Karlsruhe Institute of Technology

A Case for Adaptive Knowledge Discovery:
Concept Drift, Aggregated Measurements, and Integrating Domain Knowledge
Towards Novel Data-Efficient Learning Tasks under Uncertainty

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Béla Hraban Böhnke

Tag der mündlichen Prüfung:	16. April 2026
Erste Gutachterin:	Prof. Dr. Nadja Klein
Zweiter Gutachter:	Prof. Dr. Michael Granitzer

Foreword

The pursuit of knowledge stands as a core mission of science. Yet, in the age of data science, the foundational mechanisms of our most powerful tools—specifically *Machine Learning (ML)* and *Artificial Intelligence (AI)* algorithms—have a fundamental flaw that subtly contradicts this scientific spirit.

Having spent several years working with ML in practice, a critical weakness became apparent: once trained, these algorithms become **static** with respect to the data and knowledge they possess, not aware of their own boundaries. This contrasts sharply with the essence of intelligence, as famously captured by the Greek philosopher Socrates (as recounted by Plato), which lies in recognizing one's own limitations:

> "I was conscious that I knew practically nothing."
> — Plato, *Apology*, translated by Harold North Fowler

A truly intelligent agent understands its own gaps in knowledge, which is the necessary first step to initiate a query and seek new information. Current ML algorithms lack this self-awareness; they do not know what they do not know, and are therefore unable to actively inquire the specific data needed to complete their understanding. This inherent static nature has also been brought into focus by the rise of *Large Language Models (LLMs)*. While these models are masters of generating plausible-sounding text, they suffer from the same core deficiency. Their underlying weights are fixed after training, leading to the phenomenon of **hallucination**—the confident presentation of incorrect facts, not able to distinguish between true knowledge and mere statistical plausibility. Recent research, including studies from OpenAI itself [Kal+25], confirms this systemic issue, arguing that the reward structures in modern training pipelines incentivize "guessing over acknowledging uncertainty." As Richard P. Feynman once noted about the scientific method:

> "The scientist has a lot of experience with ignorance and doubt and uncertainty, and this experience is of very great importance, I think... we must recognize our ignorance and leave room for doubt."
> — Richard P. Feynman

If our advanced algorithms cannot recognize their own ignorance and doubt, their utility in tasks requiring real discovery is fundamentally limited.

When I began this research, my initial ambition was to work on the task of fully automated autonomous research—developing algorithms capable of forming hypotheses, identifying information gaps, and systematically executing a closed-loop research cycle. It quickly became clear that we are a long way from this "life-long" goal.

This dissertation, therefore, takes a more focused approach. It serves as **A Case for Adaptive Knowledge Discovery**, emphasizing the critical importance and practical

implementation of algorithms that can quantify and actively reduce their own uncertainty. The work highlights the pressing need for uncertainty-aware algorithms to move beyond static prediction and take the essential first step toward true intelligence by acknowledging what they do not know.

Acknowledgements

In the following, I want to acknowledge the people who guided me during my work and whom I have met along the way.

First, I thank Prof. Klemens Böhm, my doctoral supervisor. Sadly, due to a severe and unforeseen health crisis, he was unable to continue leading our chair. I thank him for his abundant feedback on writing, which enabled me to always improve my work and my skills. Next, I want to thank Edouard Fouché, my co-supervisor. I congratulate him on his successful transition to industry, which sadly led to his departure from the chair. While we did not always have the same opinions and sometimes lost our selfs in details, I know he always meant well and wanted to help – I thank him for that and his invaluable guidance and materials he provided. Which brings me to thanking my new doctoral supervisor, Nadja Klein. Even though, she took over the supervision of my thesis on short notice – she invested her valuable time to guide me, and I benefited from her deeper theoretical background. I wish I had met her earlier to have more time for deep diving discussions.

I am also thankful to all the students and HiWis I supervised and for the work they did. Further, I want to thank my coworkers for the exchanges. Not forgetting Barbara, Bettina, and Jennifer, our secretaries, it would not have been the same without their help.

I do acknowledge the DFG Research Training Group 2153: 'Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation', from which I obtained funding.

Last but not least, I thank my family, especially my wife, Christina for her support, and my son, Ilai H. Nox, who was born during my dissertation and quite literally turned my life upside down while providing a lot of joy. A child can teach you more about you and live then anyone else. Thank you Christina and thank you Ilai that I have you in my live.

Béla H. Böhnke, 19th February 2026, Karlsruhe.

Abstract

Knowledge Discovery (KD) – the process of extracting valid, novel, and actionable patterns from data – has a profound impact on modern science and industry. It facilitates critical advancements in fields such as materials science, medicine, and biology, as well as the optimization of complex industrial manufacturing processes.

However, the classic KD processes often assumes that data is readily available in a centralized warehouse. In reality, data collection is often an expensive and time-consuming endeavor, requiring significant material resources, labor, and energy. Such data acquisition costs becomes particularly challenging for KD when (1) the underlying data-generating process is non-stationary, i.e., exhibiting concept drift, and when (2) measurements are integrated or aggregated, obscuring the true quantity of interest. Here, the expensive data collection makes acquiring sufficient data for traditional solutions to these challenges often impractical. While established fields like Active Learning and Bayesian Optimization address data efficiency, existing literature often focuses on a few well-researched tasks, frequently missing the linkage between these building blocks and more diverse discovery tasks. The goal of this dissertation is to bridge this gap by providing a methodological case for **Adaptive Knowledge Discovery (AKD)** in new scenarios and for new tasks.

We first focus on the challenge of concept drift by introducing **Data Efficient Active Learning (DEAL)** a new approach for regression under drift. DEAL explicitly models drift as a stochastic process estimating the increase in uncertainty across data regions. We show that this allows for model recalibration only when uncertainty reaches a user-required threshold, thereby minimizing expensive measurements while maintaining model quality.

Then, we address the problem of hidden information in aggregated measurements. We derive the novel **Brownian Integral Kernel (BIK)** that quantifies the additional uncertainty inherent in integrated data from quantities that behave like Brownian motions – a typical behavior in physical processes. By modeling this uncertainty we enable accurate regression and quality adherence in scenarios where direct observation is impractical.

Furthermore, we investigate the inclusion of domain knowledge to enhance data efficiency. In the context of surrogate model-based optimization for industrial manufacturing, we demonstrate how incorporating domain-agnostic and domain-informed engineering knowledge reduces the need for expensive high-fidelity simulations. To align these models with specific discovery goals, we propose **Objective Alignment (OA)**, which automatically matches the training goal of a model with the specific information needs of the discovery task through gradient-weighted loss functions.

Finally, we apply the AKD methodology to the, in the AKD context novel, task of multi-sample testing, showing that **Adaptive Multi-sample Testing (AMT)** can determine distribution differences more data efficiently than non-adaptive classic testing methods while adhering to a given significance level.

Overall, this dissertation establishes AKD as a fundamental methodology to solve diverse knowledge discovery tasks in resource-constrained and non-stationary environments. We demonstrate the benefits of our methods through extensive studies on real-world use cases, ranging from process monitoring, over load forecasting, to textile draping using synthetic and real-world data. To facilitate reproduction, we release our algorithms, and benchmarks on open-source platforms.

Zusammenfassung

Knowledge Discovery (KD) – die Extraktion valider, neuartiger und handlungsrelevanter Muster aus Daten – hat einen tiefgreifenden Einfluss auf die moderne Wissenschaft und Industrie. Es ermöglicht Fortschritte in Bereichen wie den Materialwissenschaften, der Medizin und der Biologie sowie die Optimierung komplexer industrieller Fertigungsprozesse.

Klassische KD-Prozesse setzen jedoch häufig voraus, dass Daten in einem zentralen Speicher leicht verfügbar sind. In der Realität ist die Datenerhebung oft ein teures und zeitaufwendiges Unterfangen, das Materialien, Arbeitskraft und Energie erfordern kann. Solche Datenerhebungskosten werden für KD besonders dann zur Herausforderung, wenn (1) der zugrunde liegende datengenerierende Prozess nicht stationär ist, d. h. Concept Drift aufweist, und wenn (2) Messungen integriert oder aggregiert vorliegen, was die eigentlich relevante Größe verschleiert. Hier macht die kostspielige Datenerhebung die Erfassung ausreichender Datenmengen für traditionelle Lösungen oft unpraktikabel. Während etablierte Felder wie Active Learning und Bayes'sche Optimierung daran arbeiten die Dateneffizienz zu steigern, konzentriert sich die bestehende Literatur häufig auf wenige gut erforschte Aufgaben und vernachlässgt oft die Verknüpfung zwischen diesen Bausteinen und anderen KD-Aufgaben. Das Ziel dieser Dissertation ist es, diese Lücke zu schließen, indem sie die Nützlichkeit von **Adaptive Knowledge Discovery (AKD)** in neuen Szenarien und für neue Aufgaben aufzeigt.

Zunächst adressieren wir die Herausforderungen die durch Concept Drift entstehen, indem wir **Data Efficient Active Learning (DEAL)** vorstellen, einen neuen Ansatz für Regression unter Drift. DEAL modelliert Drift explizit als stochastischen Prozess und schätzt die Zunahme der Unsicherheit über verschiedene Datenregionen hinweg. Dies ermöglicht es Modellkalibrierung nur dann durchzuführen, wenn die Unsicherheit einen benutzerdefinierten Schwellenwert erreicht, wodurch teure Messungen minimiert werden, während die Modellqualität erhalten bleibt.

Anschließend widmen wir uns verborgenen Informationen in aggregierten Messungen. Hierzu leiten wir den neuartigen **Brownian Integral Kernel (BIK)** her. Dieser quantifiziert die zusätzliche Unsicherheit, die integrierten Daten aus Größen mit Brownschem Bewegungsverhalten innewohnt – ein charakteristisches Verhalten physikalischer Prozesse. Durch die Modellierung dieser Unsicherheit ermöglichen wir eine präzise Regression und Qualitätssicherung in Szenarien, in denen eine direkte Beobachtung nicht möglich ist.

Darüber hinaus untersuchen wir die Einbindung von Domänenwissen zur Steigerung der Dateneffizienz. Im Kontext der Surrogatmodell-basierten Optimierung für die industrielle Fertigung zeigen wir, wie die Einbeziehung von domänenagnostischem und domänenspezifischem Ingenieurwissen den Bedarf an teuren High-Fidelity-Simulationen reduziert. Um diese Modelle auf spezifische KD-Aufgaben auszurichten, schlagen wir **Objective Alignment (OA)** vor, welches das Trainingsziel eines Modells durch gradientengewichtete

Verlustfunktionen automatisch mit dem spezifischen Informationsbedarf der KD-Aufgabe abgleicht.

