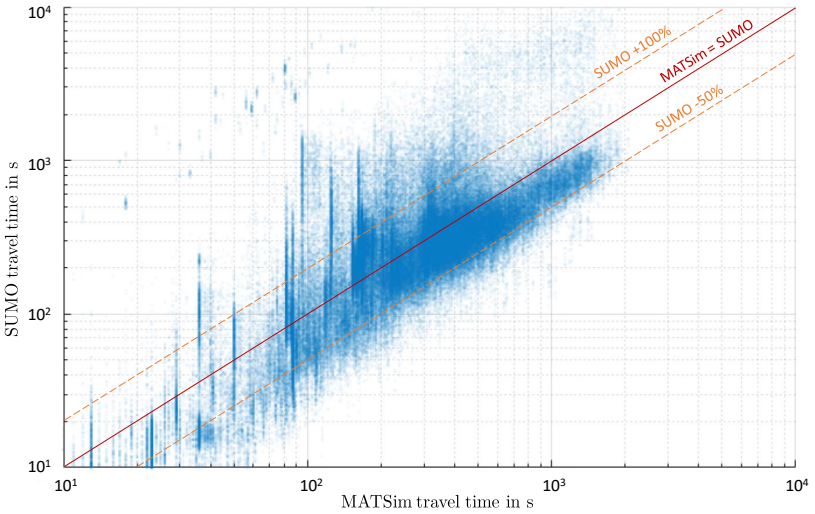

Figure 7.13: Trip-based log-log comparison of travel times (9 am to 9 pm): Heavily influenced by upper-left outliers, on average, SUMO vehicles experience a 41% slower travel time than MATSim counterparts. Sample percentiles ($p = \{0.05, 0.25, 0.50, 0.75, 0.95\}$) reveal relative travel time deviations of $F(p) = \{252.5\%, 42.5\%, -12.5\%, -37.5\%, -57.5\%\}$. Positive percentages indicate delays for SUMO vehicles, while negative percentages signify they are ahead compared to MATSim. In conformity with the median value, denser sample agglomerations appear below the red equal-value line. Vertical concentrations indicate consistent MATSim trips, while their corresponding clones in SUMO encounter varying traffic conditions.

Contrary to expectations, denser sample clusters do not align closely with the red reference line (representing equal travel times in MATSim and SUMO) but appear below it, indicating that most SUMO vehicles tend to be faster. Consistent with this observation, the median value is approximately -12.5%, using MATSim as the baseline for comparison.

In light of these suboptimal findings, we want to emphasize that our goal is to align DRT travel times, not the trips undertaken by ambient traffic. We will focus on a more ego-vehicle-centered analysis later in this section. It is also important to note that the medium size of our test case, coupled with its limited geographic latitude spread, leads to an overrepresentation of short trips that briefly traverse our network. As a result, these short trips tend to exhibit large

relative deviations, despite the absolute differences being comparatively small and statistically insignificant.

Furthermore, short trips are influenced by another systematic error: In MATSim, vehicles face delays on each link determined by the link's outflow capacity. Conversely, SUMO vehicles tend to accelerate near the network border as they encounter fewer traffic-induced impediments. This behavior is reflected in the lower left side of Figure 7.13, where SUMO vehicles often display faster travel times. These effects – such as the prevalence of short trips and reduced network resistance near SUMO border regions – are expected to diminish in larger scenarios. During toolchain development, however, a smaller test case was necessary to manage simulation times effectively.

This leads us to question why the majority of SUMO vehicles consistently outperform their corresponding MATSim counterparts, despite the implementation of global calibration. We can attribute the observed temporal offset to several factors. Firstly, the target value selected for our initial GA run, which emphasizes the relative mean speed on active edges, may not effectively align travel times at the individual vehicle level. While it shows a tendency to synchronize travel times with improved scores, the correlation strength between these quantities might be inadequate.

Secondly, we realize that the metric chosen by the GA for accumulating error deviations within our objective function may have been suboptimal in retrospect. By minimizing least square errors (see Eq. 7.13), particularly pronounced during off-peak hours, the algorithm might have focused on alleviating these larger discrepancies, potentially at the expense of optimizing smaller adjustments, such as travel time offsets during peak hours.

Thirdly, after careful consideration, we question the direct comparability between the relative mean speed on active edges in MATSim and SUMO simulations. We generally assume that MATSim has fewer active edges due to its predominantly non-saturated traffic conditions, which provide ample capacity for additional vehicles. Consequently, unlike our rerouting strategy in SUMO, ego-vehicles in MATSim do not need to seek bypasses or explore less-frequented roads in the same direction. Additionally, with only one-eighth of the vehicle population compared to SUMO, MATSim occupies fewer – and thus less diverse – street segments. Furthermore, each active street segment in MATSim and SUMO contributes equally to the final objective function,

treating less-frequented minor street segments with nearly free-flow speeds the same as heavily trafficked multi-lane highways. This approach diminishes the impact of congested, high-capacity arterial roads in SUMO.

In retrospect, relying on relative mean speed as a target value for comparison assumed that the macroscopic traffic dynamics in both frameworks are similar. However, the findings in Chapter 6, which unfortunately unfolded in parallel with the results of this chapter, challenge this assumption and reveal the inherent flaw in our approach. What initially seemed like an appropriate target value for global calibration now proves unsuitable for aligning travel times.

To address these limitations, we propose an alternative approach. We reassess all SUMO simulations derived from our initial GA run based on our updated target value, which now reflects ego-vehicle travel time. Since we did not conduct a complete DRT fleet simulation within the test case, we specifically focus on the travel times of 20 randomly selected ego-vehicles whose trip lengths exceed 7 km. In a complete DRT fleet simulation, we would utilize the travel times of all fleet representatives, as outlined in Section 5.1. Additionally, we change our approach from minimizing the least-square error to a linear summation of error deviations. This adjustment aims to provide the GA with a better incentive to align not only the significant travel time discrepancies during the night but also the smaller, yet more critical, deviations during the day.

Surprisingly, the relationship between the previous and newly updated target values remains remarkably strong across all 720 SUMO simulation trials. To evaluate the correlation between both scoring metrics, we calculated the Pearson correlation coefficient (R), which ranges from -1 to 1. A value of $R = 1$ indicates a perfect positive linear relationship, $R = -1$ signifies a perfect negative linear relationship, and $R = 0$ denotes no linear relationship between the variables. In our case, we observed a high positive correlation ($R = 0.95$), suggesting that an increase in the old score likely corresponds to an increase in the new score. Notably, we identified a new global optimum[16] at $x_{\text{opt, new}} = (40\,\text{s}, 720\,\text{s}, 0.9)$, representing `time-to-teleport`, `departure-delay`, and `rerouting-probability`, respectively. This discovery significantly improved the fit in ego-vehicle travel times, as illustrated in Figure 7.14.

[16] To recap, the global optimum identified in Section 7.4.3 was $x_{\text{opt, old}} = (65\,\text{s}, 1500\,\text{s}, 0.7)$.

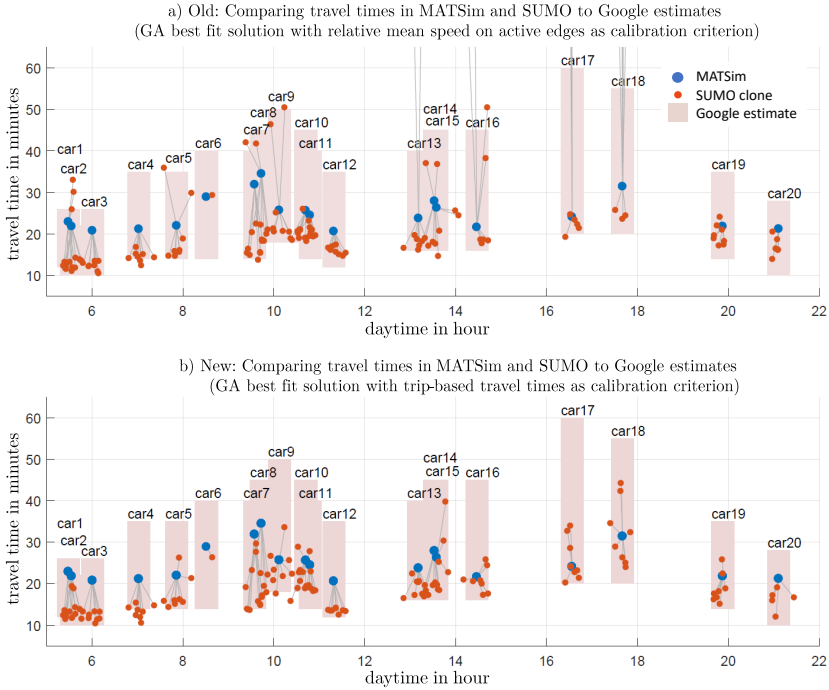

Figure 7.14: Comparison of travel times for selected ego-vehicles in MATSim and SUMO, with the top panel focusing on relative mean speed and the bottom panel utilizing trip-based travel times as calibration criterion. Blue dots represent MATSim, and red dots indicate corresponding SUMO clones with adjusted departure times. Daytime-dependent travel time estimates from Google Navigation are included for reference as shaded region. Note: Car no. 6 is an actual DRT vehicle in MATSim, not fully covered in our test case boundary. Although DRT vehicles were not prominent in calibration, we intentionally avoided cloning them in SUMO to prevent potential discarding due to crowded starting edges, resulting in only one SUMO counterpart instead of eight.

In this figure, both panels compare ego-vehicle travel times in relation to the respective GA best-fit solution. The top panel emphasizes the relative mean speed on active edges as the calibration criterion, while the bottom panel utilizes trip-based ego-vehicle travel times. Blue dots represent MATSim travel times, and red dots indicate the travel times of corresponding SUMO clones, which have slightly shifted departure times. To provide an unbiased reference, we

included daytime-dependent travel time estimates from Google Navigation[17], with their lower and upper bounds highlighted by the reddish area.

With the GA's new target value (bottom panel), most SUMO clones generally exhibit increased speeds. However, they do not risk becoming excessively late, as indicated by the outlier points in the top panel that required clipping for better readability. A tipping point appears: in SUMO runs with increased median travel times, the red SUMO dots approach the blue MATSim benchmark from below. Once they reach a certain threshold, however, they tend to flip strongly to the opposite side, surpassing MATSim values and resulting in SUMO vehicles becoming significantly slower than both MATSim and Google estimates. This scenario incurs a considerable penalty from the GA's objective function. Overall, with the new GA evaluation criterion, SUMO stays within a reasonable travel time range provided by Google Navigation, yielding particularly satisfying fitting results during late afternoon and evening hours. As discussed earlier in Section 7.5, SUMO vehicles tend to be too fast during off-peak hours, a behavior that is tolerated within the constraints of our toolchain.

With our calibration procedure generally yielding satisfactory results, we have now identified all relevant parameters necessary to automatically set up the SUMO configuration file (`.sumocfg`). This completes the final module in Figure 5.9, which previously prevented the execution of a SUMO simulation based on the network area and the population data provided by MATSim. A simplified example of a SUMO configuration file, along with detailed explanations on how to initiate a SUMO simulation, is provided in Appendix A.2. For completeness, we also included relevant SUMO command line options in Appendix A.1.

However, having reached this milestone, we still face challenges related to specific SUMO peculiarities, especially in large-scale DRT fleet simulations. These challenges pertain specifically to the management of tardy, teleported, or unintentionally discarded DRT ego-vehicles. Each scenario significantly interferes with our overarching goal of deducing realistic second-by-second driving, speed, and status profiles. Solutions to address these issues are discussed in the following section.

[17] Google Maps, https://www.google.com/maps/. Accessed: Sep. 29, 2024.

7.8 Management strategies for tardy, teleported, and discarded DRT vehicles

Despite our trip-based travel time calibration routine, we cannot guarantee that the SUMO vehicles of interest will avoid teleportation during their routes or, more concerning, that they will not fail to enter the microscopic simulation altogether due to fully occupied departure edges. As illustrated in Figure 7.7, approximately 7.5% of the vehicles in our test case experience teleportation at least once. Simultaneously, an almost equal proportion of vehicles are discarded, with a 80% population sample in SUMO deduced from a MATSim 10% sample run. This trend worsens with higher microscopic population sizes and deteriorating traffic conditions, leading to an increased occurrence of teleported and discarded vehicles.

Currently, teleportation behavior can only be configured globally, with no provision for excluding specific ego-vehicles from this functionality.[18] Given the indispensable role of teleportation in maintaining minimal traffic flow even under heavily congested conditions, disabling teleportation altogether is not a viable option. A similar issue arises with discarded vehicles; when vehicles fail to enter the simulation, they are permanently removed after a prolonged period of reattempted insertion. While extending the `departure delay` can help reduce the number of discarded vehicles, excessively long delays are counterproductive to our objective of deducing accurate driving cycles. A vehicle inserted 30 minutes behind schedule faces significant challenges in catching up, especially on busy days. In light of these considerations, we must acknowledge and navigate these inconveniences as inherent aspects of our simulation framework.

Handling of teleported vehicles. During the teleportation process, a vehicle temporarily exits the network and then travels its route at the average speed of the edge from which it was initially removed or the edge it is currently passing through. This treatment results in a deduced velocity profile characterized by a period of constant speed. To address this issue, we propose a post-processing

[18] Refer to https://www.eclipse.org/lists/sumo-user/msg10321.html. Accessed: Sep. 29, 2024.

routine that systematically examines all simulated velocity profiles for such unrealistic driving patterns, similar to the data cleansing procedure employed in our real-world measurement campaign (refer to Section 3.1.1). We replace the identified segments with cubic spline interpolations derived from neighboring data points. While more advanced methodologies exist for rectifying erroneous data, they fall outside the scope of this dissertation and are considered topics for future research. We flag interpolated driving cycles appropriately and compute the temporal ratio of altered to total driving cycles to highlight potential anomalies for drive cycle applicants.

Handling of discarded vehicles. We generally recommend simulating not just a single DRT vehicle in SUMO microscopically, but rather a representative sub-sample of the DRT fleet, as detailed in Section 5.1. This approach ensures that if the vehicle of interest is discarded, you can quickly switch to another DRT vehicle with similar KPIs. If you simulate only one DRT vehicle and it gets discarded, we suggest initiating another SUMO run with a different random seed. This randomization may help prevent the ego-vehicle from being discarded in the first place.

Handling of tardy vehicles. While DRT vehicles ahead of schedule pose minimal threats to our ability to deduce reasonable drive cycles – merely resulting in prolonged waiting times – those significantly delayed create situations where a DRT vehicle's next trip begins before the current trip has concluded. Although we could effectively manage these delays using SUMO's taxi extension[19] by dynamically adjusting the next trip's departure time, we have chosen a simpler approach.

Since two vehicles with the same identifier (ID) cannot coexist simultaneously within the simulation, we update the vehicle ID with a suffix indicating the current trip number each time a new trip starts. As a result, if trip one is still ongoing and trip two begins, a technically new vehicle is inserted into the simulation, independent of its "other" self still en route. In a post-

[19] The taxi extension has been available since version 1.5.0 (released in 2020), but it remains under active development: https://sumo.dlr.de/docs/Simulation/Taxi.html. Accessed: Sep. 29, 2024.

processing routine, we concatenate all related trip driving cycles, inserting the designated waiting times from MATSim's original velocity profile in between. However, this method may lead to instances where a vehicle's last trip concludes significantly later than intended.

To address this, we consider a second procedure where vehicles sacrifice portions of their longer waiting times to catch up with the schedule. Unfortunately, this may pose challenges in scenarios where EV charging management is critical. However, we assert that this is not a significant issue given the relatively low daily DRT travel mileage, as indicated in Table 2.5. With an average of 92 km driven in a DRT pooling scenario and 131 km in a non-pooled case, vehicles are unlikely to experience low states of charge, thus diminishing the necessity for daytime recharging. Nevertheless, considering the ecological and economic implications tied to the type of electricity used, it may be advantageous to charge SAEVs during periods of high solar radiation and, consequently, lower electricity prices.

7.9 Final toolchain and major components

Figure 7.15 illustrates the complete toolchain developed from Chapter 4 to Chapter 7, comprising seven principal components:

1. Mesoscopic MATSim (DRT) simulation
2. SUMO network generation featuring customizable network borders
3. Microscopic travel demand synthesis
4. Travel time calibration via genetic algorithm
5. Microscopic SUMO simulation
6. Post-processing of vehicle profiles
7. Enrichment procedure (as an alternative solution to SUMO simulation)

Throughout this thesis, we explored two distinct solution paths to derive realistic driving profiles for SAEV fleets. The first path involves our enrichment procedure, detailed in Section 3.1. The second solution path employs our sequential tool-coupling approach, which integrates MATSim as the mesoscopic transport planning framework and SUMO as the microscopic traffic simulation counterpart. Each approach presents unique advantages and limitations, and we propose their utilization according to the guidelines in Figure 9.2.

We primarily developed the toolchain in MATLAB, occasionally utilizing Python or GIT BASH to handle large datasets. These auxiliary tools, along with the SUMO command line interface, were orchestrated through MATLAB. Importantly, we did not integrate the Java-based MATSim framework into our MATLAB environment. Instead, our toolchain begins with existing MATSim output files. Using any travel-time calibrated MATSim scenarios as a basis, we can execute customized DRT runs with the assistance of the MATSim GUI or any preferred Java IDE, such as Eclipse or IntelliJ.

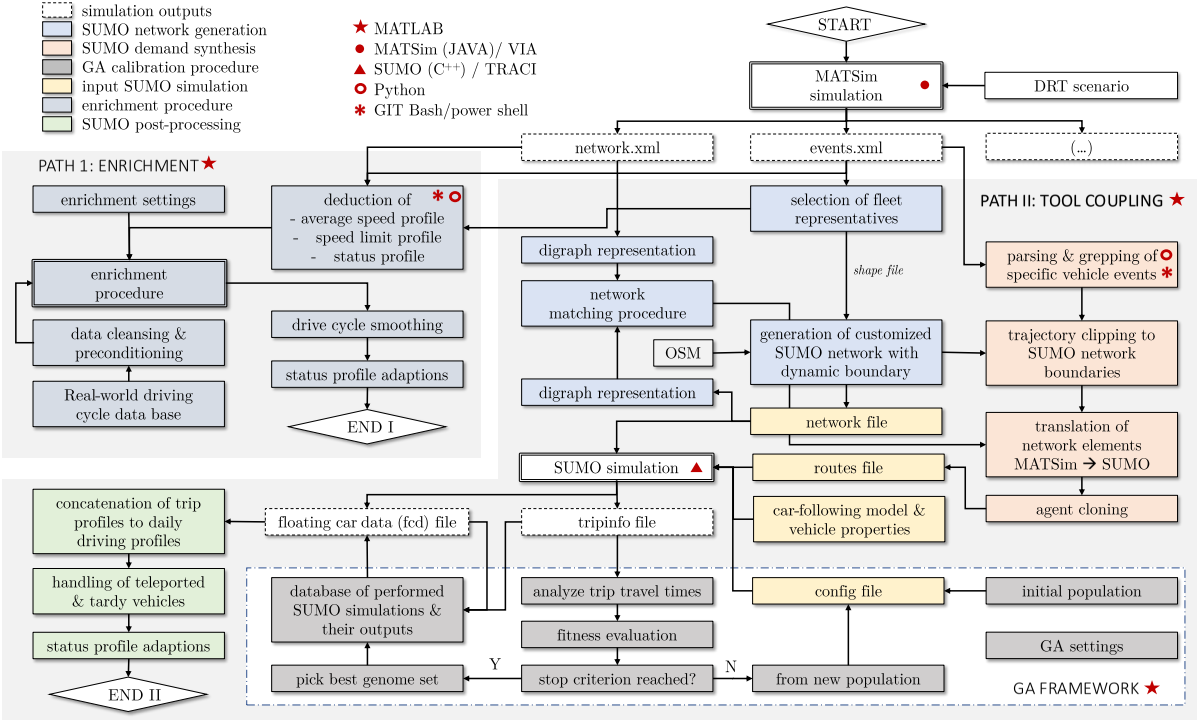

Figure 7.15: Final toolchain for deriving detailed driving profiles for SAEV requirement engineering. The toolchain consists of seven major components, each depicted in a distinct color. Corresponding software environments are indicated.

Part III

Tool-Chain Application and final Evaluation

8 Tool-chain application – a power engine centered study

The primary aim of this thesis is to refine SAEV engineering requirements. Our toolchain constitutes an important advancement in reaching this objective.

Critical factors influencing vehicle design include operating hours and mileage, directly affecting vehicle durability and lifetime. SAEV mobility patterns closely resemble those of urban taxi fleets, with data on average SAEV operating patterns available, e.g., from agent-based DRT simulation [34, 72, 155].

However, autonomous driving capabilities introduce further changes in driving pattern and behavior, thereby altering the operational conditions of the vehicle and its components. These second-order effects necessitate a comprehensive approach integrating mesoscopic transport planning, microscopic traffic simulation and nanoscopic vehicle simulation. While this thesis primarily addresses the meso-micro coupling aspect, we employ a well-established vehicle simulation model [190] here.

In the following sections, we apply our toolchain in a power-engine-centered study, aiming to identify essential operating points for electrical machines (EMAs) in SAEVs. We thereby seek to optimize EMA design to meet the specific requirements of SAEVs. The chapter begins with an exploration of technical aspects regarding EMAs in Section 8.1, providing a necessary background for understanding subsequent discussions. We then present the general setup of our methodological approach (Sec. 8.2), followed by a description of our specific use case setup (Sec. 8.3). Finally, in Section 8.4, we interpret the study results, offering valuable insights into the complex dynamics between evolving mobility needs, altered operational conditions, and vehicle design.

8.1 Efficiency of electric drive systems for private cars

Modern EV motors designed for personal cars require versatile performance, operating efficiently across a wide torque and speed[1] range. Efficiency maps, typically presented as contour plots, depict the maximum efficiency within a torque-speed plane, aiding engineers in selecting optimal operating points for critical vehicle maneuvers like acceleration, cruising, and regenerative braking. The goal is to minimize energy losses, ensuring the electric propulsion system operates with optimal efficiency and overall performance.

The top panel of Figure 8.1 displays exemplarily the efficiency map for a permanent magnet synchronous motor (PMSM) and its inverter, showcasing notable efficiency under both full and partial loads. This efficiency is attributed to the remarkable power density of PMSMs, which achieve significant power output relative to their compact size and weight, making them well-suited for EV propulsion systems. However, the high costs of rare-earth magnet materials crucial for PMSMs have driven research to further optimize motor efficiency and performance, aiming to improve cost-effectiveness and sustainability.

The efficiency of an EMA and its associated inverter (INV) is influenced by various factors, with the EV battery's direct current voltage (U_{DC}) playing a significant role. Lithium-ion batteries, common in electric vehicles, exhibit nonlinear voltage characteristics during discharge, decreasing as the state of charge (SOC) declines. This reduction in U_{DC} requires higher inverter currents to maintain constant power ($P = U \cdot I$), resulting in elevated resistive losses ($P_{\Omega} = I^2 R$) and decreased inverter efficiency. The left panel at Figure 8.2 illustrates how higher efficiency areas contract with reduced U_{DC} and shift to lower torque and speed, with the maximum torque limit declining in constant power regions.

The right panel of Figure 8.2 demonstrates the significant impact of temperature on the drive system efficiency, highlighting the importance of effective cooling systems. Semiconductor devices in the inverter perform better at lower temperatures, while higher temperatures increase resistance and reduce switching

[1] In the domain of electric drives, the term "speed" refers specifically to the rotational velocity of the EMA, distinct from the velocity of the vehicle.

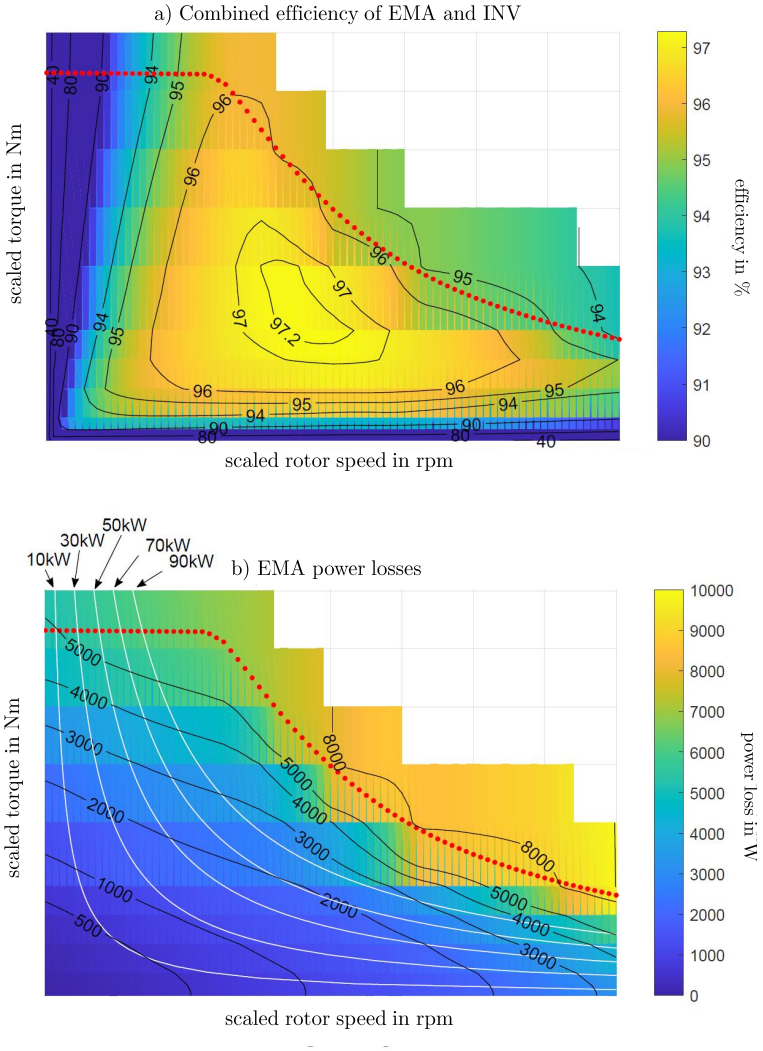

Figure 8.1: Efficiency (top) and power loss (bottom) map derived from a 2D Finite Element Method (FEM) based simulation for a PMSM (U_{DC} = 360V, stator/rotor temperature $T_{R,S}$ = 80°). By dividing the motor into small elements, FEM calculates the distribution of electromagnetic fields, voltages, currents, and forces within the motor. The red dashed line marks the maximum torque line, representing the upper limit of torque values attainable by the EMA at different speeds. The white lines in the bottom panel denote constant EMA power lines.

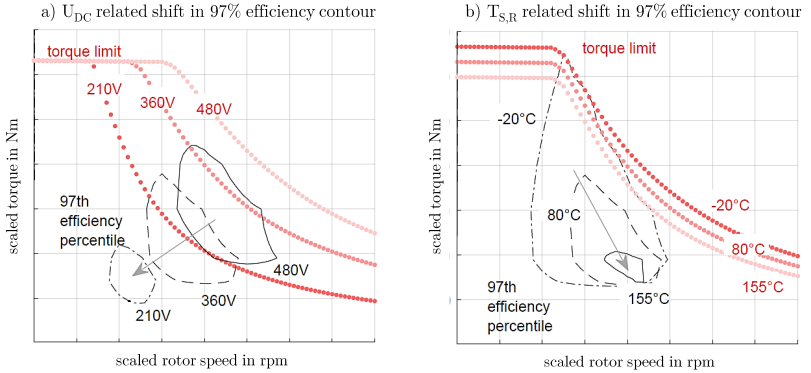

Figure 8.2: Impact of the battery's DC voltage (U_{DC}, left panel) and the EMA's stator/rotor temperature ($T_{S,R}$, right panel) on the 97% efficiency contour, and the EMA's maximum torque exemplified by a 200kW PMSM.

speeds, leading to higher resistive losses in EMA and INV components. As a result, both efficiency and maximum torque decrease significantly.

EMA losses within an EV propulsion system vary across different operating conditions, as depicted in the lower panel of Figure 8.1. As outlined in Table 8.1, ohmic losses are significant in regions of constant torque, crucial for acceleration, while core and converter losses dominate in constant power regions [186]. Regions of constant torque denote areas of low speed, where torque remains relatively constant as long as the current is kept constant. With increased motor speeds, higher back-electromotive forces (EMF) are induced, opposing the applied voltage. The transition from constant torque to constant power occurs when back-EMF equals applied voltage, leading to decreasing torque with speed while maintaining relatively constant power output. Operational transitions toward regions of constant power are essential for cruising or high speeds.

8.2 Methodological approach

Optimizing vehicles for specific use cases, such as parcel delivery or SAEVs, necessitates distinct vehicle architectures and component designs compared

Table 8.1: Significance and variation levels of each loss type as a percentage of the maximum power for an interior PMSM, depicted for both constant torque and constant power regimes. Data sourced from [186]. *Ohmic losses:* Heat generated due to electrical resistance in motor windings. *Core losses:* Energy dissipated in the motor's magnetic core due to hysteresis and eddy currents. *Permanent magnet (PM) losses:* Energy losses in the permanent magnets due to demagnetization effects. *Converter losses:* Power losses in the motor drive converter, including switching and conduction losses. *Mechanical losses:* Energy losses due to friction and windage within the motor, such as bearing friction and air resistance. *Transmission system losses:* Losses incurred in transferring mechanical power from the motor to the load, including losses in gears, or shafts.

operating region	ohmic loss	core loss	PM loss	converter loss	mechanical loss	transmission system loss
constant torque	0.1-13%	0.01-1.5%	0.001-0.5%	0.01-9%	0.1-1%	5-6%
constant power	0.1-5%	1-15%	0.001-1%	1-12%	0.5-3%	4-15%

to those optimized for average PC usage, typically assessed by the *Worldwide harmonized Light vehicle Test Procedure* (WLTP). Addressing diverse and future usage scenarios requires new vehicle development and optimization tools to be applied on both architectural and component levels.

[190] offers a holistic EV simulation tool based on Gamma Technologies GT Suite[2] to optimize EV architecture and powertrain design for specific use cases. The vehicle simulation necessitates detailed usage profiles, encompassing velocity, status, and altitude time-series data, typically describing the operational patterns of a particular vehicle across a sufficiently long time period. Throughout the simulation, all energy transfers including mechanical, electrical, and notably thermal flows within the vehicle and its subsystems are considered. This evaluation extends beyond driving to include parking and charging phases, enabling the determination of annual energy consumption. As a result, the tool provides a holistic energy flow description within EVs, enabling the comparison of different vehicle and powertrain configurations while assessing specific efficiency measures. Additionally, it generates relevant time-series data on energy flows and load profiles for various vehicle components, aiding in deriving future system engineering requirements. Concerning EMA design, it produces diverse torque-speed planes, illustrating commonly frequented operating points and their efficiency.

[2] GT-SUITE, https://www.gtisoft.com/gt-suite/. Accessed: Sep. 29, 2024.

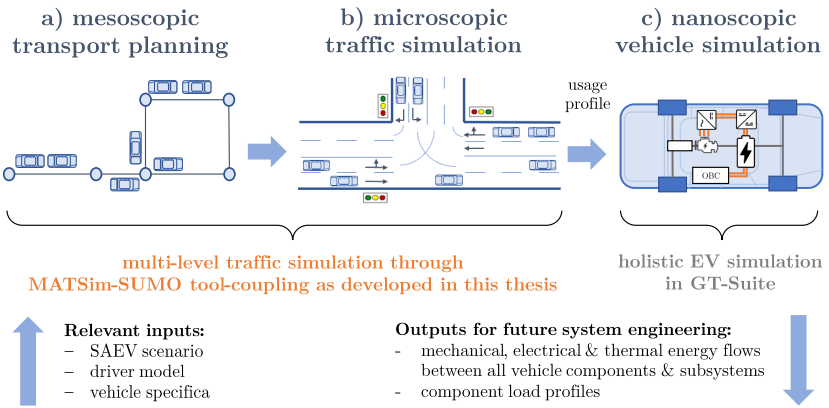

Figure 8.3: Schematic representation of the methodology for deriving automotive system requirements for future mobility concepts. Multi-level traffic simulation generates key usage patterns for SAEVs, including velocity, status, and altitude profiles. These patterns are fed to holistic EV simulation [190], facilitating the comparison of vehicle and powertrain topologies, while assessing specific efficiency measures. The methodology enables the derivation of load profiles for all relevant EV system components.