Schließlich wenden wir die AKD-Methodik auf die im AKD-Kontext neuartige Aufgabe des Multi-Sample-Testens an. Wir zeigen, dass **Adaptive Multi-sample Testing (AMT)** Verteilungsunterschiede dateneffizienter bestimmen kann als nicht-adaptive klassische Testmethoden, während AMT ein vorgegebenes Signifikanzniveau einhält.

Insgesamt etabliert diese Dissertation AKD als grundlegende Methodik zur Lösung vielfältiger KD-Aufgaben in ressourcenbeschränkten und nicht stationären Umgebungen. Unter Verwendung synthetischer und realer Daten demonstrieren wir die Vorteile unserer Methoden durch umfangreiche Studien an realen Anwendungsfällen, die von der Prozessüberwachung über die Lastprognose bis hin zum Textil-Tiefziehen reichen. Zur Erleichterung der Reproduktion stellen wir unsere Algorithmen und Benchmarks auf Open-Source-Plattformen zur Verfügung.

Table of Contents

Appendix

Part I.

Introduction

1. Disertation Overview

1.1. The Concept of Adaptive Knowledge Discovery

Traditionally, paradigms such as Knowledge Discovery in Databases (KDD) [HKP11] and the Cross-Industry Standard Process for Data Mining (CRISP-DM) [She00] describe 'the process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data' [Run25]. Despite their widespread adoption, these classical methodologies generally assume a 'passive stance' regarding data acquisition: the data is treated as a pre-existing asset, collected and stored in a warehouse or data lake before the discovery process begins. Under this convention, Knowledge Discovery (KD) is a downstream task, separated from the mechanisms that generated the data.

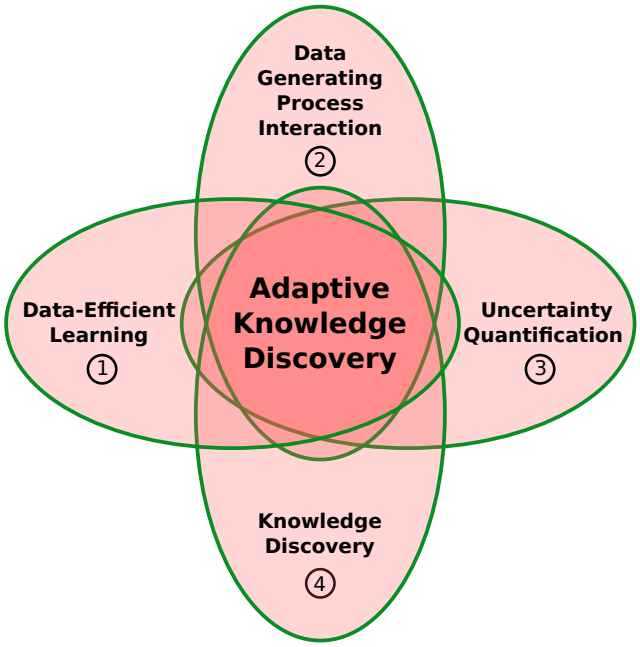


Figure 1.1.: AKD as the intersection of established fields.

In opposition to this convention, we will investigate Adaptive Knowledge Discovery (AKD). Methodically, AKD is the intersection of (1) **data-efficient learning**, through (2) **interaction with a Data-Generating Process (DGP)**, guided by (3) **Uncertainty Quantification (UQ)**, working together to solve a given (4) **knowledge discovery task**, see Figure 1.1. One can think of AKD as a methodology rather than a specific field, which can be implemented in diverse knowledge discovery tasks. By interacting with the DGP rather than treating it as a static data repository, AKD addresses the myriad of cases where data does not exist in excess, but instead is a resource constrained by time, money, and physics.

1.2. Motivation: The High Cost of Knowledge

In response to the limitations of the passive data acquisition paradigm in high-stakes environments, this dissertation addresses the struggle of traditional machine learning in sparse observation scenarios. By pivoting toward an adaptive approach to knowledge

discovery, this research overcomes the economic and physical limits of passive methods across three defining challenges.

1.2.1. Economic and Physical Barriers of Data Acquisition

The primary obstacle in these domains is that data is expensive. In materials science, for instance, characterizing a new alloy might require destructive testing, where a physical prototype is permanently rendered unusable to obtain a single measurement. Beyond physical destruction, computational overhead presents a similar bottleneck; high-fidelity simulations of complex systems, such as textile draping or climate models, can consume thousands of high-performance computing CPU hours per observation.

In such contexts, every data point carries a significant cost, making an exhaustive sampling strategy prohibitively expensive. AKD addresses this issue by learning a surrogate model of the DGP, quantifying data uncertainty in order to prioritize the acquisition of the most informative data points while minimizing the total number of costly physical or computational experiments.

1.2.2. Non-Stationarity and Concept Drift

A real-world DGP is rarely static. Industrial plants operate in environments where temperatures shift, catalysts degrade, and sensor precision fluctuates over time. This *Concept Drift* means that knowledge extracted from past data may become obsolete.

However, continuously monitoring every variable at high frequency to detect such shifts is often impractical, especially when physical observations are costly or resource-intensive. Consequently, an AKD context requires robust techniques to strategically determine *when* and *where* new measurements are most critical to maintain model validity without exhausting limited sampling budgets.

1.2.3. Information Loss in Integrated and Aggregated Data

A significant amount of real-world data is not observed directly but in an aggregated or transformed state. For instance, smart meters measure the integral of power consumption over time, rather than instantaneous demand. Similarly, temperature measurements are inherently constrained by heat capacity; a sensor cannot capture a perfectly instantaneous thermal state, but rather an aggregate of thermal energy exchanged over a finite duration.

These integrated signals obscure the underlying high-frequency dynamics of the process, creating a "low-pass filter" effect that masks critical information. Traditional learning algorithms often fail to account for the latent uncertainty and the loss of granularity inherent in these aggregates. To bridge this gap, an AKD framework necessitates specialized techniques—such as tailored kernels and strategic interaction with the process—to resolve these ambiguities and reconstruct the underlying latent phenomena from aggregated observations.

1.3. Research Questions and Contributions

The overarching goal of this dissertation is to establish the methodology of AKD by addressing the gap between data mining on cheep data and the requirements of interactive, resource-constrained environments. We formulate our contributions through four central research questions, each highlighting a specific pillar of the AKD methodology:

1.3.1. Q1: Regression under Drifting Data-Generating Processes

How can we maintain regression model quality in non-stationary environments while minimizing expensive measurements of a DGP?

Traditional Active Learning (AL) assumes a static underlying distribution. However, in industrial settings, the DGP often undergoes concept drift. In Part III, we address this challenge, focusing on pillar (2) the **interaction with the DGP**.

- **Contribution:** We introduce **Data Efficient Active Learning (DEAL)** [BFB24], a framework that utilizes time-variant Gaussian Process kernels to learn 'the drift behavior'. By identifying when temporal shifts cause uncertainty to exceed a critical threshold, DEAL strategically manages the interaction between the learner and the DGP, ensuring the model remains valid without exhausting the sampling budget.

1.3.2. Q2: Uncertainty Quantification of Aggregated Data

How can we quantify uncertainty and perform regression on quantities that cannot be observed directly due to aggregation?

When sensors provide only integrated observations, the underlying instantaneous states are hidden. In Part IV, we advance the methodical pillar (3) of **Uncertainty Quantification** for such integrated signals.

- **Contribution:** We derive the **Brownian Integral Kernel (BIK)** [BFB25], a novel covariance function that allows for exact Bayesian inference on aggregated data from a Brownian process. By providing a mathematically rigorous quantification of uncertainty for latent states, BIK enables AKD to operate in environments where the ground truth is fundamentally obscured. Further, the BIK enables the reconstruction and sampling of possible underlying latent trajectories that remain consistent with integrated observations.

1.3.3. Q3: Domain Knowledge and Objective Alignment

How can domain-specific engineering heuristics be leveraged to reduce simulation costs in industrial manufacturing?

In manufacturing, identifying optimal parameters typically relies on expensive high-fidelity simulations, such as the Finite Element Method (FEM). Because these simulations are computationally exhaustive, iterative trial-and-error is often impractical. In Part V, we investigate the pillar (1) of **data-efficient learning** through the lens of prior knowledge integration.

- **Contribution:** We propose **Objective Alignment (OA)** [Böh+24a], a method that integrates engineering heuristics directly into the acquisition function. By aligning the sampling strategy with specific physical goals rather than treating the optimization as a black box, we drastically reduce the number of simulations required to reach an optimal design. Further, we categorize and evaluate methods for incorporating both *domain-agnostic* and *domain-informed* knowledge. By mapping knowledge complexity against the difficulty of implementation, we provide a framework for enhancing surrogate accuracy in resource-constrained environments, demonstrating that strategic knowledge integration can substitute for massive datasets.

1.3.4. Q4: Expanding AKD to New Tasks—The Case of Multi-sample Testing

How can the AKD methodology be brought to new knowledge discovery tasks by synthesizing its core building blocks?

While previous parts of this dissertation focused on established frameworks like AL and Surrogate Model-based Optimization (SuMO), the final contribution demonstrates the universality of the AKD methodology. In Part VI, we synthesize our findings regarding the **(2) Data-Generating Process**, **(3) Uncertainty Quantification**, and **(1) data-efficient learning** to solve a **(4) knowledge discovery task** that has traditionally been treated with static, fixed-sample methods: *Multi-sample Testing (MT)*.

- **Contribution:** We develop **Adaptive Multi-sample Testing (AMT)** [BKew]. By utilizing Bayesian upper confidence bounds to guide sampling across different groups, we demonstrate that the AKD methodology can achieve higher statistical power with fewer observations than classical frequentist methods, while maintaining rigorous Type I error control. Here we combine our findings, demonstrating that by understanding the underlying process dynamics and quantifying epistemic uncertainty, we can transform a "passive" statistical problem into an "active" AKD task.

1.4. Real-World Use Cases

Unlike dissertations that focus on a single application, this work demonstrates the versatility of the AKD methodology by applying it across diverse domains. We selected these use cases specifically because they each embody the ‘expensive data’ constraints central to our thesis: physical destruction, high-performance computing costs, and privacy aggregation.

1.4.1. Energy Systems and Environmental Monitoring

The transition to smart grids and sustainable energy management requires precise modeling of stochastic loads and environmental conditions. We investigate two distinct scenarios within this domain:

Load Forecasting and Privacy-Preserved Metering Modern smart metering infrastructure often imposes a 'low-pass filter' on data collection. To preserve user privacy and reduce bandwidth, smart meters typically aggregate consumption into 15-minute intervals (integrated data).