Integrated with multi-level traffic simulation (refer to Figure 8.3), the EV simulation leverages an extensive database of present and future mobility patterns. With the incorporation of our devised toolchain, formerly estimated SAEV driving profiles are now methodologically derived, enabling the systematic derivation of automotive system requirements.

8.3 Study design

In the following, we apply the toolchain depicted in Figure 8.3 to analyze the impact of shifting vehicle use patterns and driving behavior on EMA load profiles and highlight implications for EMA design. Microscopic driving profiles for both PC and SAEV use pattern are deduced over a 24 hour period. PC data is sourced from the *Mobilität in Deutschland (MiD 2018)* study [170] and converted into FKFS-cycle-based driving cycles (cf. Figure 3.1), representing a mixed driving pattern with various road types and a broad velocity spectrum (cf. Figure 8.4 left side). SAEV driving profiles are derived from a 10% MATSim

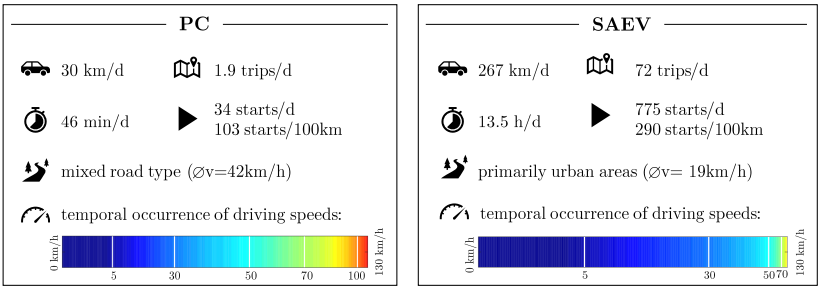

Figure 8.4: Daily KPIs of privately owned passenger cars and shared, autonomous electric vehicles. PC data sourced from *MiD 2018* study [170] and FKFS cycle, representing real-world driving in the Stuttgart region. Median SAEV KPIs derived from 10% MATSim DRT simulation, processed with our toolchain for detailed driving dynamics.

simulation of a DRT fleet in Berlin's downtown area, focusing on urban driving dynamics characterized by moderate vehicle speeds (cf. Figure 8.4 right side).

Figure 8.4 demonstrates notable disparities in daily driving KPIs between an average PC and SAEV. SAEVs cover nine times the mileage of PCs, have operation periods eighteen times longer, and experience considerably more stop-and-go events due to their urban service area. Velocity profiles for PC and SAEV (in conjunction with corresponding vehicle status and altitude profiles) are fed to subsequent vehicle simulation (cf. Figure 8.3) with its EV model relying on a 150 kW PMSM.

To ensure a sufficiently populated map of EMA operating points for PC usage, velocity profiles were extended from the initial 24 hour period to six days by incorporating additional FKFS cycles. For consistency, SAEV driving profiles were expanded to the same calendric time frame by selecting additional vehicles from the DRT fleet with median driving KPIs.

8.4 Modified power engine requirements for SAEV fleets

The modified driving patterns of SAEVs result in distinct frequency variations of EMA operating points (OPs) compared to PCs, as shown in Figure 8.5. In our case study, SAEVs exhibit considerably higher OP utilization rates due to their extended operational period and narrower range of rotor speeds, typically up to 9 000 rpm, compared to PCs, which can reach up to 13 000 rpm. While PCs show frequent OPs around 5 000 rpm and 7 500 rpm, SAEVs spend more time at low rotor speeds of up to 4 000 rpm. Contrary to our expectations, there is no noticeable broadening in torque spectra observed for PCs. It is important to note, however, that the used PC driving cycles involved specifically trained drivers, potentially excluding more aggressive driving behaviors typically encountered in real-world scenarios.

We supplement the analysis by overlaying the frequency distribution of EMA operating points with their corresponding efficiencies, thereby estimating the OP-specific energy losses over the duration of respective PC and SAEV driving cycles. This metric serves as a fundamental guiding principle for pinpointing the areas where optimization of the electric drive (ED) subsystem, comprising the EMA and INV, can be most effectively pursued. In this matter, we distinguish between OP-specific efficiency $\eta^{\mathrm{ED}}_{M,n}$ across the torque-speed plane and the overall roundtrip efficiency $\bar{\eta}_{\mathrm{ED}}$ throughout the driving cycle.

The OP-specific efficiency is defined as the ratio of mechanical power (P_{mech}) generated by the EMA to the power supplied by the EV battery (P_{Bat}) during forward motoring ($M, n \geq 0$):

$$\eta^{\mathrm{ED}}_{M,n} = \frac{P_{\mathrm{mech}}}{P_{\mathrm{Bat}}}. \tag{8.1}$$

For rotation against a torque M, the mechanical power is given by

$$P_{\mathrm{mech}} = \vec{M} \cdot \vec{\omega} \tag{8.2}$$

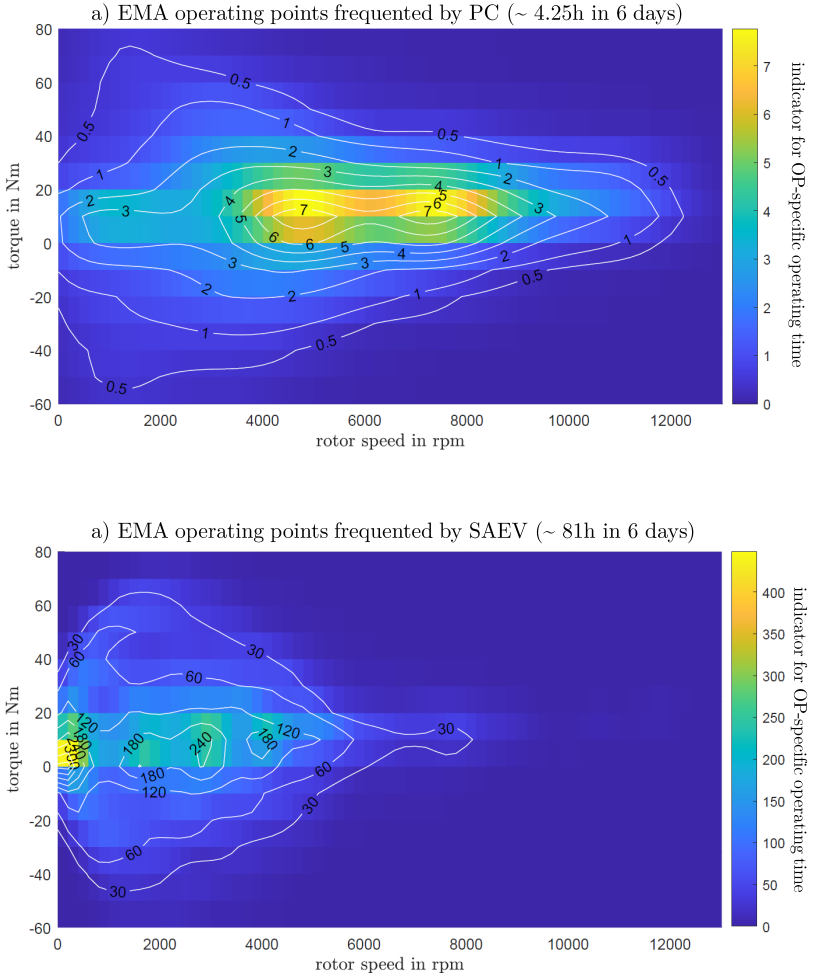

Figure 8.5: Comparison of EMA operating points frequented by PC and SAEV over 6 days. Colored areas serve as indicators for the EMAs' OP-specific operating time, facilitating cross-panel comparability. Yellow areas highlight the most frequented OP regions, while bluish areas represent the least frequented ones. Contour lines delineate areas of similar operating duration, highlighting significantly higher SAEV utilization rates. SAEVs demonstrate longer operating times that manifest in smaller torque-speed ranges.

where $\vec{\omega} = \frac{\mathrm{d}\varphi}{\mathrm{d}t}\ \vec{e}$ represents the angular velocity about an axis parallel to the direction vector $\vec{e}$. For a shaft with torque M and rotational speed $n = \frac{\omega}{2\pi}$, the shaft power is determined by

$$P_{\text{mech}} = M\, 2\pi\, n \tag{8.3}$$

which leads to

$$\eta^{\text{ED}}_{M,n} = \frac{M\, 2\pi\, n}{P_{\text{Bat}}}. \tag{8.4}$$

To compute the overall roundtrip efficiency $\bar{\eta}_{ED}$ of the electric drive system, a shift from a power-centric to an energy-centric perspective in efficiency calculation is necessary. Simply averaging all OP efficiencies $\eta^{\text{ED}}_{M,n}$ based on their temporal occurrence overlooks the significance of realized or lost power. For instance, a low-power operating point with low efficiency may not significantly affect the overall loss balance, as its realized losses are marginal compared to high-power OPs with seemingly high efficiency (cf. bottom panel of Figure 8.1). In other words, although a low-efficiency EMA OP results in high relative energy losses, its low power throughput makes these losses negligible in absolute terms. Therefore, we utilize cumulative power values derived from vehicle simulation to calculate the ED's overall efficiency, accommodating both forward motoring and regenerative braking.

As depicted in Figure 8.6 by the bluish energy flows, during forward motoring, the input to the ED system is the energy supplied by the high voltage system E_{Hv2Inv}[3]. Conversely, its output is the energy E_{Ema2Tra} forwarded to the transmission system. During braking (recuperation, highlighted in reddish tones), the energy flow reverses, and the EMA converts the mechanical energy transmitted by the transmission system into electric energy (E_{Tra2Ema}). This electric energy is then returned to the battery via the high voltage system (E_{Inv2Hv}). The roundtrip efficiency $\bar{\eta}_{\text{ED}}$ is defined as:

[3] Note that E_{Hv2Inv} may differ from the energy sourced from the battery (E_{Bat2Hv}) due to additional electrical loads in the low voltage system.

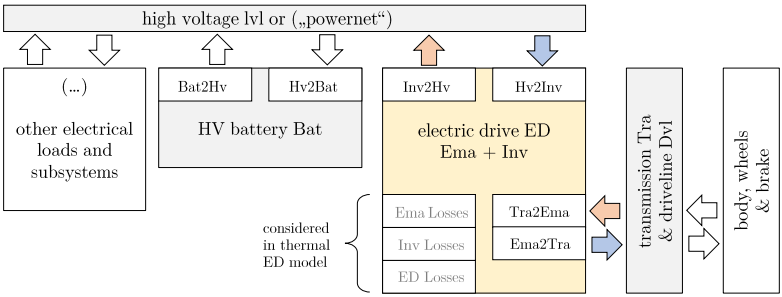

Figure 8.6: Extract of the energy flow diagram, generated through EV simulation in GT-Suite [190], depicting crucial energy pathways associated with an EV's electric drive system, comprising the electric machine and inverter.

$$\bar{\eta}_{\mathrm{ED}} = \frac{\int P_{\mathrm{ED,out}}}{\int P_{\mathrm{ED,in}}} = \frac{E_{\mathrm{Inv2Hv}} + E_{\mathrm{Ema2Tra}}}{E_{\mathrm{Hv2Inv}} + E_{\mathrm{Tra2Ema}}}. \tag{8.5}$$

Figure 8.7 depicts the cumulative power losses at specific OPs of the electric drive system. For SAEVs, significant losses occur at low torque (0 to 30 Nm) and moderate speeds (1500 to 5000 rpm), contrasting with PCs where losses peak at higher speeds (7000 to 8000 rpm). Hence, the optimization of ED efficiency for SAEVs ought to concentrate on zones characterized by moderate speed and low torque to efficiently alleviate overall energy losses, deviating from the conventional approach influenced by PCs, which typically prioritizes high-power regions. Conventional EVs, often in the higher price range, prioritize high-power machines more for marketing and emotional reasons than as a technological necessity. Therefore, electric machines in these EVs are often oversized and optimized for high-power OPs, crucial for tasks like acceleration and hill climbing. SAEVs, in contrast, would greatly benefit from alternative optimization strategies that prioritize efficiency at partial loads, aligning with their urban driving, low-speed conditions and smoother driving style.

Table 8.2 compiles KPIs relevant to power engines, derived from vehicle simulations of both PC and SAEV. Despite SAEVs exhibiting significantly greater mileage and uptime in our use case, their average energy consumption is reduced by 12%, mainly due to their refined driving style, lower urban speeds, and increased recuperation gains. In this context, SAEVs additionally benefit from

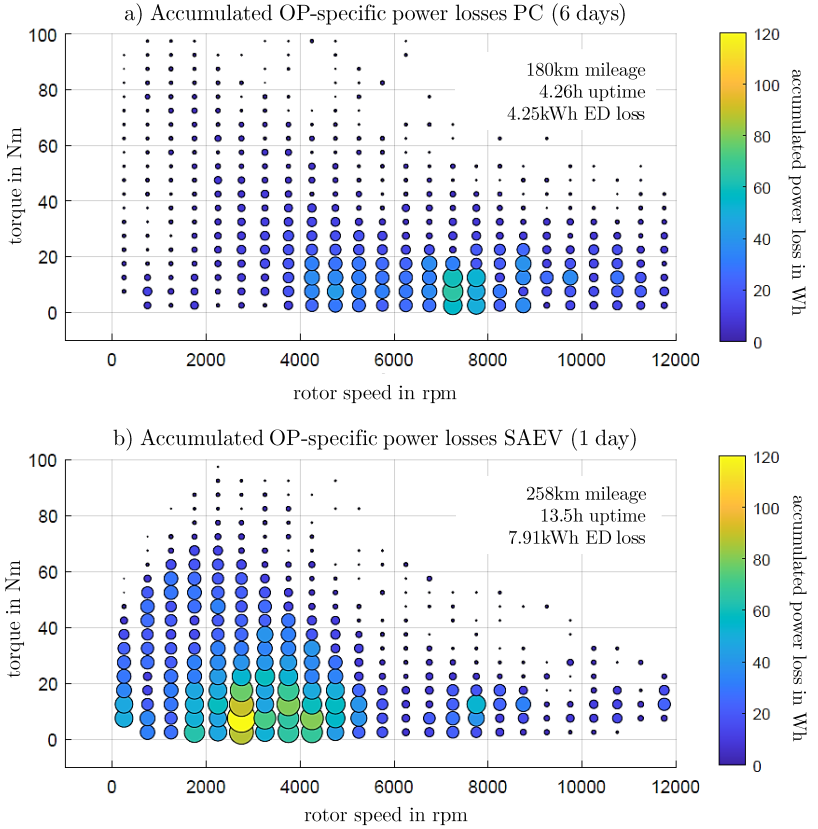

Figure 8.7: Scatter plots depicting the accumulated power losses experienced by an electric drive over a 6-day period for PC usage (top) and a 1-day interval for SAEV usage. In both plots, the size and color of each data point represent the magnitude of energy losses, with smaller bluish dots indicating regions of low energy loss and larger yellowish scatter signifying areas with higher energy losses. Additionally, KPIs related to the respective drive cycles are presented in the upper right corner of each panel for contextualization. Notably, scatter points characterized by larger size and a greenish-yellowish color exhibit substantial potential of efficiency improvements, making them particularly impactful.

Table 8.2: Compilation of power-engine relevant KPIs for PCs and SAEVs, derived from vehicle simulation using corresponding driving cycles over 6 days. The table also shows the relative growth of SAEV KPIs compared to PCs for clarity. The electric drive (ED) encompasses both electric machine and inverter.