However, grid stability requires knowledge of instantaneous peaks. We utilize data from the *High-resolution Industrial Production Energy (HIPE)* dataset [Bis+18] and the *Load Profile Generator* [Pfl+22] to demonstrate how AKD can reconstruct latent high-frequency dynamics from these privacy-preserved signals, preventing capacity violations without requiring invasive surveillance.

Stream-Based Environmental Monitoring In environmental contexts, such as soil quality monitoring or river sensing, the cost is not in generating the data stream (which runs continuously) but in *labelling* it (e.g., sending a physical sample to a lab). We simulate this using stream-based regression tasks where the learner must decide whether the current environmental state justifies the cost of an expensive measurement.

1.4.2. Virtual Optimization in Composite Manufacturing

In the field of materials science, we address the optimization of *Continuous-Fiber Reinforced Plastics*. These materials offer superior mechanical properties but are notoriously difficult to manufacture defect-free.

Textile Draping Optimization The quality of a composite part is determined during the *draping* process, where a flat woven fabric is formed into a 3D shape.

In our use case, minimizing defects like wrinkling or 'dry spots' requires optimizing the stiffness of 60 individual grippers along the textile perimeter. Since a single FEM simulation of this process is computationally expensive, we cannot afford exhaustive search. We use this application to demonstrate how **Objective Alignment (OA)** allows us to consider physical heuristics (e.g., minimizing shear angles $\boldsymbol{\gamma}$) directly into a surrogate model, drastically reducing the number of required simulations compared to black-box optimization.

1.4.3. Fundamental Scientific Discovery

Finally, we address the foundational problem of comparing populations in scientific trials – whether in medicine, biology, or psychology.

Adaptive Hypothesis Testing In many scientific workflows, data acquisition is destructive (e.g., testing material strength to failure) or ethically constrained (e.g., human clinical trials). Traditional statistical tests require collecting all data upfront, often wasting samples on obvious non-matches. We frame this challenge through the 'Fake Coin Detection' analogy: An agent must detect a deviant distribution (the weighted coin) with the minimum number of tosses. By applying AKD to this fundamental task, we show how adaptive

sampling can accelerate discoveries in fields ranging from microarray studies to superconductivity classification.

Part II.

Fundamentals, Positioning, and Related Fields

2. The Building Blocks of Adaptive Knowledge Discovery

While the previous chapter established the motivation for Adaptive Knowledge Discovery (AKD), this part serves as a joint foundation for the technical formalisms, and state-of-the-art context required for the subsequent contributions. We transition from a high-level conceptual overview to a rigorous mapping of the four building blocks identified in Figure 1.1: (1) **Data-Efficient Learning** strategies, (2) **Interaction** with the Data-Generating Process (DGP), (3) **Uncertainty Quantification (UQ)**, and (4) the resulting **Knowledge Discovery Tasks**.

The objective of this chapter is to provide a unified framework that bridges established related work with the novel methodologies presented in this dissertation.

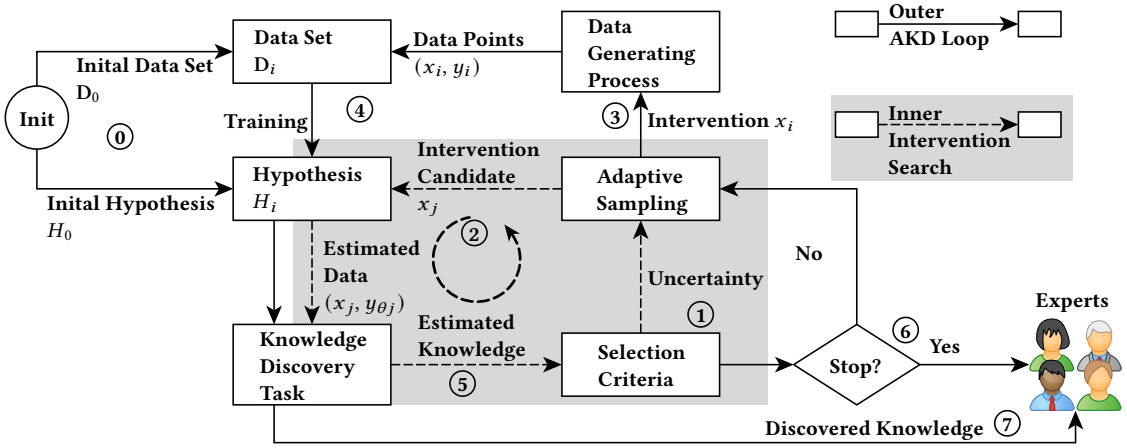


Figure 2.1.: The general AKD loop: In the inner cycle (grey) intervention candidates are searched, and in the outer cycle the most promising intervention is performed to obtain knowledge.

The four building blocks of AKD do not operate in isolation; rather, they form a closed-loop system of iterative refinement with the overall goal of **Data-Efficient Learning**. As illustrated in Figure 2.1, the AKD process begins with an initial hypothesis H_0 and data set $\mathbf{D}_0$ (Step 0). At each iteration i, a *Selection Criteria* uses **Uncertainty Quantification** to determine whether knowledge is insufficient in regions of the data space (Step 1). The *Adaptive Sampling* strategy – often enriched by Prior Knowledge – uses these uncertainties to find (Step 2) and select (Step 3) the most informative **Intervention** x_i from the DGP using the inner Intervention Search loop. The resulting observation y_i is then used to update the data set $\mathbf{D}_i$ and hypothesis H_i (Step 4), estimating the resulting knowledge (Step 5). Once sufficient accurate knowledge is obtained or the data budget is exhausted (Step 6), the final solution to the **Knowledge Discovery Task** is returned (Step 7). This

cycle, serves as the algorithmic template for the Data Efficient Active Learning (DEAL), Surrogate Model-based Optimization (SuMO), and Adaptive Multi-sample Testing (AMT) frameworks presented in the following chapters.

In the following sections, we decompose AKD into its fundamental mechanics. We begin by formalizing the variables and cost structures of the DGP (Section 2.1), establishing how one can interact with a DGP. We then synthesize the literature on data-efficient learning, positioning adaptive sampling alongside data augmentation and prior knowledge integration (Section 2.2). Furthermore, we define the mathematical sources of uncertainty that guide adaptive sampling (Section 2.3). Finally, we provide a overview of AKD tasks (Section 2.4), and related adaptive tasks such as reinforcement learning.

2.1. Adaptive Knowledge Discovery and the DGP

One of the key points of AKD is the interaction with the DGP. That is, an AKD algorithm has direct access to the DGP, can change DGP inputs, and observe the resulting DGP outputs. This sets AKD apart from classic knowledge discovery, where algorithms typically can not interact with the DGP, and data is collected before any knowledge discovery is performed. While interacting with the DGP enables a informed search of the data space, and if done correctly results in better data efficiency, it is also the main limitation of AKD. That is, AKD can only be used in scenarios where such interaction is possible. In consequence, AKD is especially useful in experimental science where experts have full control over the DGP. DGP interaction being such an important building block, we will formalize such interaction in the following.

2.1.1. Variables and Formalization of a DGP

Interaction with complex real-world systems, where variables often exhibit time-dependency, and noise, introduces nuances that are not well captured by the classic naming convention: *independent* and *dependent* variables.

Specifically, these terms fail to distinguish between process inputs, external influences, intervening mechanisms, and measured outcomes. Thus, in the following, we provide a more rigorous overview of variable types and their role in Table 2.1. Figure 2.2 conceptualizes the variable relationships in respect to the DGP, offering a visual complement to the detailed explanations provided in Table 2.1.

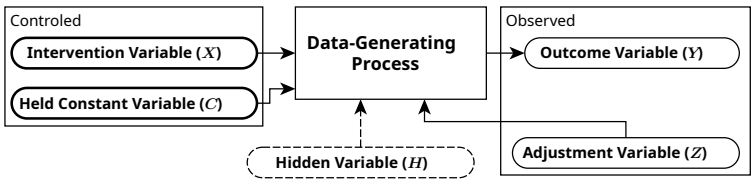


Figure 2.2.: Variables of a DGP.

In short, one can categorize the variables based on their causal role within the DGP and the researcher's level of control over them. At the core of the experiment is the *Intervention Variable*, which is actively manipulated to observe its effects on the *Outcome Variable*. To ensure a clean signal, other factors are either *Held Constant* through experimental design or treated as *Adjustment Variables* (covariates) to statistically account for their

Variable Name(s)	Role in the DGP / Experiment	Action by Researcher / Analyst
Intervention Variable (X) Manipulated Variable Treatment Variable (Varied) Control Variable	The primary factor whose *effect* is being studied.	**Actively Changed** (intervened upon) across groups or conditions to observe the result ($do(X = x)$).
Held Constant Variable (C) (Fixed) Control Variable	A factor that could influence the outcome but is *not* the primary focus.	**Explicitly Kept Constant** (Fixed) at one level to isolate the effect of the Intervention Variable.
Adjustment Variable (Z) Covariate (Observed) Confounder	A factor that can influence both the Input and Output. (Often difficult to control.)	**Observed** (or measured) to statistically *adjust for its influence.*
Hidden Variable (H) Environmental Variable Exogenous Variable (Unobserved) Confounder	A external factor that impacts the DGP. (Often difficult to measure.)	Typically **Unobserved** and introduces noise or bias.
Outcome Variable (Y) Response Variable	The resulting measurements from a DGP after inputs and interventions.	**Observed** (or measured) as the final result of the DGP or experiment.

Table 2.1.: Rigorous overview of variable types of an DGP.

influence. Finally, the process recognizes the presence of *Hidden Variables*, which represent unobserved or environmental factors that can introduce noise or bias into the system.

Formalization: Observing a random variable X is denoted as $x \leftarrow X$, where x is the observation. In the case of a *Intervention Variable* X one can set X to a fixed *Intervention Value* x (short *input*) which is denoted as: $do(X = x)$. A DGP is defined by a *Random Function* $C : X \mapsto C(X) = Y_X$ which is usually refereed to as the underlying *Concept* of the DGP. Here Y_X is the random outcome variable of the DGP. Intervening on X, i.e., setting X to a value x, is then denoted as $C(x) = Y_X$ (here $do(X = x)$ is implicit), and the resulting observation is named y_x. X, x and Y_X, y_x have *Input Space* $\mathcal{X}$, and *Outcome Space* $\mathcal{Y}$ respectively. To highlight that x, y_x is made at a specific time t or iteration i one may use x_t, y_t and x_i, y_i respectively.

Random Functions extends the notion of random variables to a continuous function space [LRS13]. They are in some sense equivalent to conditional random variables $C(X) = Y_X = Y \mid X$ where Y alone would be the random outcome variable not conditioned on intervention variable X [C05]. The extension to a continuous function space comes into play, when instead of drawing a single observation y_x from a concept $C(x)$ one draws a *Sample Function* $c(\mathcal{X}) \leftarrow C(\mathcal{X})$. Here, one can think of a sample function $c(\mathcal{X})$ as the function build by observing all possible x across the input space $\mathcal{X}$. It follows that given a specific x we have a mapping $c : x \mapsto c(x) = y_x$ to a single observation y_x.

2.1.2. Observation Cost of a DGP

In the context of AKD, a fundamental premise is that acquiring observations from the DGP is associated with a non-negligible cost. These costs act as the primary constraint on the adaptive sampling process, necessitating an acquisition function that maximizes information gain per unit of resource expended. We categorize these costs into three distinct dimensions: (1) *Intervention Costs*, (2) *Process Costs*, and (3) *Measurement Costs*.

Intervention Costs encompass the resources required to initialize the state of the DGP or to perturb its parameters according to a specific intervention value. This includes the direct procurement of finite physical resources, such as high-purity chemical reagents in laboratory experiments or the fuel consumed during the test flight of an aerospace prototype. Furthermore, this category accounts for the 'setup cost' of reconfiguring a production line or recalibrating sensitive hardware between different experimental runs. In the digital domain, this might manifest as the cloud egress fees or API credits required to trigger a remote data-generating service.

Process Costs occur during the execution of the process itself and is primarily driven by the transition from an input state to the final observable output. The most ubiquitous process cost is time. For instance, longitudinal clinical trials or high-fidelity Finite Element Method (FEM) simulations may take weeks or months to yield a single data point. Beyond temporal constraints, this category includes significant computational overhead, such as the thousands of GPU hours required for training deep learning models or running complex climate simulations. Additionally, process costs can take a non-monetary form, such as the moral or physiological cost of treating a patient with an experimental drug for an unknown disease. In this scenario, the DGP involves a living system that may be negatively impacted during the observation period, making every intervention a high-risk endeavor.

Measurement Costs are incurred at the final stage of the DGP when extracting quantitative values from the system. In many engineering contexts, this involves destructive testing, where a product is rendered unusable to obtain data on its material properties (e.g., tensile strength testing). Measurement also carries significant ethical weight in biological research; for example, obtaining histological data often necessitates the sacrifice of a rodent subject [HS12], representing a permanent loss of a biological unit. In the social sciences, this might involve 'survey fatigue', where the act of measurement itself reduces the likelihood of the subject providing accurate data in future iterations.

Beyond the nature of these costs, their temporal distribution significantly impacts the design of AKD strategies. Specifically, we distinguish between *sequential costs*, where each query must be completed before the next can be initialized (common in physical material testing), and *batch-parallel costs*, where multiple simulations or experiments can be run concurrently. While this dissertation primarily focuses on sequential data-efficiency to minimize the total number of queries, the formalisms provided lay the groundwork for transition into batch-adaptive scenarios where the trade-off between lead time and total resource consumption is critical.

2.2. Adaptive Knowledge Discovery and Data-Efficient Learning

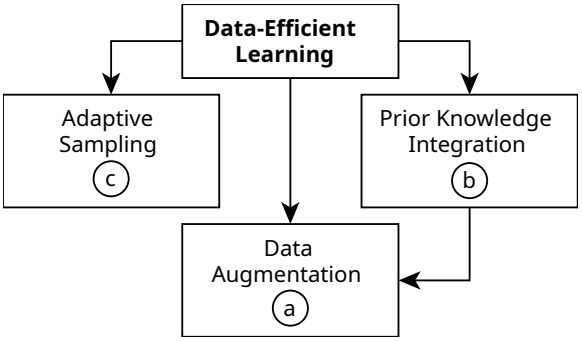


Figure 2.3.: Common practices used for data-efficient learning.

One of the key benefits and promises of AKD is data-efficient learning. Thus, in the following, we will position AKD in the context of data efficiency. Data-Efficient Learning is commonly achieved in three orthogonal practices: (a) *Data Augmentation*, which enriches the data with plausible synthetic data; (b) *Prior Knowledge Integration*, which is used to reduce the space of possible models; (c) *Adaptive Sampling*, which samples data iteratively conditioned on the previously observed data to obtain data with higher informational content; and combinations of these techniques, see Figure 2.3. Within this classification, AKD is clearly situated within adaptive sampling. Nonetheless, data-efficient learning being one of the main goals of AKD, the other two practices for data efficiency can often be combined with adaptive sampling, making the solution even more efficient. This is why we will provide an overview of all three practices and discuss their interconnection.

2.2.1. Data Augmentation

The goal of data augmentation is to enrich training data and generate larger, *meaningful* datasets from sparse ones. Models trained with augmented data should ideally generalize better and be more robust than models trained without augmentation. This brings us to an important point: The augmented data set $\mathbf{D}_{\text{good}}$ must be a superset of the original data set $\mathbf{D}$, where data outside the original distribution should remain consistent with the real-world data $\mathbf{D}_{\text{real}}$ [CDL20]. Over-augmented data $\mathbf{D}_{\text{bad}}$ is not directly harmful as long as it does not contain non-matching data $\mathbf{D}_{\text{BAD}}$, however, it reduces model capacity by forcing the model to learn useless connections. These relationships are expressed as:

$$\mathbf{D} \subset \mathbf{D}_{\text{good}} \subseteq \mathbf{D}_{\text{real}}; \quad \mathbf{D}_{\text{Bad}} : \mathbf{X}_{\text{Bad}} \cap \mathbf{X}_{\text{real}} = \emptyset; \quad \mathbf{D}_{\text{BAD}} : \mathbf{X}_{\text{BAD}} \subset \mathbf{X}_{\text{real}}, \mathbf{Y}_{\text{BAD}} \cap \mathbf{Y}_{\text{real}} = \emptyset, \tag{2.1}$$

with *Input Set* $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$ and *Observation Set* $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\}$.

From this connection directly follows that one can only augment data by employing *Prior Knowledge*, which highlights the close connection between the two fields. In fact, data augmentation can be seen as a method to include *Prior Knowledge*. In the following, we will give a short, non-exhaustive overview of augmentation techniques with increasing complexity.

Noise, and Transforms are one of the simplest forms of augmentation. Examples include: Adding noise [SD91; Akb23] which can be equivalent to using loss regularization [Bis95]. Masking data [Che+24] to make models robust against missing data, which is a problem

in many real-world datasets [Cui+24]. Rotation, shifting, and cropping of data [Che+20; TMU20], often used with geometric data such as image data but also applicable to geometric sensor arrays or spatial material distribution.

Data-Driven Augmentation uses cheap, readily available data of a similar kind. Here, the prior knowledge comes into play when choosing this external data - ensuring its similarity. One of the simplest forms of using the external data is directly extending the dataset with the external data, often pretraining the model first, and then finetuning the model on the original data [Yos+14]. This is closely related to model preconditioning which we will discuss in more detail in Section 2.2.2. More complex techniques use generative models [ASE18] trained on the external data to generate even more additional data of a similar kind.

Domain-Inspired Augmentation is at the upper end of the use of prior knowledge. Here, one uses the known domain knowledge to augment the data [ASE18]. Examples include: Using the tabular structure of databases for augmentation [Cui+24]. Modeling known physical processes using them as an advanced version of transform-based augmentation [Kar+17b; Wil+20]. Or calibrating the physical models with the given data, using them to generate synthetic data for model pretraining [ASE18; Wil+20; Liu+25]. Such methods can be considered at the very border of Augmentation and Prior Knowledge Integration by Model Preconditioning, Section 2.2.2.

2.2.2. Prior Knowledge Integration

As discussed previously, data augmentation can be considered a basic way of incorporating prior knowledge. Here, we distinguish between adapting merely the data (Data Augmentation) and adapting the actual model (Prior Knowledge Integration). We categorize the integration of prior knowledge as follows: (I) data encoding, (II) loss function, (III) model preconditioning, and (VI) model adaptation.

Data Encoding for prior knowledge integration refers to building an encoding for a specific task so that it contains additional prior information not included in the raw data. This can start with simple feature engineering [Joh20], over to data-driven encoding like Word2Vec [Mik+13], where readily available external data is used to embed semantic prior knowledge into the encoding, up to encodings that explicitly embed positions [Vas+23], topologies [Cui+24], or physical constraints [BKC13; Böh+24b]. It is important to note, and often overlooked, that not only the encoding of the model input but also the encoding of the output matters, as the contained information back-propagates during training.

Loss Functions can also be used to include prior knowledge, often done in the form of (Domain-Informed) regularization terms [Hua+24; HMT25]. Here, one main challenge is balancing the different loss terms [WTP21; CGK18], and formulating the terms themselves from existing domain knowledge, which mostly requires a domain expert.

Model Preconditioning tries to select parameters for a model that are already beneficial for solving the given task. We will start with data-driven approaches: The simplest form is pretraining a model on external data for transfer learning [Yos+14], which we already introduced as part of data augmentation in Section 2.2.1. By such pretraining one effectively reduces the space of possible models. By date, a lot of already pretrained models exist publicly available, for example on *huggingface* or in *Model Zoos*. A different approach is knowledge distillation [HVD15], where one tries to benefit from the knowledge contained in already pretrained models, but using it to train a completely new (mostly smaller) model. The final data-driven approach is using pretrained meta-models for few-shot learning [FAL17]. While such an approach requires very few new data points, it requires a meta-model that fits the task, trained on very large datasets, which is often not available or impractical for computational reasons. Moving away from data-driven approaches, are methods building on the already discussed concept of Physically Inspired Augmentation [ASE18; Wil+20]. Here, one can use the physical constraints to generate synthetic data for pretraining. Similarly to this is the use of existing physical simulations for synthetic data generation [Liu+25; Gol+18].

Model Adaptation is the most direct method for including prior knowledge, but also the most complex as it prevents the use of standard models. Here, the architecture of the model itself is changed to reflect the prior knowledge. This can range from simple architecture decisions like using convolutions [FM82], over to using other transformations or constraints like enforcing symmetry [CW16], to models whose structure reflects known physical relations [GDY19], or which explicitly contain physical equations [Xu+24; LKN25]. A noteworthy method is that of self-supervision [Che+19; Che+20], which bridges the gap between model preconditioning, model adaptation, data augmentation, and loss functions. A model is built such that it creates multiple different (intermediate) predictions, which are transformed into each other by using known physical relations, making sure with an additional loss term that they are consistent with each other, thereby preconditioning the weight of the model.