KPI	PC	SAEV	growth	
mileage in km	179.7	1540.2	757 %	↑
uptime in h	4.26	81.02	1802%	↑
average speed in km/h	42.2	19.0	-55%	↓
consumption in kWh/100km	11.7	10.3	-12%	↓
mean ED efficiency in %	87.9	84.9	-3.4%	↓
accumulated ED energy losses in kWh	4.25	47.12	1009%	↑
mileage normalized ED energy losses in kWh/100km	2.37	3.06	29%	↑

second-order effects, given their sustained operation with minimal cooldown intervals, thereby diminishing the necessity for energy-intensive preconditioning procedures associated with the vehicle's battery or cabin. However, SAEVs show a lower average efficiency of the electric drive system, calculated using Equation 8.5. This decreased efficiency results from suboptimal EMA design[4], prioritizing efficiency under full load conditions rarely encountered in SAEV operation. Even when normalized for mileage, energy losses of the EMA and inverter in SAEVs are increased by 29% compared to PC usage.

Scientific consensus acknowledges that battery electric vehicles (BEVs) require less maintenance and repair than conventional vehicles [87]. However, when considering AVs, factors supporting and opposing this trend arise. While AVs prioritize comfort and often experience less aggressive driving, enhancing component longevity, higher daily mileage and uptime, particularly in SAEVs, lead to increased wear and tear on replaceable parts, potentially reducing the lifespan of EMA and other EV components. According to our study, a typical SAEV accumulates the runtime equivalent of a PC over a year in just 19 days. Furthermore, PCs typically cover annual distances ranging from 13,000 to 16,000 km [170]. SAEVs achieve this mileage in just 2 months according to our DRT simulation. This stark difference underscores the need for SAEV

[4] The study's EMA model is based on a PMSM optimized for PC usage.

manufacturers to reassess the lifespan of fleet vehicles and consider potential powertrain updates to ensure safety standards are upheld.

In closing, let us consider the hypothetical scenario of new EMA design specifically tailored to meet the operational demands of SAEVs. With 14,127 fleet vehicles (cf. DRT scenario in Table 5.1) and moderate energy prices (10 ct/kWh), even a modest 3 percent increase in the mean ED efficiency can lead to substantial annual savings (0.81 million euro) due to reduced energy losses, as shown in Table 8.3. This underscores the importance for future SAEV fleet operators to prioritize investments in EVs equipped with more efficient powertrain components.

Table 8.3: Economic impacts of enhanced ED efficiency (1/3/5% improvement) for SAEVs at various energy prices (5/10/20 ct/kWh). Results are based on the current ED efficiency of 85% and associated ED energy losses of 47.12 kWh over 6 days. Annual savings offer insights into the additional expenditure that SAEV fleet providers might consider for investing in optimized ED components. All savings are presented in terms of the economic impact per one average SAEV vehicle.

assumed ED efficiency gain	1% ↑	3% ↑	5% ↑	
accumulated ED losses	43.98	37.70	31.41	kWh/6d
accumulated annual ED losses	2675.37	2293.17	1910.98	kWh/a
saved annual ED losses	191.10	573.29	955.49	kWh/a
5 ct/kWh	9.55	28.66	47.77	€/SAEV/a
10 ct/kWh	19.11	57.33	95.55	€/SAEV/a
20 ct/kWh	38.22	114.66	191.10	€/SAEV/a

9 Conclusion and Outlook

Summary of Contributions The principal contribution of this work lies in the development of a two-pathed toolchain designed to facilitate the derivation of SAEV driving cycles for automotive requirement engineering. Central to this contribution is the successful integration of the mesoscopic transport planning framework MATSim with the microscopic traffic simulation tool SUMO.

A significant academic achievement of this work lies in bridging the gap between two inherently incompatible simulation frameworks of different representation levels. Given the fundamental differences in network representation and traffic dynamics, seamless co-simulation of MATSim and SUMO was deemed unfeasible. However, through leveraging additional degrees of freedom inherent to our vehicle-centric use case, we devised a sequential tool-coupling approach. This approach adopted a novel (automotive-inspired) perspective on meso-micro model calibration, prioritizing the consistency of ego-vehicle travel times over traffic counts. This emphasis enabled the derivation of detailed driving dynamics from large-scale DRT fleet simulation.

Notably, our toolchain exhibits high robustness, excelling in scenarios with limited data availability and imperfect microscopic networks. Unlike many large-scale microscopic traffic simulations that struggle with extensive network cleaning and travel demand synthesis efforts, our approach minimizes these challenges by subduing microscopic modeling effort to the less restrictive requirements of mesoscopic transport planning.

We also developed a secondary, more pragmatic (albeit less measure-responsive) data-driven solution for deriving SAEV drive cycles. This alternative method involves enhancing mesoscopic speed profiles in MATSim by incorporating real-world driving data, allowing for detailed driving dynamics to be derived from purely mesoscopic transport planning.

In essence, our contributions span both pragmatic data-driven solutions and sophisticated model-based methodologies. The versatility of our toolchain

makes it a valuable asset to investigate the impact of new mobility concepts on vehicle usage and component load profiles. This not only aids the automotive industry in designing future vehicle systems but also advances the field of multi-level traffic simulation and modeling.

Revisit of Research Objectives Initially, driven by the need for dependable SAEV driving profiles in automotive requirement engineering, our primary objective was to devise a method capable of deducing these driving cycles while adhering to specific criteria (see *KR0-5* in Section 1.4).

Originating from this industry-driven objective, our task was to identify SAEV travel demand in a scenario-based manner (*KR0*). We aimed to extract driving profiles that accurately depicted the daily movements of SAEVs across entire metropolitan regions, considering all EV range and charging constraints, as well as diverse DRT routing and dispatching strategies (*KR1*). These profiles needed to encompass specific vehicle states (*KR2*), including *charging*, *idling*, or *occupied*, to facilitate optimal climate control and battery management. The primary emphasis lay in modeling second-by-second velocity profiles with intricate driving dynamics (*KR3*), reflecting a wide spectrum of driving behaviors, traffic states, and transport infrastructures for vehicle simulation. Furthermore, the incorporation of additional time-series data, such as vehicle-specific altitude and occupancy profiles (*KR4*), was deemed advantageous for SAEV prototyping and component design. Due to the application-driven nature of this thesis, practical feasibility (*KR5*) was prioritized over strict adherence to specific scientific methodologies.

Over time, our initially industry-oriented research objective evolved into a more academic pursuit, shifting towards the exploration of multi-level traffic simulation (cf. Figure 9.1). This evolution eventually led to the development of a sequential tool coupling between mesoscopic transport planning in MATSim and microscopic traffic simulation in SUMO. Bridging the gap between these inherently incompatible frameworks of different representation levels presents an academic challenge that has not been adequately investigated or practically demonstrated thus far.

Due to the broad scope of the thesis, we deliberately relied on existing functionalities from MATSim and SUMO to meet our SAEV related key requirements (*KR0-4*) to strengthen our focal point on the actual tool-coupling development

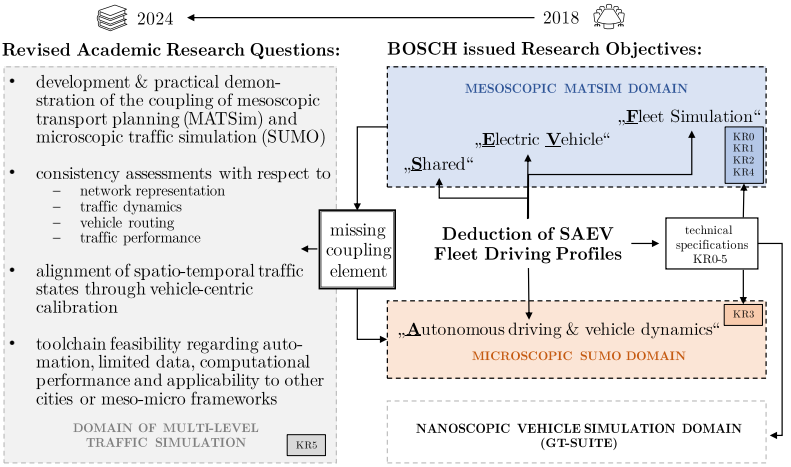

Figure 9.1: Evolution of research objectives: The transition from industry-driven to academic inquiry in automotive research. Initially, prompted by the need for dependable SAEV driving profiles for automotive requirement engineering, specific drive cycle criteria (*KR0-5*) were outlined for SAEV prototyping. Over time, the focus shifted towards a more academic exploration, culminating in the coupling of mesoscopic transport planning in MATSim with microscopic traffic simulation in SUMO. The figure further illustrates the allocation of various SAEV modeling components to the respective MATSim and SUMO domains.

and practical demonstration. So, in Chapter 2, we leveraged MATSim's capability to simulate and track electric DRT fleets within large-scale multi-modal transport environments. This approach addressed the requirements *KR0, KR1, KR2*, and *KR4* as outlined in Section 1.4.

To address MATSim's limitations in traffic dynamic modeling, we developed two approaches to deduce realistic velocity profiles (*KR3*) in Chapter 3. First, we enhanced mesoscopic driving cycles from MATSim simulations with real-world data using data-driven methods. However, this enrichment method showed limited sensitivity to various driving styles and autonomous driving features like platooning. To overcome this, and to reduce the need for resource-intensive on-road measurements, we also explored microscopic traffic simulation in the same chapter. This provided a more responsive, albeit more complex, model-based approach to fulfill *KR3*.

To achieve our revised objective of meso-micro tool-coupling, we developed automated methods for building and populating microscopic SUMO models within areas of interest in MATSim. This included implementing a robust network matching routine to translate mesoscopic network attributes to their microscopic counterparts (Chapter 4). Furthermore, we created a technique to disaggregate MATSim's sampled travel demand into the higher-resolved microscopic simulation domain across dynamic meso-micro network borders (Chapter 5). Using stochastic methods, we minimized computational efforts in the microscopic part by reducing the share of DRT fleet vehicles being transferred, while maintaining fleet representativeness. Throughout our toolchain development, meeting the requirements of *KR5* was our top priority.

Our primary focus was on establishing consistency between MATSim and SUMO, a prerequisite for successful tool-coupling. In Chapter 6, we conducted various experiments to analyze the conformity of both frameworks in terms of traffic dynamics and network capacity. The results revealed inherent non-compatibility between the frameworks, thwarting our initial aspirations for meso-micro co-simulation.

To overcome this challenge, we devised an alternative toolchain design in Chapter 7 that effectively aligns traffic states spatio-temporally in both meso and microscopic simulation components. This involved a novel calibration approach tailored for microscopic traffic simulations, focusing on reproducing observed travel times of individual vehicles rather than traffic counts.

In alignment with the automotive context of the thesis, we demonstrated the toolchain's feasibility in a study centered on the design of power engines in Chapter 8. Specifically, we exemplified the deduction of modified power engine requirements arising from distinct driving patterns and behaviors inherent to SAEV fleets compared to privately-owned passenger cars through vehicle simulation.

Limitations, Methodological Reflection and Areas of Future Research

Our toolchain's practical feasibility is subject to certain limitations. Notably, it primarily confines to our automotive-inspired use case where precise modeling of microscopic driving patterns and associated ego-vehicle dynamics takes precedence over the accurate representation of traffic conditions. While a multi-level co-simulation approach is desirable for broader applicability, its

feasibility remains uncertain and necessitates substantial research efforts, as elucidated in Section 6.1. In hindsight, while reflecting on our methodological approach, it would have been beneficial to adopt a more open-minded approach to multi-level co-simulation, starting with the careful selection of inherently compatible tools, as exemplified by prior efforts [37]. However, the tool choice of MATSim and SUMO was fixed by the customer.

Our toolchain relies on an existing travel-time calibrated mesoscopic transport model. While theoretically transferable to other metropolitan areas, its practical feasibility hinges on the availability of such a model. Constructing and calibrating a new MATSim model from scratch using surveys and land-use data is resource-intensive, defying automotive business economics. To circumvent this, we suggest outsourcing future model construction to SENOZON AG[1], which offers customer-specific mobile-based demand data for MATSim integration, leveraging MATSim's recent applications in epidemiological research [158, 157, 160, 159].

We recognize the need for further improvements in the user-friendliness and performance of our toolchain, especially regarding automation capability and computational efficiency in large-scale scenarios. Firstly, meaningful simulation initiation and effective troubleshooting still require a significant background in traffic engineering. Secondly, despite efforts to minimize manual intervention (e.g., toolchain's resilience to imperfect, uncleaned networks) and computational resources (e.g., extraction and microscopic simulation of fleet representatives in an area-minimized network), substantial resources are still necessary for a single toolchain run, requiring either a high-performance computing cluster or significant time investment.

A notable bottleneck is the genetic algorithm-based calibration process, which aligns spatio-temporal network states in SUMO with observed states in MATSim. Exploring more deterministic calibration approaches with superior pre-screening capabilities for local optima could reduce the need for repetitive or redundant microscopic simulations in SUMO.

Another area requiring improvement pertains to our approach in modeling autonomous driving behavior. Despite SUMO's various Car2X interfaces and suitable car-following models, we chose the simplistic Krauss model due to the

[1] SENOZON, https://senozon.com/. Accessed: Sep. 29, 2024.

relatively broad scope of our thesis. We prioritized our focus on meso-micro tool coupling over delving deeper into the intricacies of autonomous driving modeling. By doing so, we acknowledge the possibility of overlooking relevant issues in this area, warranting future investigation.

Theoretical and Practical Implications In the operational setting of an automotive supplier, our toolchain adeptly addresses the challenges posed by uncertain travel demands, usage patterns, and driving behaviors stemming from emerging mobility concepts. Although estimates exist for SAEVs' daily mileage and there is a consensus on a smoother driving style, the absence of "measure-sensitive" 24-hour velocity profiles has impeded detailed vehicle simulation and component design efforts to date. Our contribution addresses this limitation, offering a method for more precise SAEV prototyping.

Furthermore, our engagement in multi-level traffic simulation provides a foundation for the automotive industry to explore new business opportunities. Beyond its core focus on SAEVs, our toolchain offers diverse applications in the automotive sector by facilitating access to microscopic SUMO functionalities within mesoscopic transport simulation. This encompasses the evaluation of other emerging mobility concepts like urban air mobility, platooning vehicle convoys, or integrated passenger and freight transport. Additionally, it aids mobility service providers in fleet operations and enables virtual testing of Car2X technologies, traffic control innovations, and other initiatives aimed at fostering sustainable urban living.

From an academic standpoint, our automotive-focused study illustrates how companies without traditional traffic engineering backgrounds can innovate and enrich the conventional field of traffic and transport simulation with their novel requirements and applications. Our approach, emphasizing realistic driving behavior over traffic counts, offers a blueprint for future meso-micro coupling endeavors within the context of automotive requirement engineering. Our tool-coupling automates the creation, population, and calibration of microscopic SUMO models from mesoscopic MATSim data, making it appealing for vehicle-centered studies.

Despite the tool-coupling's capabilities, it is essential to recognize that the less intricate enrichment method likely suffices for most engineering cases. While the meso-micro tool-coupling offers greater versatility and measure-sensitivity, its

setup demands significant computational resources and time investment. From an entrepreneurial standpoint, this approach may not always be justified. Thus, subjecting every fleet scenario to subsequent SUMO simulation is impractical, as the benefits must outweigh the added effort. Depending on the use case, pure mesoscopic MATSim simulation (with or without sequential real-data enhancement) is often adequate. Sequential microscopic simulation is valuable for initial exploratory scenarios impacting vehicle design (cf. Figure 9.2), particularly where the enrichment method may miss relevant effects, such as driving in vehicle platoons.

Broader Significance in Automotive Context This thesis, initiated in 2018, maintains relevance in today's rapidly evolving automotive landscape. As the industry witnesses the decline of traditional ICE-based vehicles and awaits the full realization of autonomous driving and electric vehicles, significant consolidation looms, with emerging market dynamics yet to stabilize. Traditional players must navigate this complexity, balancing the need for agility with a cautious decision-making approach aimed at safeguarding employment stability.