2.2.3. Adaptive Sampling

Adaptive sampling aims at selecting data that is most beneficial for solving a given task. To achieve this, it requires prior knowledge of the task, which distinguishes it from reinforcement learning. The latter is unaware of the task and must learn it while interacting with the data-generating process. While *Data Augmentation* and *Prior Knowledge Integration* can be used in many contexts, and are orthogonal to AKD, adaptive sampling is much more closely related to AKD. This is if one uses adaptive sampling for the goal of data-efficiently solving a knowledge discovery task by directly interacting with the data-generating process, it is an instance of AKD. In this section we will focus on adaptive sampling for the task of machine learning, which is usually referred to as *Active Learning (AL)*. Active learning specifically is a well researched AKD Task, and thereby provides a good starting basis to learn about AKD. A detailed overview is, for example, provided by [Set09; Set12]. An extensive survey on active learning sampling strategies is provided by [KG20]. However, not being the only AKD Task, in the following we will also provide some

adaptive sampling strategies from other fields. Generally, we can distinguish between two classes of adaptive sampling: heuristic-based sampling and probabilistic-based sampling.

Heuristic-Based Sampling uses heuristics to select the next data point(s) based on the current model state or the data distribution. These heuristics are often designed to target areas where the model is performing poorly or where the data is expected to be most informative, based on domain knowledge or computational measures. A common heuristic is Difficulty-Based Sampling, selecting points that are difficult to distinguish for the model, i.e., selecting points close to a decision boundary [Set12], or inconsistently labeled ones [Gao+20]. Another class of heuristics focuses on diversity and coverage, aiming to cover the input space as broadly as possible with as few data points as possible [SS18] or to select points far from those already observed [Wu13].

Probabilistic-Based Sampling is often rooted in *Information Theory* or *Bayesian Statistic*. These methods typically aim to minimize the expected uncertainty or maximize the expected information gain with respect to the model parameters or predictions. *Uncertainty Sampling* simply selects the data points for which the current model yields the least confident prediction. Here, prediction confidence (or the inverse, prediction uncertainty) can be measured in various ways giving rise to the last pillar of AKD: *Uncertainty Quantification*, discussed in Section 2.3. Moving closer to Bayesian statistics, *Query-by-Committee (QBC)* uses an ensemble of models, selecting points where the ensemble disagrees the most [SOS92]. Here, the empiric predictive distribution of the ensemble can be interpreted as an estimate of the version space or as a rough approximation of a Bayesian prediction. Extending this thought naturally results in the use of Bayesian models to directly estimate such a distribution and select according to the distribution properties like predictive variance [CGJ94] or expected entropy [HPB08]. Then, there is *Expected Error Reduction (EER)* [RM01]. While some models alow for (partly) analytic calculation of EER, general models require empiric approximation: Pseudo labeling data points for one input using the predictive distribution, training on these synthetic labels one at a time, pseudo labeling them again with the updated model, calculating the expected error reduction between the original and the new model across all pseudo labels and all possible inputs, and finally finding and selecting the one input that reduces this metric the most. In short, while statistically grounded, EER requires integrating over the predictive distribution to obtain an expected error, which is often computationally intractable. Similarly the *Knowledge Gradient* is a utility-based approach, involving a one-step look-ahead [FPD09] requiring integration. Switching to the task of *Surrogate Model-Based Optimization*, one often relies on the previews techniques to learn a cheap and sufficiently acurate mode and then uses this model in a second step for a optimum search [Loo+19; Zim+21]. Instead, *Bayesian Optimization* directly tries to sample data that helps finding a optimum. *Expected Improvement (EI)* [Moč75] chooses the highest expected difference between model predictions and the best function value found so far. *Upper Confidence Bound (UCB)*, which is also used in Bandit Algorithms [ACF02], simply chooses the highest function value after adding a confidence bound like the variance at the location. UCB performance depends strongly on the used confidence bound, i.e., how tight the bound is. For *Best Arm Identification*

(BAI) the method *Successive Rejects* [AB10], can be seen as a highly exploitative version of UCB, systematically eliminating one non-optimal option per phase. *Entropy Search* [HS12] pivots away from utility maximization toward a purely information-theoretic sampling, maximizing the information gain about a (unknown) maximum function value globally. However this global approach is computational more expensive again.

2.2.4. The Interconnection Between Practices for Data Efficiency

Looking at the two classes of adaptive sampling, we observe that all sampling methods are based on (I) the model, (II) its uncertainty, or (III) the data used for training. As summarized in Table 2.2, the synergy between these pillars is fundamental to the success of AKD:

(I) **The Model:** A better, more generalized model enables a more informed selection of data points. Such a model already incorporates *prior knowledge*, which reduces the need for broad exploration by identifying which data points are actually critical for the task. It is vital to distinguish between a *generalizing model* – which is constrained by task-specific knowledge and able to generalize from sparse data to unseen data – and a *general model*, which, while flexible and non task-specific, requires significantly more data to resolve the underlying process structure and determine data usefulness.

(II) **Uncertainty Quantification (UQ):** A task-specific estimation of uncertainty allows for focused data selection. While data augmentation provides better generalizing models, the inclusion of domain knowledge is essential for distinguishing between irreducible noise and areas where the model truly lacks knowledge.

(III) **Training Data:** Domain knowledge is required for initial filtering of the input space. An unlabeled dataset covering relevant inputs can reduce the search space and increase the 'relevant' model performance, whereas a dataset that is too dense or misaligned with the DGP can waste resources or harm model capacity. In this case, data augmentation needs to be aligned with the given task.

We can observe in each of these three cases that the efficiency of AKD is closely coupled with the specific knowledge discovery task. By injecting prior knowledge the 'precision' of the adaptive sampling process improves, leading to a more data-efficient solution of a given knowledge discovery task, all without necessarily changing the underlying sampling algorithm itself. This demonstrates that strategic knowledge integration acts as a orthogonal efficiency multiplier for data-efficient learning.

2.3. AKD and Uncertainty Quantification

Uncertainty Quantification (UQ) is a prerequisite for adaptive sampling. That is before a algorithm can decide where in the data space to obtain additional data it needs to know where knowledge is missing. In this section we will first give an overview over different types and sources of uncertainty and their interconnection. We will then dive into techniques for UQ.

Table 2.2.: Interconnection between practices for data efficiency and their benefit to AKD.

Target Pillar	Synergy with Prior Knowledge / Augmentation	Benefit to AKD Efficiency
The Model	Restricts the hypothesis space to physically plausible or task-specific functions.	Accelerates convergence by requiring fewer samples to find the 'true' function.
Uncertainty Quantification	Tailors the noise model to the physical constraints of the DGP.	Prevents sampling in regions where inherent noise (Aleatoric Uncertainty (AU)) is mistaken for missing knowledge (Epistemic Uncertainty (EU)).
Training Data	Domain-inspired filtering removes irrelevant or physically impossible parameter combinations.	Concentrates the limited data budget on regions with high informational or physical relevance.

2.3.1. Uncertainty Sources and their Interconnection

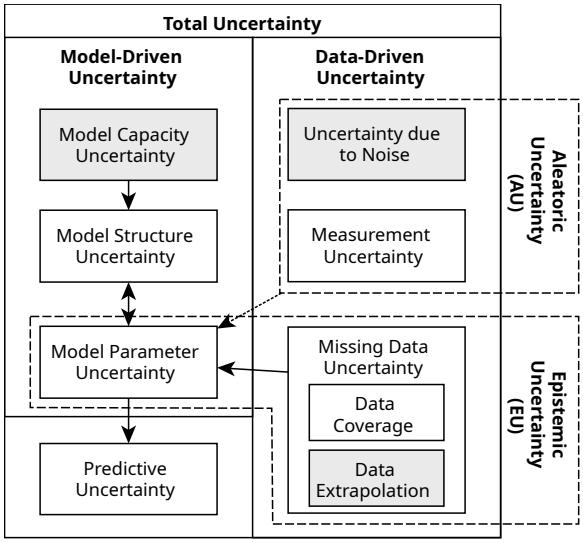


Figure 2.4.: Uncertainty sources and their interconnection.

The common high-level classification divides the Total Uncertainty (TU) into the irreducible Aleatoric Uncertainty (AU) and the reducible Epistemic Uncertainty (EU) [HW21]. AU is often associated with the uncertainty stemming from the data, arising from inherent noise or stochasticity within the DGP itself, such as measurement errors or the intrinsic randomness of physical quantities [Mal19]. EU on the other hand, often used synonymous with model uncertainty, arises directly from a lack of knowledge regarding the true model parameters or the underlying function [Sen+14]. We note that discerning between AU and EU is not straight forward, and there exist some critique concerning common decomposition methods [Wim+23]. AKD, and thereby adaptive sampling is explicitly designed to minimize EU to reduce model variance and improve generalization by gathering additional data. However, looking closer at the types and sources of uncertainties and their interconnection one can make a finer grain classification [HW21; KD09], see Figure 2.4.

Aleatoric Uncertainty (AU) consists of *Uncertainty due to Noise* (hidden variables) and *Uncertainty due to Measurement Errors* [Sen+14]. In classical statistics, AU is considered irreducible. However, in AKD contexts, AU can sometimes be reduced at a additional *Cost*

(Section 2.1.2) – for example, by deploying higher-precision sensors or by controlling or measuring hidden variables [KD09]. In later cases – transforming hidden variables to adjustment variables – practitioners must weigh the cost of reducing AU against the risk of increasing dimensionality which consequently increases EU [HW21].

Epistemic Uncertainty (EU) , while often associated with the model, actually stems from missing data and is the **primary target of AKD**. It manifests as uncertainty due to *Data Coverage* (sparsely sampled regions) and *Data Extrapolation* (regions outside of the observable data space). While coverage-based EU is reducible through iterative sampling at the expense of collecting more data, extrapolation-based EU (e.g., predicting the future) remains a fundamental challenge. Most AKD strategies aim to minimize the *Data Coverage* component of EU to ensure the model parameters accurately reflect the true DGP. This connection to the model parameters is why EU is often considered to be a model uncertainty. However, as discussed next, EU is only an indirect source of model uncertainty and not the only source.

Model-Driven Uncertainty comprises *Capacity, Structure, and Parameter Uncertainty.*

Model Capacity refers to the fact that even with access to infinite data, one is constrained by compute and storage demands which limits a model's learning capacity, leading to uncertainty. The current trend in deep learning buys its way out of this by using massive compute infrastructure but at a significant cost.

Structure Uncertainty stems from the fact that the *model choice* does not fit the underlining structure of the DGP, thus unable to learn the true structure. This mismatch can be due to an incorrect model selection, or by design, when AKD seeks 'simpler', interpretable models over complex black-boxes with the goal to gain human understandable knowledge. Further, AKD often aims at learning the structure itself instead of having a fixed structure and learning its parameters. Here, structure uncertainty can stem from limited model capacity not being able to reflect all structures, or from the uncertainty of the learned parameters as outlined next.

Parameter Uncertainty primarily stems from missing data, leading to a set of candidate parameters that satisfy the empirical data but fail to recover the DGP. This uncertainty is further compounded by model structure and capacity, impacting how parameters are learned and which space of parameters is allowed in the first place. While unbiased universal approximators mitigate architectural bias, they remain sensitive to the data-driven ambiguity. Inversely, when the models structure is a function of its parameters, parameter uncertainty propagates directly into structural uncertainty, consequently limiting the extracted knowledge.

Predictive Uncertainty manifests due to all the aforementioned uncertainty sources. With model capacity mostly assumed fixed, model structure assumed fixed or learned, with learned parameters, the remaining uncertainty in model's predictions can be attributed to the interplay of AU and the remaining EU which are both propagated through the parameters. Thus, as visualized in Figure 2.4, all these uncertainties are deeply interconnected. Crucially, because AU and EU interact with each other via the parameters, AU can only

be accurately estimated once EU is sufficiently reduced via AKD, until then one can only provide lower bounds on AU [Wim+23].

2.3.2. Formalization of Uncertainty Sources

In the previous section, we introduced different sources of uncertainty; following [HW21] with some adaptations, we will now provide a more formal understanding of the origin of these uncertainties. Note that this formalization is not restricted to AKD.

Data: We have *Intervention Variable* X, and *Outcome Variable* Y_X (both random variables). Accordingly we have a *Intervention Value* x (short input), which is also the *Input Value* to a model (clear from context), and *Observation* $y_x \leftarrow Y_X$. X, x and Y_X, y_x have *Input Space* $\mathcal{X}$, and *Output Space* $\mathcal{Y}$, respectively. From our interventions and observations we then have:
Data Set $\mathbf{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$.
Input Set $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$.
Observation Set $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\}$.
We further have our data distribution [probability measure, mass (PMF) or density (PDF), clear from context] $P(X, Y), p(x, y)$, the marginals $P(X), p(x), P(Y), p(y)$, and conditional $P(Y \mid X) = P(Y_X), p(y \mid x) = p(y_x)$.

Model: We then have a *Hypothesis Space* $\mathcal{H}$ in which a model can reside. A *Hypothesis* $h : \mathcal{X} \to \mathcal{Y}$ is a model that describes a connection between X and Y_X. As most models have *Parameters* θ, we have further a *Parameter Space* Θ and a *Parametrized Hypothesis Space* $\mathcal{H}_\Theta$. Formally, we consider the parametrized hypothesis space a subset of the general hypothesis space, $\mathcal{H}_\Theta \subseteq \mathcal{H}$, where $\mathcal{H}$ represents the broad family of potential models and $\mathcal{H}_\Theta$ denotes the specific functional forms reachable by a chosen parameterization $\theta \in \Theta$.

Accordingly, a model is a *Parametrized Hypothesis* $h_\theta(x) = y_{\theta x}$, with *Parameters* θ, and (point) *Prediction* $y_{\theta x}$. The *Prediction Set* $h_\theta(\mathbf{X}) = \mathbf{Y}_\theta$ is then the set of all predictions. To emphasize the connection between x and $y_{\theta x}$ at a specific time t one may also write $x_t, y_{\theta t}$.

Instead of a point prediction $y_{\theta x}$, a (stochastic) parametrized hypothesis $H_\theta(x) = Y_{\theta x}$ may also provide an (random) *Estimated Outcome Variable* $Y_{\theta x}$. This enables observing a realization $y_{\theta x} \leftarrow Y_{\theta x}$ from a predictive distribution $P(Y_{\theta x} \mid \mathrm{do}(X = x)) = P(Y_{\theta x})$. Some models directly provide the predictive distribution $P(Y_{\theta x})$ or its mass or density function $p(y_{\theta x})$, while others only allow sampling from it $y_{\theta x} \leftarrow Y_{\theta x}$.

Further, a parametrized (space) hypothesis $\mathcal{H}_\theta(x) = \mathcal{Y}_{\theta x}$ can provide a *Predictive Space* $\mathcal{Y}_{\theta x} \subseteq \mathcal{Y}$. A special case is a *Prediction Sample* $\mathbf{Y}_{\theta x} \underset{N}{\leftarrow} Y_{\theta x}$ of realizations from $Y_{\theta x}$. Here, $\underset{N}{\leftarrow}$ denotes obtaining a sample of realizations $\mathbf{Y}_{\theta x} = \{y_{\theta 1}, y_{\theta 2}, \ldots, y_{\theta N}\} = \{y_{\theta n} \leftarrow Y_{\theta x}\}_{n=1}^{N}$ of size N instead of a single realization $y_{\theta x} \leftarrow Y_{\theta x}$.

At this point, we want to highlight the difference between the *Marginal Coverage Probability* $P(Y \in \mathcal{Y}_{\theta x}) = \int_{\mathcal{Y}_{\theta x}} p(y)\,\mathrm{d}y$, which is the probability to observe any y from the DGP without conditioning on the input x to the DGP, and the *Conditional Coverage Probability* $P(Y \in \mathcal{Y}_{\theta x} \mid do(X = x)) = \int_{\mathcal{Y}_{\theta x}} p(y \mid x)\,\mathrm{d}y$, which is conditioned on the input x. *Conditional Coverage* is an important quality metric of algorithms and given if $P(Y \in \mathcal{Y}_{\theta x} \mid do(X = x)) \geq 1 - \beta$ for (almost) all $x \in \mathcal{X}$ is satisfied. The predictive

distribution $P(Y_{\theta x} \mid \mathrm{do}(X = x))$ of the estimated outcome variable $Y_{\theta x}$ can be used as an estimation $P(Y_{\theta x} \mid \mathrm{do}(X = x)) \approx P(Y \mid \mathrm{do}(X = x))$.

Optimization: We want to minimize the *Overall Loss* $\mathscr{L}(\mathbf{Y}_\theta, \mathbf{Y})$, that is some form of integration over the *Element Loss* $\ell(y_{\theta x}, y_x)$. Further, we have a space of hypothesis that satisfy the given data, the *Version Space* $\mathcal{V} = \mathcal{V}(\mathcal{H}, \mathbf{D}) \subset \mathcal{H}$. With perfect distributional knowledge the optimal point-wise Bayes predictor is then given by:

$$f^*(x) := \operatorname*{arg\,min}_{y' \in \mathcal{Y}} \int_{\mathcal{Y}} \ell(y', y) p(y \mid x) \, \mathrm{d}y.$$

Aleatoric Uncertainty: Usually, even with perfect distributional knowledge, the conditional probability, given by Bayes rule: $p(y \mid x) = \frac{p(x,y)}{p(x)}$, does not identify a single outcome y and instead only provides the probability $p(y \mid x)$ of observing y, thus uncertainty about the actual outcome y remains. This remaining uncertainty is the aleatoric uncertainty.

Structure Uncertainty: The best possible hypothesis one could arrive at with perfect distributional knowledge is:

$$h^* := \operatorname*{arg\,min}_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(x), y) p(x, y) \, \mathrm{d}y \, \mathrm{d}x.$$

Here, h^* can only be equal to $f^*(x)$ if $f^* \in \mathcal{H}$. $f^* \notin \mathcal{H}$ is the first source of structure uncertainty. Similar, for a hypothesis $H^*(x) = Y_{\theta x}$ the predictive distribution $P(Y_{\theta x})$ can only be equal to $P(Y \mid \mathrm{do}(X = x))$ if $F^*(x) \in \mathcal{H}$.

Capacity Uncertainty: Let us consider the best possible predictor h_{θ^*} with perfect distributional knowledge, which has parameters:

$$\theta^* := \operatorname*{arg\,min}_{\theta \in \Theta} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h_\theta(x), y) p(x, y) \, \mathrm{d}y \, \mathrm{d}x.$$

Here, h_{θ^*} can only be equal to $f^*(x)$ if $f^* \in \mathcal{H}_\Theta$. $f^* \notin \mathcal{H}_\Theta$ is the source of capacity uncertainty, i.e., the capacity of the parameter space Θ is too low to represent f^* even if $f^* \in \mathcal{H}$. Further, this then directly leads to the previous type of structure uncertainty. Again, for a hypothesis $H_{\theta^*}(x) = Y_{\theta^* x}$ one can not guarantee that $P(Y_{\theta x})$ is equal to $P(Y \mid \mathrm{do}(X = x))$.

Data Uncertainty: The data distribution $P(X, Y)$ is usually unknown and needs to be approximated $P_{\mathbf{D}}(X, Y)$ using the data $\mathbf{D}$. The difference between these distributions is the source of (epistemic) data uncertainty.

Parameters Uncertainty: Directly resulting from the data uncertainty is the parameters uncertainty. That is, even assuming $f^* \in \mathcal{H}_\Theta$, the resulting parameters:

$$\theta^*_{\mathbf{D}} := \arg\min_{\theta\in\Theta} \int_{\mathcal{X}\times\mathcal{Y}} \ell(h_\theta(x), y) p_{\mathbf{D}}(x, y)\, \mathrm{d}y\, \mathrm{d}x,$$

can only approximate $\theta^* \in \Theta$, leading to parameter uncertainty. Further, the learned hypothesis $h_{\theta^*_{\mathbf{D}}}$ is then unequal to $f^*(x)$ even if $f^* \in \mathcal{H}_\Theta$, leading to the second source of structure uncertainty. This holds again for a hypothesis $H_\theta(x) = Y_{\theta x}$, for which $P(Y_{\theta x})$ is then unequal to $P(Y \mid \mathrm{do}(X = x))$.

The parameter uncertainty can also be viewed from the perspective of the version space $\mathcal{V}_\Theta = \mathcal{V}(\mathcal{H}_\Theta, \mathbf{D}) \subset \mathcal{H}_\Theta$. From this perspective one expects $f^* \in \mathcal{H}_\Theta$. However, $\mathcal{V}_\Theta$ does not contain a single hypotheses $h_{\theta^*_{\mathbf{D}}}$, instead the given data allows for a whole set of possibly correct hypotheses, and one can not tell which $h_{\theta^*_{\mathbf{D}}} \in \mathcal{V}_\Theta$ is equal to f^*. Further, the version space has an equivalent in the parameter space, i.e., the space of all parameters that satisfy the data $\Theta_{\mathbf{D}} = \{\theta \in \Theta \mid h_\theta \in \mathcal{V}_\Theta\} \subset \Theta$. We can then think of a posterior distribution $P(\Theta_{\mathbf{D}})$ over random parameters $\Theta_{\mathbf{D}}$, supported on $\Theta_{\mathbf{D}}$, with density $p_{\mathbf{D}}(\theta) = p(\theta \mid \mathbf{D})$, which is the basis for the Bayesian paradigm.