In response to these challenges, automotive suppliers are compelled to diversify revenue streams and shift away from reliance on ICE technologies. The exploration of alternative travel modes presented in this thesis marks a step toward identifying new business opportunities, enabling scenario-based decision-making in this context. With intensifying competitiveness and cost pressures, there is a growing urgency for suppliers to adopt modular solutions adaptable to diverse use cases, departing from the prevalent practice of customized component development based on OEM specifications. Thus, embracing new model- or data-driven methodologies to determine universal vehicle requirements independently in a scenario-based manner becomes paramount, facilitating the development of modular component systems that enhance adaptability, reduce costs, and bolster competitiveness.

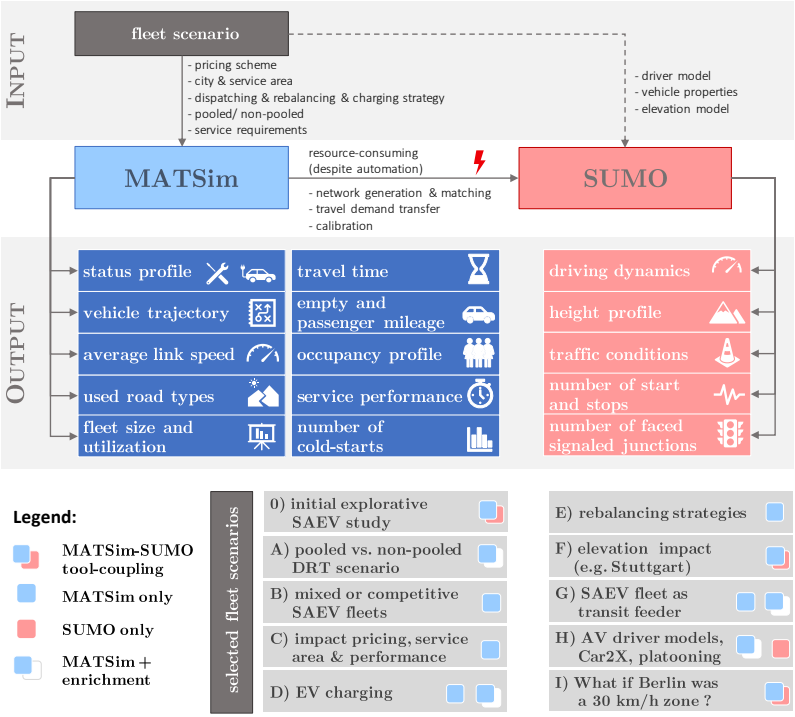

Figure 9.2: Comparison of simulation outputs for automotive SAEV requirement engineering: Blue text boxes denote outputs from pure mesoscopic transport simulation, while red ones represent additional results from microscopic traffic simulation. Compared to SUMO outputs, the enrichment procedure exclusively enhances the MATSim results with realistic driving dynamics. The figure further illustrates challenges in meso-micro integration and, depending on the preferred SAEV scenario, offers recommendations to use (a) pure mesoscopic transport planning, (b) pure microscopic traffic simulation, (c) mesoscopic simulation with real-data enrichment, and (d) meso-micro coupling.

A Appendix

A.1 Applied SUMO command line options

a) Conversion of .osm.pbf file to .osm file

```
osmconvert input.osm.pbf --out-osm -o=output.osm
```

b) Cutting .osm files using a boundary box (longitude, latitude)

```
osmconvert input.osm -b=minlon,minlat,maxlon,maxlat
--out-osm -o=extract.osm
```

c) Clip OSM file based on produced polyfile

```
osmosis --read-xml file=input.osm.xml --bounding-polygon
file=polyfile.poly --write-xml file=clipped.osm.gz
```

Osmosis is a command line Java application for processing OSM data and available at https://github.com/openstreetmap/osmosis, accessed: Oct. 03, 2024.

d) SUMO network import from OSM

```
netconvert --osm-files input.osm -o network.net.xml
--geometry.remove --ramps.guess --junctions.join --tls.guess-signals
--tls.discard-simple --tls.join --tls.default-type actuated
--keep-edges.by-vclass passenger --remove-edges.by-type highway.track
--roundabouts.guess --remove-edges.isolated --log network.log
```

e) Routing SUMO vehicles with DUAROUTER

```
duarouter -n network.net.xml --route-files trips.rou.xml.gz
-o duaroutes.rou.xml.gz --departlane best --departpos last
--departspeed max --arrivalpos max --remove-loops true
--repair.from true --repair.to true --xml-validation always
--error-log error_log.txt --ignore-errors true --routing-threads 4
```

f) Run SUMO simulation

```
sumo-gui -c testrun.sumocfg
```

The relevant options of the SUMO configuration file (in this case, `testrun.sumocfg`) are illustrated as an example in Section A.2.

A.2 Example SUMO configuration file (`.sumocfg`)

The SUMO configuration file requires two key input files: the `net-file` and `route-files`. In a basic setup, the `net-file` can be generated from OSM data using command **(d)** in Section A.1, which configures import heuristics and road network granularity. OSM files can be obtained through commands **(a)**, **(b)**, or **(c)**. Travel demand, represented by `route-files`, is generated using the DUAROUTER tool (command **(e)**), converting (in our case MATSim) trips into SUMO-compatible routes.

For the calibration procedure in Chapter 7, the configuration file must specify `summary-output` and `tripinfo-output` options for aggregated and detailed trip information, respectively, essential for our GA's objective function. The `fcd-output` option, which generates floating car data, is necessary for reconstructing microscopic driving cycles. The optimal processing and routing options in the configuration file are based on best practices and refined through calibration, as detailed in Chapter 7. The SUMO simulation is executed using command **(f)**, which invokes the specified configuration file to run the simulation scenario.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated on 06/26/20 19:03:17 by Eclipse SUMO Version 1.6.0
-->
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/
     sumoConfiguration.xsd">

    <input>
        <net-file value="final_network_397134901.net.xml"/>
        <route-files value="trips_397134901_80p_wego.rou.xml"/>
        <additional-files value="dua_dump_80p.add.xml"/>
    </input>

    <output>
        <summary-output value="summary_80p.xml"/>
        <tripinfo-output value="tripinfo_80p.xml"/>
        <fcd-output value="fcddata_80p.xml"/>
    </output>

    <time>
        <begin value="0"/>
    </time>

    <processing>
        <route-steps value="200"/>
        <no-internal-links value="false"/>
        <time-to-teleport value="40"/>
        <time-to-teleport.highways value="0"/>
        <eager-insert value="false"/>
        <ignore-route-errors value="true"/>
        <max-depart-delay value="720"/>
        <ignore-junction-blocker value="60"/>
    </processing>

    <routing>
        <device.rerouting.probability value="0.9"/>
        <device.rerouting.deterministic value="true"/>
        <device.rerouting.period value="60"/>
        <device.rerouting.adaptation-steps value="180"/>
        <device.rerouting.adaptation-interval value="1"/>
    </routing>

    <report>
        <verbose value="true"/>
        <no-warnings value="false"/>
        <log value="iteration_80p.sumo.log"/>
        <no-step-log value="true"/>
    </report>