Predictive Uncertainty: Finally, from all the above follows that $P(Y_{\theta x}) \neq P(Y \mid \mathrm{do}(X = x))$ in many cases, which is the reason for predictive uncertainty. Further, $\Theta_{\mathbf{D}} \neq \{\theta^*\}$ in most cases, thus $P(\Theta_{\mathbf{D}})$ is not the Dirac distribution, and the distribution over possible predictive distributions $P(P(Y_{\theta x}) \mid \Theta_{\mathbf{D}})$ has non-zero variance. This leads to a epistemic part $p_{\mathbf{D}}(\theta)$ in the Bayesian predictive density:

$$p_{\mathbf{D}}(y_{\theta x}) = \int_\Theta p(y_{\theta x}) p_{\mathbf{D}}(\theta)\, \mathrm{d}\theta.$$

2.3.3. Methods to Quantify Uncertainty

In this section we will briefly discuss methods for *Uncertainty Quantification (UQ)*, which are often closely coupled with the knowledge discovery task and the used adaptive sampling method. This is why we will have some overlap with Section 2.2.3 [Adaptive Sampling], but with a focus on the UQ rather than data selection.

Predictive Uncertainty is often estimated in a simplified form in the context of *Uncertainty Sampling*. Such metrics can not distinguish between AU and EU, and they neglect the uncertainty about model parameters. Further, for them to work correctly with adaptive sampling, AU needs to be low in comparison to EU: Assume AU is high in some regions of the data space and low in others, such metrics will lead to over exploration of high AU regions as AU can not be reduced with more data by definition, and under exploration of the low AU regions where EU could have been reduced further.

In classification tasks, one uses the prediction of a probabilistic classifier $H_\theta(x) = Y_{\theta x}$. That is, for a given input x one obtains the class membership probability $p(y_{\theta x})$ for each class $y_{\theta x} \in \mathcal{Y}$. If all classes are similarly likely, this hints that the classifier is uncertain about its prediction. To measure the similarity between classes (and thereby the

uncertainty) common metrics are [WS14; Set09]: The maximum class probability [LG94], which is lower (and thereby more uncertain) if other classes also have a high probability; The difference between the two highest class probabilities [SDW01], which is lower (and thereby more uncertain) if one other class has a similar probability; The entropy of all class probabilities [Set09], which is higher (and thereby more uncertain) if other classes have similar probability.

While these metrics are easy to compute for a finite number of classes, computing them for regression tasks is often intractable. However, for regression models which provide a gaussian predictive distribution $P(Y_{\theta x} \mid \mathrm{do}(X = x))$ with density $p(y_{\theta x}) = \mathcal{N}(\bar{y}_{\theta x}, \sigma_{\theta x})$ one can use the prediction variance $\sigma_{\theta x}$. Here, prediction entropy is a monotonic function of $\sigma_{\theta x}$, making both equivalent uncertainty measures that are easy to compute [Set09]. An example for such regression models is Gaussian Process Regression (i.e. Kringing) [Kan+18]. The variance of such models is also often used to derive an upper bound in the context of UCB sampling.

A heuristic to measure predictive uncertainty for a set of given (unlabeled) inputs $\mathbf{X}$, originally developed for the Support Vector Machine (SVM) [TK01], uses the distance to the decision boundary. The intuition is, the closer the input is to the decision boundary the likelier it is on the wrong side of the boundary and the higher the uncertainty.

For surrogate modeling the impact of the input uncertainties (part of AU) on the prediction is often relevant for sensitivity analysis. To this end, *Polynomial Chaos Expansion* is applied [App+17; Müh+17], representing the model response as a series expansion.

Data Uncertainty base methods focus on EU and often neglect AU. In consequence, they often need to be combined with AU focused methods. Purely EU focused methods often consider the density of the training data. A simple heuristic uses the distance between training data $\mathbf{D}$ and the decision boundary. A larger distance hints more uncertainty about the position of the decision boundary. Other methods [TLR02] perform density-based clustering on a given unlabeled input set $\mathbf{X}$, propagating given class labels through the cluster to estimate how likely a data point belongs to a cluster. That is, data points in dense regions of a cluster likely belong to the cluster, while data points between clusters are uncertain, and data points far away of clusters are possibly outliers and uninformative.

To include AU weighted combinations with uncertainty sampling are used. *Information Density* [SC08] for example considers the density of a given unlabeled input set $\mathbf{X}$ and prioritizes uncertain inputs in dense regions of the input space over similar uncertain inputs in less dense regions. Similarly, *K-Nearest-Neighbor-based Density* [Zhu+10] trades off between high uncertainty estimates and high local density estimates. In essence such methods boil down to a trade off between maximizing the distance to labeled data (less density), and minimizing the distance to unlabeled data (higher density) [Fuj+98].

It is easy to see, that such metrics based on a input set $\mathbf{X}$ are only meaningful if the distribution of $\mathbf{X}$ has any meaning, i.e., $\mathbf{X}$ is observed rather than arbitrarily selected. Further $\mathcal{X}$ needs to be of low to medium dimension, because density becomes meaningless in a higher dimensional space due to the curse of dimensionality.

Parameter Uncertainty based methods are constructed in a way to only capture EU [HW21] and work for arbitrary inputs x. While estimating this type of uncertainty relies on the parameter uncertainty it is commonly expressed as 'The part of the predictive uncertainty that is caused by the parameter uncertainty'. We note that while tailored to EU, AU can still be estimated with orthogonal methods.

Expected Model Change [SCR07] directly uses the gradient of the loss $\nabla_\theta \ell(y_{\theta x}, y_x)$ with respect to the model parameters θ for a fixed input x and across all possible predictions $y_{\theta x} \in \mathcal{Y}$ weighted by their probability $p(y_{\theta x})$. Thus, it quantifies how sensitive a prediction is to parameter changes, the intuition being that one can be less certain about a prediction if it is highly sensitive.

QBC also bases on parameter uncertainty. QBC treats uncertainty as the number of opposing hypotheses that are still consistent with the labeled data, i.e., the size of the version space $\mathcal{V}_\Theta = \mathcal{V}(\mathcal{H}_\Theta, \mathbf{D})$ [SOS92]. By training an ensemble of M models, we obtain an empirical parameter set $\hat{\Theta}_\mathbf{D} = \{\theta_m\}_{m=1}^M$ with individual model parameters θ_m. One can interpret $\hat{\Theta}_\mathbf{D}$ as a sample from the posterior $P(\Theta_\mathbf{D})$ or as an empiric estimate of $\Theta_\mathbf{D} = \{\theta \in \Theta \mid H_\theta \in \mathcal{V}_\Theta\}$. Instead of using the difference in parameters directly, one then uses the ensemble to perform predictions. This results in a set of predictive distributions:

$$\mathbf{P}_{\hat{\Theta}_\mathbf{D}, x} = \{P(Y_{\theta_m x})\}_{m=1}^M \text{ which empirically estimates } P(P(Y_{\theta x}) \mid \Theta_\mathbf{D}).$$

As measure of uncertainty the disagreement among the models is used, which can be measured in various ways: *The Vote Entropy* [AD99], i.e., the entropy of predictions between ensemble members; *The Average Kullback-Leibler Divergence* [MN98], i.e., the difference of the ensemble consensus to the individual predictions; Metrics used for the *Simple Predictive Uncertainty* previously discussed, by hard pooling the ensemble predictions to estimate the hard class membership probability $p(y_{\Theta_\mathbf{D} x}) = p(\arg\max_{y_{\theta x} \in \mathcal{Y}} p(y_{\theta x}) \mid \Theta_\mathbf{D})$ [KW06]. Similarly, for regression tasks, one can estimate $p(y_{\Theta_\mathbf{D} x})$, measuring, for example, disagreement as the variance which is comparable to vote entropy.

However the ensemble size of QBC is often computationally restricted. While small ensembles are still accurate enough for data selection [Set12], they are not precise enough for accurate uncertainty estimation. In such cases one can use more advanced methods to estimate $p(y_{\Theta_\mathbf{D} x})$ empirically. Examples include: Bayesian Ensembling [PLB20],Concrete Dropout [GG16b; GG16a; GHK17], Drop Connect [Mob+21], Bayesian Neural Networks [MKH18], and Variational Inference [Gra11]. Using such methods then slowly transitions from QBC like methods to empiric Bayesian inference. While such methods are computational more efficient, and often provide more robust uncertainty estimation, they often still suffer from imprecise uncertainty calibration [BML25]. Thus requiring additional uncertainty calibration steps when used for risk estimation and graceful failure, i.e., the model signaling when its prediction is likely to be wrong.

Structure and Capacity Uncertainty is often relevant in the context of surrogate modeling. Here, the uncertainty is then measured using *Multi-Fidelity Discrepancy Modeling* [Zha+22; HT26]. That is a special case of an ensemble with only two models, one low-fidelity (cheap) and high-fidelity (expensive) model. The discrepancy between the models is then used as uncertainty.

Global Uncertainty focuses on estimating not only uncertainty based on distributional properties of $P(Y_{\theta x} \mid \mathrm{do}(X = x))$, but instead uses properties of $P(Y_\theta \mid X)$. That is, it calculates how a selected input x would change the uncertainty across the whole input space $\mathcal{X}$. Examples include: Expected Error; Expected Log-Loss which is equivalent to expected entropy, expected information or mutual information; For details see [Set09]. The main benefit of such strategies is that, if used for adaptive sampling, they gain robustness against sampling outliers. That is, robustness against sampling data points that are uncertain because they are at the border of the data distribution, but do not yield much information about the data distribution. However, such methods often estimate $P(Y_\theta \mid X)$ by integrating over $P(Y_{\theta x} \mid \mathrm{do}(X = x))$, which is in many cases computationally expensive [Zhu+10], only possible on lower dimensional data, or can only be done approximately. Similarly methods such as (global) Variance Reduction [Coh93; CGJ96], not only consider the uncertainty at a single input x, but how observing a single input-output pair (x, y_x) would influence the variance across all possible inputs X, leading to the same computational deficiencies.