    <random_number>
        <seed value="23423"/>
    </random_number>

</configuration>
```

Listing A.1: Structural composition of a SUMO configuration file (.sumocfg).

List of Figures

List of Tables

Bibliography

[1] Amit Agarwal, Gregor Lämmel, and Kai Nagel. "Incorporating within link dynamics in an agent-based computationally faster and scalable queue model". In: *Transportmetrica A: Transport Science* 14.5-6 (2017), pp. 520–541. DOI: 10.1080/23249935.2017.1364802.

[2] Amit Agarwal et al. "An Elegant and Computationally Efficient Approach for Heterogeneous Traffic Modelling Using Agent Based Simulation". In: *Procedia Computer Science* 52 (2015), pp. 962–967. DOI: 10.1016/j.procs.2015.05.173.

[3] European Environment Agency. *Air quality in Europe 2021*. 15/2021. Publications Office, 2021.

[4] European Environment Agency. *Transitions towards a more sustainable mobility system. TERM 2016 - transport indicators tracking progress towards environmental targe in Europe*. Publications Office, 2016. DOI: 10.2800/895670.

[5] European Environment Agency. *Decarbonising road transport: the role of vehicles, fuels and transport demand.* Publications Office, 2022. DOI: 10.2800/68902.

[6] Soyoung Ahn and Michael Cassidy. "Freeway traffic oscillations and vehicle lane-change maneuvers". In: *Transportation and Traffic Theory 2007* (2007), pp. 691–710.

[7] Glareh Amirjamshidi and Matthew J. Roorda. "Development of simulated driving cycles for light, medium, and heavy duty trucks: Case of the Toronto Waterfront Area". In: *Transportation Research Part D: Transport and Environment* 34 (2015), pp. 255–266. DOI: https://doi.org/10.1016/j.trd.2014.11.010.

[8] Anton Anastassov, Dongwook Jang, and Gavril Giurgiu. "Driving speed profiles for autonomous vehicles". In: *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017). DOI: 10.1109/ivs.2017.7995913.

[9] Kay W. Axhausen, Andreas Horni, and Hans J. Herrmann. *Final report: The risk for a gridlock and the macroscopic fundamental diagram*. en. Tech. rep. 2015. DOI: 10.3929/ethz-a-010433547.

[10] Dawn B. Valentine and Thomas L. Powers. "Generation Y values and lifestyle segments". In: *Journal of Consumer Marketing* 30.7 (2013), pp. 597–606. DOI: 10.1108/jcm-07-2013-0650.

[11] Bekir Bartin et al. "Calibration and validation of large-scale traffic simulation networks: a case study". In: *Procedia Computer Science* 130 (2018), pp. 844–849. DOI: https://doi.org/10.1016/j.procs.2018.04.076.

[12] Wilhelm Bauer et al. *ELAB 2.0 - Wirkungen der Fahrzeugelektrifizierung auf die Beschäftigung am Standort Deutschland*. de. Fraunhofer Publica. 2018. DOI: 10.13140/RG.2.2.29993.21605.

[13] Fabio Bellifemine et al. "Jade — A Java Agent Development Framework". In: *Multiagent Systems, Artificial Societies, and Simulated Organizations* (2005), pp. 125–147. DOI: 10.1007/0-387-26350-0_5.

[14] Richard Bellman. "On a routing problem". In: *Quarterly of applied mathematics* 16.1 (1958), pp. 87–90.

[15] Edward A. Bender and S. Gill Williamson. *Lists, Decisions and Graphs – With an Introduction to Probability*. University of California, San Diego, 2010.

[16] Roland Berger. *Automotive Disruption Radar ADR*. Accessed: Mar. 14, 2024. 2024. URL: https://www.automotive-disruption-radar.com/.

[17] Roland Berger. *Automotive Disruption Radar Issue 1- Tracking disruption signals in the automotive industry*. Tech. rep. Roland Berger GmbH, 2017.

[18] Roland Berger. *Automotive Disruption Radar Issue 10 - So close yet so far*. Tech. rep. Roland Berger GmbH, 2021.

[19] Roland Berger. *Automotive Disruption Radar Issue 11 - China back on top*. Tech. rep. Roland Berger GmbH, 2022.

[20] Roland Berger. *Automotive Disruption Radar Issue 12 - Reality bites. Global downturn hits disruption leaders in Europe and China*. Tech. rep. Roland Berger GmbH, 2022.

[21] Roland Berger. *Automotive Disruption Radar Issue 13 - Stability returns. Uncertainty eases as electrification race hots up*. Tech. rep. Roland Berger GmbH, 2023.

[22] Roland Berger. *Automotive Disruption Radar Issue 2 - Asia ahead*. Tech. rep. Roland Berger GmbH, 2017.

[23] Roland Berger. *Automotive Disruption Radar Issue 4 - China speeds ahead*. Tech. rep. Roland Berger GmbH, 2018.

[24] Roland Berger. *Automotive Disruption Radar Issue 5 - Mobility's great leap forward*. Tech. rep. Roland Berger GmbH, 2019.

[25] Roland Berger. *Automotive Disruption Radar Issue 6 - Time to think smart*. Tech. rep. Roland Berger GmbH, 2019.

[26] Roland Berger. *Automotive Disruption Radar Issue 7 - Challenges abound*. Tech. rep. Roland Berger GmbH, 2020.

[27] Roland Berger. *Automotive Disruption Radar Issue 8 - Autonomous mobility is coming*. Tech. rep. Roland Berger GmbH, 2020.

[28] Roland Berger. *Automotive Disruption Radar Issue 9 - The electric break through*. Tech. rep. Roland Berger GmbH, 2021.

[29] Roland Berger. *Fast forward after the Covid-19 crisis - How suppliers can successfully adjust their strategies for the future*. Tech. rep. Roland Berger GmbH, 2020.

[30] Robert Bertini and Monica Leal. "Empirical Study of Traffic Features at a Freeway Lane Drop". In: *Journal of Transportation Engineering* 131 (2005). DOI: 10.1061/(ASCE)0733-947X(2005)131:6(397).

[31] Lasse Bienzeisler et al. "Uncertainty and Variability Analysis of Agent-Based Transport Models". In: *Transportation Research Procedia* 62 (2022), pp. 719–726. DOI: 10.1016/j.trpro.2022.02.089.

[32] J. Bischoff, M. Maciejewski, and K. Nagel. "City-wide shared taxis: A simulation study in Berlin". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 275–280. DOI: 10.1109/ITSC.2017.8317926.

[33] Joschka Bischoff and Michal Maciejewski. "Agent-based Simulation of Electric Taxicab Fleets". In: *Transportation Research Procedia* 4 (2014), pp. 191–198. DOI: 10.1016/j.trpro.2014.11.015.

[34] Joschka Bischoff and Michal Maciejewski. “Simulation of City-wide Replacement of Private Cars with Autonomous Taxis in Berlin”. In: *Procedia Computer Science* 83 (2016), pp. 237–244. DOI: 10.1016/j.procs.2016.04.121.

[35] Joschka Bischoff et al. “Impacts of vehicle fleet electrification in Sweden – a simulation-based assessment of long-distance trips”. In: *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2019. DOI: 10.1109/mtits.2019.8883384.

[36] Vuk Bogdanović et al. “The Research of Vehicle Acceleration at Signalized Intersections”. In: *PROMET - Traffic&Transportation* 25.1 (2013), pp. 33–42. DOI: 10.7307/ptt.v25i1.1245.

[37] Emmanuel Bourrel and Jean-Baptiste Lesort. “Mixing Microscopic and Macroscopic Representations of Traffic Flow: Hybrid Model Based on Lighthill–Whitham–Richards Theory”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1852.1 (2003), pp. 193–200. DOI: 10.3141/1852-24.

[38] Joseph L. Bower and Clayton M. Christensen. “Disruptive Technologies: Catching the Wave”. English. In: *The Journal of Product Innovation Management* 13.1 (1996), pp. 75–76.

[39] Lars Briem, Nicolai Mallig, and Peter Vortisch. “Creating an integrated agent-based travel demand model by combining mobiTopp and MATSim”. en. In: *Procedia Computer Science* 151 (2019), pp. 776–781. DOI: 10.1016/j.procs.2019.04.105.

[40] Parsons Brinckerhoff. *Transit Capacity and Quality of Service Manual, Third Edition*. Ed. by Inc. Kittelson Associates and Parsons Brinckerhoff. Transportation Research Board, 2013. DOI: 10.17226/24766.

[41] Jason Brownlee. *How to Calculate the KL Divergence for Machine Learning*. Machine Learning Mastery. Accessed: Sep. 05, 2023. 2019. URL: https://machinelearningmastery.com/divergence-between-probability-distributions/.

[42] W. Burghout, H.N. Koutsopoulos, and I. Andreasson. “A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation”. In: *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006. DOI: 10.1109/itsc.2006.1707369.

[43] Abi Carter. *More and more cars in Germany: Car density reaches new peak*. Accessed Mar. 13, 2024. 2022. URL: https://www.iamexpat.de/expat-info/german-expat-news/more-and-more-cars-germany-car-density-reaches-new-peak.

[44] Jordi Casas, Josep Perarnau, and Alex Torday. "The need to combine different traffic modelling levels for effectively tackling large-scale projects adding a hybrid meso/micro approach". In: *Procedia - Social and Behavioral Sciences* 20 (2011), pp. 251–262. DOI: 10.1016/j.sbspro.2011.08.031.

[45] Avishai (Avi) Ceder. "Urban mobility and public transport: future perspectives and review". In: *International Journal of Urban Sciences* 25.4 (2020), pp. 455–479. DOI: 10.1080/12265934.2020.1799846.

[46] Nurhan Cetin, Adrian Burri, and Kai Nagel. "A large-scale agent-based traffic microsimulation based on queue model". In: *Swiss Transport Research Conference (STRC), Monte Verita, CH*. 2003. URL: https://api.semanticscholar.org/CorpusID:17207386.

[47] David Charypar and Kai Nagel. "Generating complete all-day activity plans with genetic algorithms". In: *Transportation* 32.4 (2005), pp. 369–397. DOI: 10.1007/s11116-004-8287-y.

[48] Rongsheng Chen and Michael W. Levin. "Dynamic User Equilibrium of Mobility-on-Demand System with Linear Programming Rebalancing Strategy". In: *Transportation Research Record: Journal of the Transportation Research Board* 2673.1 (2019), pp. 447–459. DOI: 10.1177/0361198118821629.

[49] T. Donna Chen, Kara M. Kockelman, and Josiah P. Hanna. "Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions". In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 243–254. DOI: https://doi.org/10.1016/j.tra.2016.08.020.

[50] Evelyn Cheng. *China's Nio to expand battery swap services to gain an edge on EV infrastructure*. CNBC. Retrieved 05/05/2024. 2024. URL: https://www.cnbc.com/2024/04/05/chinas-nio-to-expand-battery-swap-services-to-gain-ev-infra-edge-.html.

[51] Sandro Chiappone et al. “Traffic simulation models calibration using speed–density relationship: An automated procedure based on genetic algorithm”. In: *Expert Systems with Applications* 44 (2015), pp. 147–155. DOI: https://doi.org/10.1016/j.eswa.2015.09.024.

[52] Sachin Chugh et al. “Development of Delhi Driving Cycle: A Tool for Realistic Assessment of Exhaust Emissions from Passenger Cars in Delhi”. In: *SAE Technical Paper Series*. SAE International, 2012. DOI: 10.4271/2012-01-0877.

[53] Biagio Ciuffo, Vincenzo Punzo, and Vincenzo Torrieri. “Comparison of Simulation-Based and Model-Based Calibrations of Traffic-Flow Microsimulation Models”. In: *Transportation Research Record* 2088.1 (2008), pp. 36–44. DOI: 10.3141/2088-05.

[54] Rutger Claes and Tom Holvoet. “Multi-model traffic microsimulations”. In: 2010, pp. 1113–1123. DOI: 10.1109/WSC.2009.5429657.

[55] Carlos Cobos et al. “A Multi-Objective Approach for the Calibration of Microscopic Traffic Flow Simulation Models”. In: *IEEE Access* 8 (2020), pp. 103124–103140. DOI: 10.1109/ACCESS.2020.2999081.

[56] William G Cochran. *Sampling techniques*. John Wiley & Sons, 1977.

[57] Tom Cohen and Clémence Cavoli. “Automated vehicles: exploring possible consequences of government (non)intervention for congestion and accessibility”. In: *Transport Reviews* 39.1 (2018), pp. 129–151. DOI: 10.1080/01441647.2018.1524401.

[58] W. Daamen, C. Buisson, and S.P. Hoogendoorn, eds. *Traffic Simulation and Data*. Taylor & Francis Ltd, 2017. 262 pp. URL: https://www.ebook.de/de/product/28981401/traffic_simulation_and_data.html.

[59] Erwin Danneels. “Disruptive Technology Reconsidered: A Critique and Research Agenda”. In: *Journal of Product Innovation Management* 21.4 (2004), pp. 246–258. DOI: 10.1111/j.0737-6782.2004.00076.x.

[60] G. B. Dantzig and J. H. Ramser. “The Truck Dispatching Problem”. In: *Management Science* 6.1 (1959), pp. 80–91. DOI: 10.1287/mnsc.6.1.80.

[61] Gerard Debreu. “A Social Equilibrium Existence Theorem”. In: *Proceedings of the National Academy of Sciences* 38.10 (1952), pp. 886–893. DOI: 10.1073/pnas.38.10.886.

[62] Narsingh Deo. *Graph Theory with Applications to Engineering and Computer Science*. Dover Publications, Incorporated, 2016, p. 496.

[63] Reinhard Diestel. *Graph Theory*. Electronic library of mathematics. Graduate Texts in Mathematics. Springer, 2006, p. 415.

[64] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *NUMERISCHE MATHEMATIK* 1.1 (1959), pp. 269–271.

[65] Kavya P. Divakarla, Ali Emadi, and Saiedeh N. Razavi. "Journey Mapping—A New Approach for Defining Automotive Drive Cycles". In: *IEEE Transactions on Industry Applications* 52.6 (2016), pp. 5121–5129. DOI: 10.1109/tia.2016.2593694.

[66] Christoph Dobler. "Travel behaviour modelling for scenarios with exceptional events: Methods and implementations". en. PhD thesis. 2013. DOI: 10.3929/ETHZ-A-010019952.

[67] Gabriel Domingues. "Modeling, Optimization and Analysis of Electromobility Systems". English. PhD thesis. Department of Biomedical Engineering, Lund university, 2018. URL: https://api.semanticscholar.org/CorpusID:169117291.

[68] R. Dowling et al. "Guidelines for Calibration of Microsimulation Models: Framework and Applications". In: *Transportation Research Record* 1876 (2004), pp. 1–9. DOI: https://doi.org/10.3141/1876-01.

[69] Adam Duran and Matthew Earleywine. "GPS Data Filtration Method for Drive Cycle Analysis Applications". In: *SAE Technical Paper Series*. SAE International, 2012. DOI: 10.4271/2012-01-0743.

[70] United Nations Department of Economic and Social Affairs. "World Population Prospects 2022: Summary of Results". In: *UN DESA/POP/2022/TR/NO. 3., New York: United Nations* (2022).

[71] Jakob Erdmann. "SUMO's Lane-Changing Model". In: *Modeling Mobility with Open Data*. Springer International Publishing, 2015, pp. 105–123. DOI: 10.1007/978-3-319-15024-6_7.

[72] Daniel J. Fagnant and Kara M. Kockelman. "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas". In: *Transportation* 45.1 (2018), pp. 143–158. DOI: 10.1007/s11116-016-9729-z.

[73] Sharon Feigon and Colin Murphy. *Broadening Understanding of the Interplay Between Public Transit, Shared Mobility, and Personal Automobiles*. Transportation Research Board, 2018. DOI: 10.17226/24996.

[74] Joachim Becker und Felix Reek. *Das Märchen von der Jugend ohne Auto*. Süddeutsche Zeitung. Accessed Mar. 13, 2024. 2023. URL: https://www.sueddeutsche.de/auto/autofahren-studie-jugend-1.6317078?reduced=true.

[75] M. Fellendorf and Peter Vortisch. "Validation of the Microscopic Traffic Flow Model VISSIM in Different Real-World Situations". In: *Transportation Research Board (TRB). 80nd Annual Meeting, Washington, DC, 07.-11.01.2001*. 80th Annual Meeting Transportation Research Board. TRB 2001 (Washington, DC, USA, Jan. 7–11, 2001). 2001. URL: https://www.researchgate.net/publication/268521190.

[76] Sean Fleming. *China joins list of nations banning the sale of old-style fossil-fuelled vehicles*. World Economic Forum. Retrieved 06/05/2024. 2020. URL: https://www.weforum.org/agenda/2020/11/china-bans-fossil-fuel-vehicles-electric/.

[77] Gunnar Flötteröd. "Cadyts - a free calibration tool for dynamic traffic simulations". In: *9th Swiss Transport Research Conference* (2009). URL: https://www.researchgate.net/publication/46788058.

[78] Gunnar Flötteröd. "Queueing Representation of Kinematic Waves". In: *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016, pp. 347–352. DOI: 10.5334/baw.50.

[79] Yun-Pang Flötteröd. "SUMO-Cadyts calibration with limited data quality". In: *SUMO User Conference 2017 - Towards Simulation for Autonomous Mobility*. 2017, pp. 155–162. URL: https://elib.dlr.de/113211/.

[80] Lester Randolph Ford. *Network flow theory*. Rand Corporation Santa Monica, CA, 1956.

[81] Marcus Foth. *Smart mobility alone is no substitute for strong policy leadership . The Conversation, pp. 1-5*. 2018. URL: https://eprints.qut.edu.au/122917/.

[82] A. Fotouhi and M. Montazeri-Gh. "Tehran driving cycle development using the k-means clustering method". In: *Scientia Iranica* 20.2 (2013), pp. 286–293. DOI: https://doi.org/10.1016/j.scient.2013.04.001.

[83] Fraunhofer Institute for Industrial Mathematics (ITWM). *Virtual Measurement Campaign*. Product Flyer. 2020. URL: https://www.itwm.fraunhofer.de/en/departments/mf/products-and-services/vmc/vmc-simulation.html.

[84] Rainer Gasper et al. "Simulation of Autonomous RoboShuttles in Shared Space". In: *EPiC Series in Engineering*. EasyChair, 2018. DOI: 10.29007/h58z.

[85] C. Gawron. "An Iterative Algorithm to Determine the Dynamic User Equilibrium in a Traffic Simulation Model". In: *International Journal of Modern Physics C* Vol. 09.No. 03 (1998), pp. 393–407. DOI: https://doi.org/10.1142/S0129183198000303.

[86] Qiuming Gong et al. "An Iterative Markov Chain Approach for Generating Vehicle Driving Cycles". In: *SAE International Journal of Engines* 4.1 (2011), pp. 1035–1045. DOI: 10.4271/2011-01-0880.

[87] Monica Grosso et al. "How will vehicle automation and electrification affect the automotive maintenance, repair sector?" In: *Transportation Research Interdisciplinary Perspectives* 12 (2021), p. 100495. DOI: 10.1016/j.trip.2021.100495.

[88] M. Gütlein, R. German, and A. Djanatliev. "Towards a Hybrid Co-Simulation Framework: HLA-Based Coupling of MATSim and SUMO". In: *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. 2018, pp. 1–9. DOI: 10.1109/DISTRA.2018.8601004.

[89] Chana J. Haboucha, Robert Ishaq, and Yoram Shiftan. "User preferences regarding autonomous vehicles". In: *Transportation Research Part C: Emerging Technologies* 78 (2017), pp. 37–49. DOI: 10.1016/j.trc.2017.01.010.

[90] Mustapha Harb et al. "Projecting travelers into a world of self-driving vehicles: estimating travel behavior implications via a naturalistic experiment". In: *Transportation* 45.6 (2018), pp. 1671–1685. DOI: 10.1007/s11116-018-9937-9.

[91] P. E. Hart, N. J. Nilsson, and B. Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107. DOI: 10.1109/TSSC.1968.300136.

[92] Jürgen Hedderich and Lothar Sachs. *Angewandte Statistik: Methodensammlung mit R*. 7.2.6 Kolmogoroff-Smirnoff Anpassungstest, S. 494–497, Springer Berlin Heidelberg, 2020. DOI: 10.1007/978-3-662-62294-0.

[93] Florian Herrmann et al. *BESCHÄFTIGUNG 2030 - Auswirkungen von Elektromobilität und Digitalisierung auf die Qualität und Quantität der Beschäftigung bei Volkswagen*. de. Fraunhofer Publica. 2020. DOI: 10.13140/RG.2.2.16571.44323.

[94] Mohamed Said El Hmam et al. "Towards an Hybrid Simulation Approach of Transportation Systems". In: *IFAC Proceedings Volumes* 37.19 (2004), pp. 75–80. DOI: 10.1016/s1474-6670(17)30662-6.

[95] Thomas Holdstock and Michael Bryant. "Electric drivetrain architecture optimisation for autonomous vehicles based on representative cycles". In: *17th Int. CTI Symp. Automotive Transmissions, Berlin*. 2018.

[96] Sebastian Hörl, Francesco Ciari, and Kay W. Axhausen. "Recent perspectives on the impact of autonomous vehicles". en. Zurich, 2016. DOI: 10.3929/ethz-b-000121359.

[97] Sebastian Hörl et al. *Induzierter Verkehr durch autonome Fahrzeuge. Eine Abschätzung*. de. Tech. rep. SNF and ETH Zürich, 2019. URL: https://www.research-collection.ethz.ch/handle/20.500.11850/346381.

[98] A. Horni, K. Nagel, and K. W. Axhausen, eds. *The multi-agent transport simulation MATSim.* Ubiquity Press, London, 2016. DOI: 10.5334/baw.

[99] Andreas Horni, David Charypar, and Kay W. Axhausen. "Variability in Transport Microsimulations Investigated With the Multi-Agent Transport Simulation MATSim". en. In: *Arbeitsberichte Verkehrs- und Raumplanung* (2011). DOI: 10.3929/ETHZ-A-006686329.

[100] Andreas Horni and Kai Nagel. "More About Configuring MATSim". In: *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016, pp. 35–44. DOI: 10.5334/baw.4.

[101] Andreas Horni, Kai Nagel, and Kay W. Axhausen. *MATSim User Guide*. Retrieved 09/05/2024. 2024. URL: https://www.matsim.org/docs/userguide/.

[102] Yunfei Hou et al. “Suitability of Synthetic Driving Profiles from Traffic Micro-Simulation for Real-World Energy Analysis: Preprint”. In: *National Renewable Energy Laboratory (NREL)* (2015). URL: https://www.nrel.gov/docs/fy16osti/64609.pdf.

[103] John Hourdakis, Panos G. Michalopoulos, and Jiji Kottommannil. “Practical Procedure for Calibrating Microscopic Traffic Simulation Models”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1852.1 (2003), pp. 130–139. DOI: 10.3141/1852-17.

[104] Wolfgang Hugemann and Markus Nickel. *Longitudinal and Lateral Accelerations in Normal Day Driving*. Tech. rep. Ingenieurbüro Morawski, Käckel und Hädrich, 2003. URL: https://api.semanticscholar.org/CorpusID:215742524.

[105] Irena Ištoka Otković, Tomaž Tollazzi, and Matjaž Šraml. “Calibration of microsimulation traffic model using neural network approach”. In: *Expert Systems with Applications* 40.15 (2013), pp. 5965–5974. DOI: https://doi.org/10.1016/j.eswa.2013.05.003.

[106] J3016. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE Standard, 2021. DOI: 10.4271/j3016_201609.

[107] Eric J. Johnson, Steven Bellman, and Gerald L. Lohse. “Cognitive Lock-In and the Power Law of Practice”. In: *Journal of Marketing* 67.2 (2003), pp. 62–75. DOI: 10.1509/jmkg.67.2.62.18615.

[108] Mahtab Joueiai, Hans van Lint, and Serge Hoogendoorn. “Generic solutions for consistency problems in multi-scale traffic flow models - Analysis and preliminary results”. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013. DOI: 10.1109/itsc.2013.6728250.

[109] Ihab Kaddoura, Gregor Leich, and Andreas Neumann andKai Nagel. “A simulation-based heuristic for the improvementof ride-hailing services”. VSP Working Paper, Submitted to TRB2021. 2020.

[110] Ihab Kaddoura, Gregor Leich, and Kai Nagel. “The impact of pricing and service area design on the modal shift towards demand responsive transit”. In: *Procedia Computer Science* 170 (2020), pp. 807–812. DOI: https://doi.org/10.1016/j.procs.2020.03.152.

[111] Ihab Kaddoura and Tilmann Schlenther. “The impact of trip density on the fleet size and pooling rate of ride-hailing services: A simulation study”. In: 184 (2021), pp. 674–679. DOI: 10.1016/j.procs.2021.03.084.

[112] Sanghpriya H. Kamble, Tom V. Mathew, and G.K. Sharma. “Development of real-world driving cycle: Case study of Pune, India”. In: *Transportation Research Part D: Transport and Environment* 14.2 (2009), pp. 132–140. DOI: 10.1016/j.trd.2008.11.008.

[113] Ioannis Karakikes, Matthias Spangler, and Martin Margreiter. “Designing a Vissim-Model for a motorway network with systematic calibration on the basis of travel time measurements”. In: *Transportation Research Procedia* 24 (2017), pp. 171–179. DOI: https://doi.org/10.1016/j.trpro.2017.05.086.

[114] Juffrizal Karjanto et al. “Simulating autonomous driving styles: Accelerations for three road profiles”. In: *MATEC Web of Conferences* 90 (2016). Ed. by S.A. Che Ghani, W.A. Wan Hamzah, and A. Alias, p. 01005. DOI: 10.1051/matecconf/20179001005.

[115] Seung-Jun Kim, Wonho Kim, and L. R. Rilett. “Calibration of Microsimulation Models Using Nonparametric Statistical Techniques”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1935.1 (2005), pp. 111–119. DOI: 10.1177/0361198105193500113.

[116] Lucas Koch et al. “Accurate physics-based modeling of electric vehicle energy consumption in the SUMO traffic microsimulator”. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021. DOI: 10.1109/itsc48978.2021.9564463.

[117] Murali Krishna Kondaru et al. “Generating a Real World Drive Cycle–A Statistical Approach”. In: *SAE Technical Paper Series*. SAE International, 2018. DOI: 10.4271/2018-01-0325.

[118] Arun Koyadan Koroth, Grzegorz Mazurek, and Przemysław Pater. “Disruptive Innovation in Automotive Retailing”. In: *Journal of Management and Business Administration. Central Europe* 27.1 (2019), pp. 44–59. DOI: 10.7206/jmba.ce.2450-7814.238.

[119] Dexter C. Kozen. “Depth-First and Breadth-First Search”. In: *The Design and Analysis of Algorithms*. New York, NY: Springer New York, 1992, pp. 19–24. DOI: 10.1007/978-1-4612-4400-4_4.

[120] Matthias Kracht et al. *PAVE- Potentiale Automatisierter Verkehrssysteme*. Forschungsprojekt gefördert durch das Bundesministerium für Verkehr und digitale Infrastruktur (FKZ: 16AVF2147A). 2021. URL: https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwp/2021/21-30/KrachtEtAl2021PAVE.pdf.

[121] S. Krauss. "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics". Dissertation. Mathematisch-Naturwissenschaftliche Fakultät, Universität Köln and German Aerospace Center (DLR). LIDO-Berichtsjahr 1999. PhD thesis. 1998. URL: https://elib.dlr.de/8380/.

[122] S. Krauss, P. Wagner, and C. Gawron. "Metastable states in a microscopic model of traffic flow". In: *Physical Review E* 5 (1997). LIDO-Berichtsjahr 1997, pp. 5597–5602. URL: https://elib.dlr.de/30685/.

[123] Nerijus Kudarauskas. "Analysis of emergency braking of a vehicle". In: *Transport* 22 (2007). DOI: 10.1080/16484142.2007.9638118.

[124] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694.

[125] Solomon Kullback. *Information Theory and Statistics*. John Wiley & Sons, 1959.

[126] Tamás Kurczveil, Pablo Alvarez López, and Eckehard Schnieder. "Implementation of an Energy Model and a Charging Infrastructure in SUMO". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2014, pp. 33–43. DOI: 10.1007/978-3-662-45079-6_3.

[127] Guido Lauriks et al. "UIC Comfort Tests: Investigation of Ride Comfort and Comfort Disturbance on Transition and Circular Curves". In: 2003. URL: https://api.semanticscholar.org/CorpusID:106681040.

[128] Crystal Legacy et al. "Planning the driverless city". In: *Transport Reviews* 39.1 (2018), pp. 84–102. DOI: 10.1080/01441647.2018.1466835.

[129] M. J. Lighthill and G.B. Whitham. "On kinematic waves. II. A theory of traffic flow on long crowded roads." In: *Proceedings of the Royal Society A* (1955).

[130] R. Lindgren. “Analysis of flow features in queued traffic on a German freeway”. PhD thesis. Portland State University, 2005. URL: https://www.researchgate.net/publication/251637478.

[131] Jun Liu, Kara Kockelman, and Aqshems Nichols. “Anticipating the emissions impacts of smoother driving by connected and autonomous vehicles, using the MOVES model”. In: *The 96th Annual Meeting of the Transportation Research Board*. 2017. URL: https://www.researchgate.net/publication/319243433.

[132] Jun Liu et al. “Tracking a System of Shared Autonomous Vehicles across the Austin, Texas Network using Agent-Based Simulation”. In: *Transportation* (2017). DOI: 10.1007/s11116-017-9811-1.

[133] Carlos Llorca and Rolf Moeckel. “Effects of scaling down the population for agent-based traffic simulations”. In: *Procedia Computer Science* 151 (2019), pp. 782–787. DOI: 10.1016/j.procs.2019.04.106.

[134] Benjamin Loeb, Kara M. Kockelman, and Jun Liu. “Shared autonomous electric vehicle (SAEV) operations across the Austin, Texas network with charging infrastructure decisions”. In: *Transportation Research Part C: Emerging Technologies* 89 (2018), pp. 222–233. DOI: https://doi.org/10.1016/j.trc.2018.01.019.

[135] Pablo Alvarez Lopez et al. “Microscopic Traffic Simulation using SUMO”. In: *The 21st IEEE IC on Intelligent Transportation Systems*. IEEE, 2018, pp. 2575–2582. URL: https://elib.dlr.de/127994/.

[136] Qiong Lu and Tamás Tettamanti. “Impacts of Autonomous Vehicles on the Urban Fundamental Diagram”. In: *Road and Rail Infrastructure V*. University of Zagreb Faculty of Civil Engineering, 2018. DOI: 10.5592/co/cetra.2018.714.

[137] Michal Maciejewski. “Dynamic Transport Services”. In: *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016, pp. 145–152. DOI: 10.5334/baw.23.

[138] Michał Maciejewski. *Benchmarking minimum passenger waiting time in online taxi dispatching with exact offline optimization methods*. en. 2014. DOI: 10.14279/depositonce-6661.

[139] Michal Maciejewski and Joschka Bischoff. “Large-scale Microscopic Simulation of Taxi Services”. In: *Procedia Computer Science* 52 (2015), pp. 358–364. DOI: 10.1016/j.procs.2015.05.107.

[140] Michal Maciejewski, Joschka Bischoff, and Kai Nagel. "An Assignment-Based Approach to Efficient Real-Time City-Scale Taxi Dispatching". In: *IEEE Intelligent Systems* 31.1 (2016), pp. 68–77. DOI: 10.1109/mis.2016.2.

[141] Michał Maciejewski and Kai Nagel. "Towards Multi-Agent Simulation of the Dynamic Vehicle Routing Problem in MATSim". In: *Parallel Processing and Applied Mathematics*. Springer Berlin Heidelberg, 2012, pp. 551–560. DOI: https://doi.org/10.1007/978-3-642-31500-8_57.

[142] Michal Maciejewski et al. "Towards a Testbed for Dynamic Vehicle Routing Algorithms". In: *Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems*. Springer International Publishing, 2017, pp. 69–79. DOI: https://doi.org/10.1007/978-3-319-60285-1_6.

[143] Ricardo Maia et al. "Electric vehicle simulator for energy consumption studies in electric mobility systems". In: *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*. IEEE, 2011. DOI: 10.1109/fists.2011.5973655.

[144] Sebastian Maier. *Digital eDrive - Abschlussbericht*. Tech. rep. Forschungsinstitut fürKraftfahrwesen undFahrzeugmotorenStuttgart (FKFS), 2019.

[145] Nicolai Mallig, Martin Kagerbauer, and Peter Vortisch. "mobiTopp – A Modular Agent-based Travel Demand Modelling Framework". In: *Procedia Computer Science* 19 (2013), pp. 854–859. DOI: https://doi.org/10.1016/j.procs.2013.06.114.

[146] P. Marchand and L. Marmet. "Binomial smoothing filter: A way to avoid some pitfalls of least-squares polynomial smoothing". In: *Review of Scientific Instruments* 54.8 (1983), pp. 1034–1041. DOI: 10.1063/1.1137498.

[147] MATSim. *Java-related information*. retrieved 26/08/2024. URL: https://matsim.org/docs/devguide/java.

[148] Sandeep Menneni, Carlos Sun, and Peter Vortisch. "Microsimulation Calibration Using Speed-Flow Relationships". In: *Transportation Research Record* 2088 (2008), pp. 1–9. DOI: 10.3141/2088-01.

[149] Tim Mercy, Ruben Van Parys, and Goele Pipeleers. “Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment”. In: *IEEE Transactions on Control Systems Technology* 26.6 (2018), pp. 2182–2189. DOI: 10.1109/tcst.2017.2739706.

[150] Dimitris Milakis et al. “Development and transport implications of automated vehicles in the Netherlands: scenarios for 2030 and 2050”. en. In: *European Journal of Transport and Infrastructure Research* (2017), Vol 17 No 1 (2017). DOI: 10.18757/EJTIR.2017.17.1.3180.

[151] Vicente Milanés and Steven E. Shladover. “Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data”. In: *Transportation Research Part C: Emerging Technologies* 48 (2014), pp. 285–300. DOI: 10.1016/j.trc.2014.09.001.

[152] Márk Miskolczi et al. “Urban mobility scenarios until the 2030s”. In: *Sustainable Cities and Society* 72 (2021), p. 103029. DOI: 10.1016/j.scs.2021.103029.

[153] Sabine Moeller and Kristina Wittkowski. “The burdens of ownership: reasons for preferring renting”. In: *Managing Service Quality: An International Journal* 20.2 (2010), pp. 176–191. DOI: 10.1108/09604521011027598.

[154] Felix Mogge et al. *Global Automotive Supplier Study 2023 - Opportunities for growth amid the challenge of change*. Tech. rep. Roland Berger GmbH and Lazard & Co. GmbH, 2023.

[155] A.T. Moreno et al. “Shared Autonomous Vehicles Effect on Vehicle-Km Traveled and Average Trip Duration”. In: *Journal of Advanced Transportation* 2018 (2018). DOI: 10.1155/2018/8969353.

[156] Ted Morris et al. *A Comprehensive System for Assessing Truck Parking Availability*. Tech. rep. Center for Transportation Studies, University of Minnesota, 2017. URL: https://conservancy.umn.edu/items/360b6b8b-3b1f-47da-8e29-abfd7c63212d.

[157] Sebastian A Müller et al. “Using mobile phone data for epidemiological simulations of lockdowns: government interventions, behavioral changes, and resulting changes of reinfections”. medRxiv 2020.07.22.20160093. 2020. DOI: 10.1101/2020.07.22.20160093.

[158] Sebastian A. Müller et al. “A realistic agent-based simulation model for COVID-19 based on a traffic simulation and mobile phone data”. arXiv:2011.11453. 2020. DOI: 10.48550/ARXIV.2011.11453.

[159] Sebastian A. Müller et al. “Predicting the effects of COVID-19 related interventions in urban settings by combining activity-based modelling, agent-based simulation, and mobile phone data”. In: *PLOS ONE* 16.10 (2021), e0259037. DOI: 10.1371/journal.pone.0259037.

[160] Sebastian Alexander Müller et al. “Mobility traces and spreading of COVID-19”. en. medRxiv 2020.03.27.20045302. 2020. DOI: 10.14279/DEPOSITONCE-9835.

[161] Kyle B. Murray and Gerald Häubl. “Explaining Cognitive Lock-In: The Role of Skill-Based Habits of Use in Consumer Choice”. In: *Journal of Consumer Research* 34.1 (2007), pp. 77–88. DOI: 10.1086/513048.

[162] Kai Nagel et al. *Simulation-based analysis of the impacts of effects of autonomous vehicles on urban traffic*. Fachgebiet Verkehrssystemplanung und Verkehrstelematik, Institut für Land- und Seeverkehr, TU Berlin, VSP Working Paper 19-06. 2019. URL: http://www.vsp.tu-berlin.de/publications/vspwp.

[163] Gopindra Sivakumar Nair et al. “An application of a rank ordered probit modeling approach to understanding level of interest in autonomous vehicles”. In: *Transportation* 45.6 (2018), pp. 1623–1637. DOI: 10.1007/s11116-018-9945-9.

[164] John Nash. “Non-Cooperative Games”. In: *The Annals of Mathematics* 54.2 (1951), p. 286. DOI: 10.2307/1969529.

[165] John F. Nash. “Equilibrium points in n-person games”. In: *Proceedings of the National Academy of Sciences* 36.1 (1950), pp. 48–49. DOI: 10.1073/pnas.36.1.48.

[166] Andreas Neumann and Michael Balmer. “Mobility Pattern Recognition (MPR) und Anonymisierung von Mobilfunkdaten”. Whitepaper. 2020. URL: https://senozon.com/wp-content/uploads/Whitepaper_MPR_Senozon_DE.pdf.

[167] G.F. Newell. “A simplified theory of kinematic waves in highway traffic, part I: General theory”. In: *Transportation Research Part B: Methodological* 27.4 (1993), pp. 281–287. DOI: 10.1016/0191-2615(93)90038-c.

[168] Johannes Nguyen et al. “An overview of agent-based traffic simulators”. In: *Transportation Research Interdisciplinary Perspectives* 12 (2021), p. 100486. DOI: 10.1016/j.trip.2021.100486.

[169] T.W. Nicolai. “Using MATSim as a travel model plug-in to UrbanSim”. Technische Universität Berlin. 2012.

[170] Claudia Nobis and Tobias Kuhnimhof. *Mobilität in Deutschland - MiD Ergebnisbericht.* Tech. rep. Studie von infas, DLR und IVT im Auftrag des Bundesministers für Verkehr und digitale Infrastruktur (FE-Nr. 70.904/15). Bonn, Berlin, 2018. URL: https://www.mobilitaet-in-deutschland.de/.

[171] Peter Nyberg, Erik Frisk, and Lars Nielsen. “Using Real-World Driving Databases to Generate Driving Cycles With Equivalence Properties”. In: *IEEE Transactions on Vehicular Technology* 65.6 (2016), pp. 4095–4105. DOI: 10.1109/tvt.2015.2502069.

[172] Ohde. “Statistical analysis of real-world urban driving cycles for modelling energy consumption of electric vehicles”. In: *Journal of Mechanical and Transport Engineering* (2016). DOI: 10.21008/J.2449-920X.2016.68.2.03.

[173] Hector Ortega-Arranz, Diego R. Llanos, and Arturo Gonzalez-Escribano. *The Shortest-Path Problem: Analysis and Comparison of Methods*. Springer International Publishing, 2015. DOI: 10.1007/978-3-031-02574-7.

[174] Vasileia Papathanasopoulou, Ioulia Markou, and Constantinos Antoniou. “Online calibration for microscopic traffic simulation and dynamic multi-step prediction of traffic speed”. In: *Transportation Research Part C: Emerging Technologies* 68 (2016), pp. 144–159. DOI: https://doi.org/10.1016/j.trc.2016.04.006.

[175] Byungkyu (Brian) Park and J. D. Schneeberger. “Microscopic Simulation Model Calibration and Validation: Case Study of VISSIM Simulation Model for a Coordinated Actuated Signal System”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1856.1 (2003), pp. 185–192. DOI: 10.3141/1856-20.

[176] European Parliament. *Fit for 55: zero CO2 emissions for new cars and vans in 2035*. European Parliament, Press release. Retrieved 06/05/2024. 2023. URL: https://www.europarl.europa.eu/news/en/press-room/20230210IPR74715/fit-for-55-zero-co2-emissions-for-new-cars-and-vans-in-2035.

[177] Mads Paulsen, Thomas Kjær Rasmussen, and Otto Anker Nielsen. "Output variability caused by random seeds in a multi-agent transport simulation model". In: *Procedia Computer Science* 130 (2018), pp. 850–857. DOI: 10.1016/j.procs.2018.04.078.

[178] Constantinos-Vasilios Priporas, Nikolaos Stylos, and Anestis K. Fotiadis. "Generation Z consumers' expectations of interactions in smart retailing: A future agenda". In: *Computers in Human Behavior* 77 (2017), pp. 374–381. DOI: 10.1016/j.chb.2017.01.058.

[179] Harilaos N. Psaraftis. "Dynamic Vehicle Routing Problems". English. In: *Vehicle Routing: Methods and Studies*. North-Holland, 1988, pp. 223–248.

[180] Christian Rakow et al. "Investigation of the system-wide effects of intelligent infrastructure concepts with microscopic and mesoscopic traffic simulation". In: *27th ITS World Congress, Hamburg, Germany*. 2021. URL: https://elib.dlr.de/144810/.

[181] *Research on the Impacts of Connected and Autonomous Vehicles (CAVs) on Traffic Flow. Stage 2: Traffic Modelling and Analysis*. Tech. rep. Atkins Ltd, Department of Transport, 2016.

[182] Paul I. Richards. "Shock Waves on the Highway". In: *Operations Research* 4.1 (1956), pp. 42–51. DOI: 10.1287/opre.4.1.42.

[183] Gerald Richter et al. "Anticipating Automated Vehicle Presence and the Effects on Interactions with Conventional Traffic and Infrastructure". In: *EPiC Series in Computing*. EasyChair. DOI: 10.29007/s6m7.

[184] Horst Rinne. *Taschenbuch der Statistik*. Harri Deutsch, 1997.

[185] Thamara Vieira da Rocha et al. "Are vehicle trajectories simulated by dynamic traffic models relevant for estimating fuel consumption?" In: *Transportation Research Part D: Transport and Environment* 24 (2013), pp. 17–26. DOI: 10.1016/j.trd.2013.03.012.

[186] Emad Roshandel et al. “Losses in Efficiency Maps of Electric Vehicles: An Overview”. In: *Energies* 14.22 (2021), p. 7805. DOI: 10.3390/en14227805.

[187] Insaf Sagaama et al. “Evaluation of the Energy Consumption Model Performance for Electric Vehicles in SUMO”. In: *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2019. DOI: 10.1109/ds-rt47707.2019.8958704.

[188] Insaf Sagaama et al. “Improving The Accuracy of The Energy Consumption Model for Electric Vehicle in SUMO Considering The Ambient Temperature Effects”. In: *2018 IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*. IEEE, 2018. DOI: 10.23919/pemwn.2018.8548928.

[189] Abraham. Savitzky and M. J. E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures.” In: *Analytical Chemistry* 36.8 (1964), pp. 1627–1639. DOI: 10.1021/ac60214a047.

[190] Lothar Schindele et al. “Future e-mobility and the change in system requirements”. In: *19. Internationales Stuttgarter Symposium*. Springer Fachmedien Wiesbaden, 2019, pp. 241–251. DOI: 10.1007/978-3-658-25939-6_21.

[191] Martin Schoenhof and Dirk Helbing. “Empirical Features of Congested Traffic States and Their Implications for Traffic Modeling”. In: *Transportation Science* 41 (2004). DOI: 10.1287/trsc.1070.0192.

[192] Markus Schönfeld, Uli Baumann, and Luca Leicht. *CCS Adieu: Fast alle Hersteller wechseln zu Tesla-Ladetechnik*. Retrieved 05/05/2024. 2024. URL: https://www.auto-motor-und-sport.de/verkehr/eauto-hersteller-wechseln-ladestecker-ccs-auf-tesla-standard-nacs/.

[193] Ihab El-Shawarby et al. “Evaluation of Driver Deceleration Behavior at Signalized Intersections”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2018.1 (2007), pp. 29–35. DOI: 10.3141/2018-05.

[194] Shuming Shi et al. “Research on Markov property analysis of driving cycles and its application”. In: *Transportation Research Part D: Transport and Environment* 47 (2016), pp. 171–181. DOI: 10.1016/j.trd.2016.05.013.

[195] Fred Shone and Kasia Kozlowska. “Dealing with uncertainty: stability of agent based model outputs”. In: *European Transport Conference*. 2019. URL: https://trid.trb.org/View/1768490.

[196] Óscar Silva et al. “Environmental impacts of autonomous vehicles: A review of the scientific literature”. In: *Science of The Total Environment* 830 (2022), p. 154615. DOI: 10.1016/j.scitotenv.2022.154615.

[197] Patrice M. Simon, Joerg Esser, and Kai Nagel. “Simple Queuing Model Applied to the City of Portland”. In: *International Journal of Modern Physics C* 10.05 (1999), pp. 941–960. DOI: 10.1142/s0129183199000747.

[198] C Sommer, R German, and F Dressler. “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis”. In: *IEEE Transactions on Mobile Computing* 10.1 (2011), pp. 3–15. DOI: 10.1109/tmc.2010.133.

[199] Guohua Song, Lei Yu, and Yanhong Zhang. “Applicability of Traffic Microsimulation Models in Vehicle Emissions Estimates”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2270.1 (2012), pp. 132–141. DOI: 10.3141/2270-16.

[200] Edward R. Straub and Kristin E. Schaefer. “It takes two to Tango: Automated vehicles and human beings do the dance of driving – Four social considerations for policy”. In: *Transportation Research Part A: Policy and Practice* 122 (2019), pp. 173–183. DOI: 10.1016/j.tra.2018.03.005.

[201] Lars Svensson and Jenny Casey Eriksson. “Tuning for Ride Quality in Autonomous Vehicle : Application to Linear Quadratic Path Planning Algorithm”. 2015. URL: https://api.semanticscholar.org/CorpusID:109973025.

[202] Shir Tavor and Tal Raviv. “Anticipatory rebalancing of RoboTaxi systems”. In: *Transportation Research Part C: Emerging Technologies* 153 (2023), p. 104196. DOI: 10.1016/j.trc.2023.104196.

[203] Igor Haman Tchappi et al. “Towards an Multilevel Agent-based Model for Traffic Simulation”. In: *Procedia Computer Science* 109 (2017), pp. 887–892. DOI: https://doi.org/10.1016/j.procs.2017.05.416.

[204] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. “Congested Traffic States in Empirical Observations and Microscopic Simulations”. In: *Physical Review E* 62 (2000), pp. 1805–1824. DOI: 10.1103/PhysRevE.62.1805.

[205] Martin Treiber and Arne Kesting. *Traffic Flow Dynamics*. Springer Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-32460-4.

[206] Martin Treiber and Arne Kesting. “Validation of traffic flow models with respect to the spatiotemporal evolution of congested traffic patterns”. In: *Transportation Research Part C: Emerging Technologies* 21.1 (2011), pp. 31–41. DOI: 10.1016/j.trc.2011.09.002.

[207] Henriette Triebke, Markus Kromer, and Peter Vortisch. “Bridging the Gap Between Mesoscopic Transport Planning and Microscopic Traffic Simulation: An Analytical and Numerical Analysis of Traffic Dynamics”. In: *Transportation Research Record: Journal of the Transportation Research Board* (2022), p. 036119812211282. DOI: 10.1177/03611981221128284.

[208] Henriette Triebke, Markus Kromer, and Peter Vortisch. “Calibrating spatio-temporal network states in microscopic traffic simulation on a global level”. In: *SUMO User Conference*. 2021.

[209] Henriette Triebke, Markus Kromer, and Peter Vortisch. “Pre-study and insights to a sequential MATSim-SUMO tool-coupling to deduce 24h driving profiles for SAEVs”. In: *SUMO Conference Proceedings* 1 (2022), pp. 93–112. DOI: 10.52825/scp.v1i.103.

[210] Monica Tutuianu et al. “Development of the World-wide harmonized Light duty Test Cycle (WLTC) and a possible pathway for its introduction in the European legislation”. In: *Transportation Research Part D: Transport and Environment* 40 (2015), pp. 61–75. DOI: 10.1016/j.trd.2015.07.011.

[211] Department of Economic United Nations and Social Affairs. “World Urbanization Prospects: The 2018 Revision.” In: *(ST/ESA/SER.A/420) New York: United Nations.* (2019).

[212] Gregory C. Unruh. “Understanding carbon lock-in”. In: *Energy Policy* 28.12 (2000), pp. 817–830. DOI: 10.1016/s0301-4215(00)00070-7.

[213] Aso Validi, Walter Morales-Alvarez, and Cristina Olaverri-Monreal. "Analysis of the Battery Energy Estimation Model in SUMO Compared with Actual Analysis of Battery Energy Consumption". In: *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2021. DOI: 10.23919/cisti52073.2021.9476579.

[214] András Varga. "Discrete event simulation system". In: *Proc. of the European Simulation Multiconference (ESM'2001)*. Vol. 17. 2001.

[215] Bill Visnic. *Europe steps back from 2035 ICE ban*. SAE International. Retrieved 06/05/2024. 2023. URL: https://www.sae.org/news/2023/03/european-ice-ban.

[216] Volkswagen Group. *In „China-Speed" zum smarten Auto: Volkswagen entwickelt mit XPENG digitale Hochleistungs-E/E Architektur für Elektro-Fahrzeuge in China*. Retrieved 03/08/2024. 2024. URL: https://www.volkswagen-group.com/de/artikel/in-china-speed-zum-smarten-auto-volkswagen-entwickelt-mit-xpeng-digitale-hochleistungs-ee-architektur-fuer-elektro-fahrzeuge-in-china-18335.

[217] Volkswagen Group. *Volkswagen AG beschließt Investition von Volkswagen in Rivian Automotive, Inc., in Höhe von zunächst 1 Mrd. USD und beabsichtigt Gründung eines Joint Venture*. Retrieved 03/08/2024. 2024. URL: https://www.volkswagen-group.com/de/ad-hoc/volkswagen-ag-beschliesst-investition-von-volkswagen-in-rivian-automotive-inc-in-hoehe-von-zunaechst-1-mrd-usd-und-beabsichtigt-gruendung-eines-joint-venture-18468.

[218] Rashid A. Waraich and Joschka Bischoff. "The Multi-Agent Transport Simulation MATSim". In: ed. by A.Horni, K. Nagel, and K. W. Axhausen. London: Ubiquity Press, 2016. Chap. Electric Vehicles, pp. 93–96.

[219] Waraich, Rashid A. "Agent-based simulation of electric vehicles: Design and implementation of a framework". en. PhD thesis. 2013. DOI: 10.3929/ETHZ-A-010111112.

[220] Lin Xiao, Meng Wang, and Bart van Arem. "Realistic Car-Following Models for Microscopic Simulation of Adaptive and Cooperative Adaptive Cruise Control Vehicles". In: *Transportation Research Record: Journal of the Transportation Research Board* 2623.1 (2017), pp. 1–9. DOI: 10.3141/2623-01.

[221] Lin Xiao et al. “Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks”. In: *Transportation Research Part C: Emerging Technologies* 96 (2018), pp. 380–397. DOI: 10.1016/j.trc.2018.10.008.

[222] Zhang Xiao, Zhao Dui-Jia, and Shen Jun-Min. “A Synthesis of Methodologies and Practices for Developing Driving Cycles”. In: *Energy Procedia* 16 (2012), pp. 1868–1873. DOI: 10.1016/j.egypro.2012.01.286.

[223] Wenda Xu et al. “A real-time motion planner with trajectory optimization for autonomous vehicles”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012. DOI: 10.1109/icra.2012.6225063.

[224] Isaak Yperman. “The Link Transmission Model for dynamic network loading”. PhD thesis. KU Leuven, 2007. URL: https://www.researchgate.net/publication/28360292.

[225] Peng Yuhui, Zhuang Yuan, and Yang Huibao. “Development of a representative driving cycle for urban buses based on the K-means cluster method”. In: *Cluster Computing* 22.S3 (2018), pp. 6871–6880. DOI: 10.1007/s10586-017-1673-y.

[226] Nidzamuddin Md. Yusof et al. “The Exploration of Autonomous Vehicle Driving Styles”. In: *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 2016. DOI: 10.1145/3003715.3005455.

[227] Juan Carlos Páez Zamora. “The “Avoid-Shift-Improve” Model: A Powerful Planning Tool for Transportation Schemes with Low GHG Emissions”. In: *MIPALCON 2014 Climate Change – A Global Challenge. Contribution of Infrastructure Planning*. 2014.

[228] Daniel Zehe et al. “Traffic Simulation Performance Optimization through Multi-Resolution Modeling of Road Segments”. In: *Proceedings of the 3rd ACM Conference on SIGSIM-Principles of Advanced Discrete Simulation - SIGSIM-PADS 15*. ACM Press, 2015. DOI: 10.1145/2769458.2769475.

[229] Hongcai Zhang et al. “Joint Fleet Sizing and Charging System Planning for Autonomous Electric Vehicles”. 2018. URL: https://arxiv.org/pdf/1811.00234.pdf.

[230] Xuan Zhao et al. “Development of a Representative EV Urban Driving Cycle Based on a k-Means and SVM Hybrid Clustering Algorithm”. In: *Journal of Advanced Transportation* 2018 (2018), pp. 1–18. DOI: 10.1155/2018/1890753.

[231] Zuduo Zheng et al. “Freeway Traffic Oscillations: Microscopic Analysis of Formations and Propagations using Wavelet Transform”. In: *Procedia - Social and Behavioral Sciences* 17 (2011), pp. 702–716. DOI: https://doi.org/10.1016/j.sbspro.2011.04.540.

[232] Benjamin A. Zielke, R. Bertini, and M. Treiber. “Empirical Measurement of Freeway Oscillation Characteristics”. In: *Transportation Research Record* 2088 (2008), pp. 57–67.

[233] D. Ziemke and K. Nagel. *Development of a fully synthetic and open scenario for agent-based transport simulations – The MATSim Open Berlin Scenario*. VSP Working Paper 17-12. URL. TU Berlin, Transport Systems Planning and Transport Telematics, 2017. URL: https://www.vsp.tu-berlin.de/menue/forschung/publications/bibtex_db/liste_jahr/?tx_sibibtex_pi1%5Bcontentelement%5D=tt_content%3A280673&tx_sibibtex_pi1%5BshowUid%5D=1900698&cHash=e434793921eb9d97c2bfc92ac8ad46a3.

[234] Dominik Ziemke, Ihab Kaddoura, and Kai Nagel. “The MATSim Open Berlin Scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data”. In: *Procedia Computer Science* 151 (2019), pp. 870–877. DOI: https://doi.org/10.1016/j.procs.2019.04.120.

[235] Dominik Ziemke, Kai Nagel, and Chandra Bhat. “Integrating CEMDAP and MATSIM to Increase the Transferability of Transport Demand Models”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2493.1 (2015), pp. 117–125. DOI: 10.3141/2493-13.

[236] Theresa Ziemke, Lucas N. Alegre, and Ana L.C. Bazzan. “Reinforcement learning vs. rule-based adaptive traffic signal control: A Fourier basis linear function approximation for traffic signal control”. In: *AI Communications* (2021), pp. 1–15. DOI: 10.3233/AIC-201580.

[237] Theresa Ziemke and Söhnke Braun. “Automated generation of traffic signals and lanes for MATSim based on OpenStreetMap”. In: *Procedia Computer Science* 184 (2021), pp. 745–752. DOI: 10.1016/j.procs.2021.03.093.

[238] Sarah Ellie Ziems et al. “Stochastic Variability in Microsimulation Modeling Results and Convergence of Corridor-Level Characteristics”. In: *Transportation Research Board 90th Annual Meeting*. 2011. URL: https://trid.trb.org/view/1093106.

[239] M. Zilske. “The Multi-Agent Transport Simulation MATSim”. In: ed. by A. Horni, K. Nagel, and K. W. (eds.) Axhausen. London: Ubiquity Press, 2016. Chap. How toWrite Your Own Extensions and Possibly Contribute Them to MATSim. Pp. 297–304.