Bayesian Uncertainty combines AU and EU focused methods in a statistically founded way. Specifically, it computes predictive uncertainty by marginalizing over the parameter posterior (the version space):

$$p_{\mathbf{D}}(y_{\theta x}) = \int_{\Theta} p(y_{\theta x}) p_{\mathbf{D}}(\theta)\, \mathrm{d}\theta \approx \frac{1}{M} \sum_{m=1}^{M} p(y_{\theta_m x}) \text{ where } \theta_m \sim P(\Theta_{\mathbf{D}}) \tag{2.2}$$

In contrast to QBC, which frequently employs *hard pooling* to identify the most likely label per hypothesis, empiric Bayesian methods utilize *soft pooling*. Soft pooling preserves the full predictive distribution $P(\Theta_{\mathbf{D}})$ of each ensemble member, effectively averaging the aleatoric components. Analytic approaches typically leverage Gaussian Process Regression or other domain-specific probabilistic models to quantify these uncertainties [BW22]. The total predictive uncertainty can then be decomposed into aleatoric and epistemic parts using information-theoretic measures like *Mutual Information*. Mutual Information quantifies the difference between the entropy of the soft-pooled average prediction (Total Uncertainty) and the average entropy of the individual predictions (AU), thereby isolating the epistemic part caused by $\Theta_{\mathbf{D}}$. However, there has been some critique of using entropy decomposition [Wim+23]. That is, AU can only be estimated using the data, thereby this estimate depends on the EU, such that the two measures are not independent of each other.

2.4. Adaptive Knowledge Discovery Tasks

This section will introduce diverse Knowledge Discovery Tasks relevant to AKD. It will highlight the characteristics a task must possess to apply AKD, as well as inherent constrains of AKD. We will introduce well-researched tasks, tasks where AKD is relatively new, tasks that may benefit from AKD, and, finally, tasks that are related but for which AKD is not directly applicable. We will also discuss connections between different tasks. The list of tasks we provide in this section will not be exhaustive; it is intended to give the reader a sense for which tasks AKD becomes relevant and which are out of scope.

2.4.1. Characteristics and Constrains of AKD Tasks

To perform AKD first of all we require a DGP with which we can interact. That is we can select input values for Intervention Variables, and observe the Outcome Variables. Secondly, intervention and / or observation of the DGP is expensive, see Section 2.1.2. Hence, the observation budget is limited and it is impossible to perform an exhaustive search over the entire DGP domain before exhausting the budget [Hu+25]. Thirdly, the DGP is low- to moderate dimensional or can be transformed into lower dimensional space. That is because uncertainty about unobserved data is related to the distance to the nearest observation, and in higher-dimensional space, the concept of distance becomes relativized by the curse of dimensionality [HS12; Hu+25]. Fourthly, a common challenge is that observations of the DGP are inherently noisy due to hidden variables, thus there is no true value of a DGP at a given value x, only a expected value $\mu(x) = \mathbb{E}[C(x)]$. In consequence, one has to rely, for example, on the average $\bar{y}_x$ over multiple observations y_{xi}, and keep in mind the intrinsic uncertainty [FPD09] (AU). Finally, the derivative of the DGP is unavailable or impractical to estimate, rendering classical gradient-based methods inapplicable [Hu+25]. Even so, one has to keep in mind these preconditions, there are still a vast amount of applications for AKD, specifically in experimental science and engendering. We will give examples in the following Section 2.4.2 describing specific knowledge discovery tasks.

When performing AKD one should also keep in mind that the main benefit of AKD is also its main drawback: AKD selects data points specifically for the given knowledge discovery task, thereby achieving its gain in data-efficiency. In consequence, using the data for a different task is only possible if the new task requires a subset of the data the original task required. Any other task would require the collection of additional data, now tailored to the new task. Such additional data collection is only possible if one has longtime access to the DGP. If this is not the case, because the DGP can not be reproduced or is some sort of one time event, we advise to consider if one wants to spend additional data budget (not using AKD) to make sure the collected data can be used task independently, considering future applications for which the data may become relevant. In experimental sciences, this is seldom a problem because a good experiment should always be reproducible, making the DGP long term accessible. Similarly in industry and engineering either the same (e.g. production) process is still running, enabling additional data collection, or the process has changed significantly, making the old data obsolete and requiring new data collection any way.

2.4.2. Introduction to AKD Tasks

The applications for AKD are extensive. This section focuses on the paradigms most relevant to this work or those that maintain significant methodological overlap with AKD principles. A core distinction between these tasks lies in their objective: while some seek to reduce global uncertainty across the entire domain, others prioritize local targets, such as finding an extrema or a specific boundary.

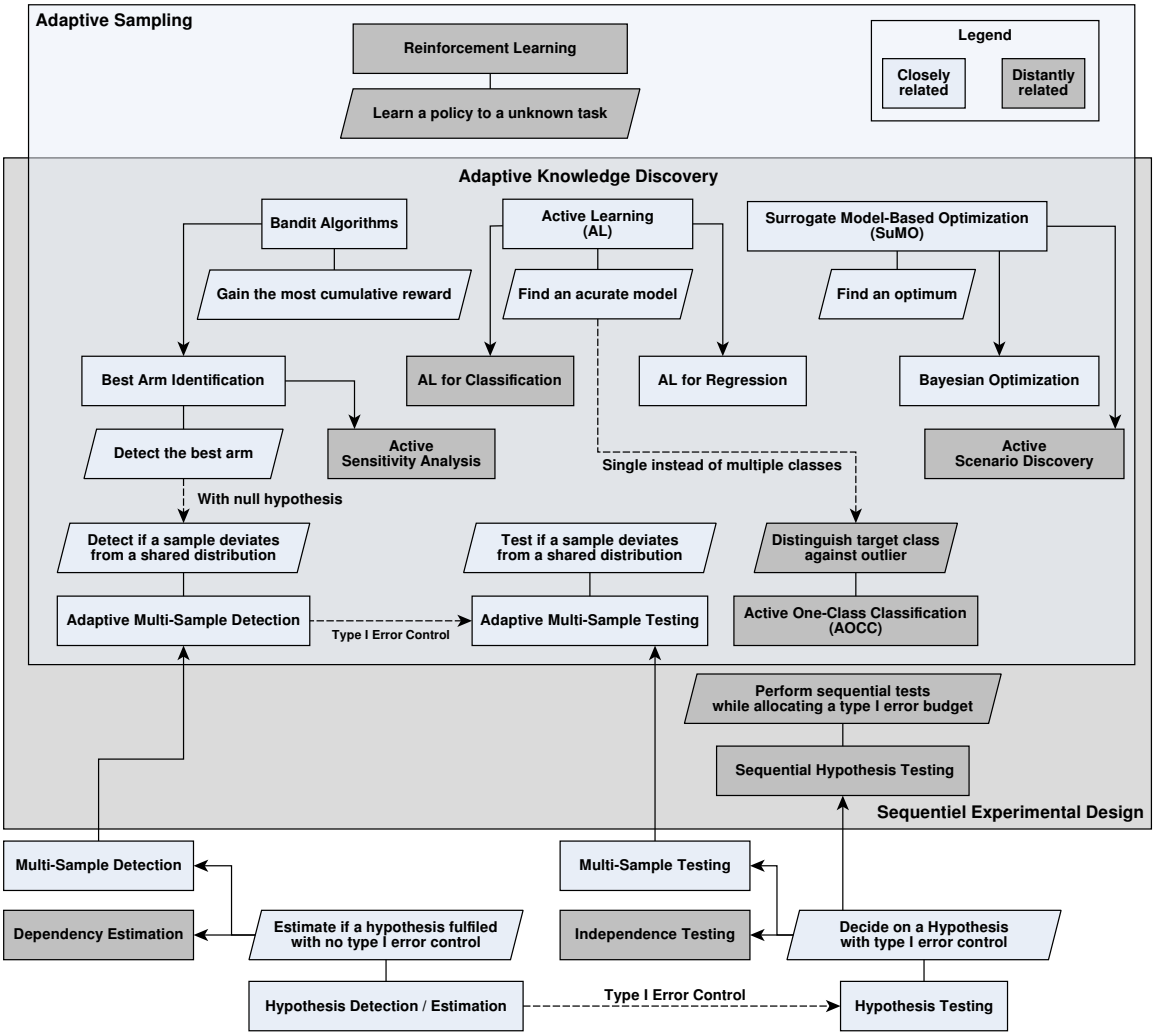


Figure 2.5.: Overview and interconnection of AKD tasks.

Active Learning (AL) [Set12] aims to build the most accurate global model of the DGP with the fewest possible samples. This is a global objective task where the goal is to minimize model uncertainty across the entire input space.

AL is often categorized into *pool-based* sampling (selecting from a predefined set of unlabeled data) and *query synthesis* (generating new data points), this work focuses on the latter. Query synthesis is particularly relevant in the experimental sciences and engineering, where new observations are actively generated and the distribution of inputs is not always known beforehand [Wan+25]. Furthermore, while AL often addresses classification, we focus on the more complex and less researched regression scenario [WS25; RJZ17; WLH19]. Regression-based query synthesis is more challenging because the search space is continuous, enabling for infinitely many possible outcomes [Hol+23].

Example: In material science, AL can be used to map the hardness of a new alloy across all possible mixing ratios of three metals.

Active One-class Classification (AOCC) [EB20; Eng+20; Jia+19] or active outlier detection, aims to find the bounds of a single target class against outliers. While algorithmically related to pool-based AL, Active One-class Classification (AOCC) relies heavily on the underlying input data distribution. Because the data is typically highly unbalanced, AKD strategies must be adapted to explore the tails of the distribution where outliers or rare class members reside.

Example: In structural health monitoring, AOCC identifies the specific vibration patterns of a 'healthy' bridge. Because 'failure' data is rare, the algorithm must adaptively sample the boundaries of the known 'healthy' distribution to improve detection sensitivity.

Surrogate Model-based Optimization (SuMO) [ABT21; Loo+19] seeks to find the single best input x that maximizes or minimizes a objective function computed on the output of a DGP. A prominent special case is *Bayesian Optimization.* In contrast to AL, SuMO is a *local objective task*. It ignores high-uncertainty regions if the surrogate model indicates they are unlikely to contain the global optimum, focusing instead on the exploration-exploitation trade-off near potential extrema.

Example: Optimizing the shape of a turbine blade to maximize lift; the algorithm ignores shapes that are clearly inefficient to focus on refining the most aerodynamic designs.

Active Scenario Discovery [AB21] involves identifying 'interesting' scenarios or critical regions within the input space. Rather than optimizing a single value, this task focuses on characterizing specific boundaries where the DGP exhibits significant behavior changes or enters a failure state. As such it is related to the task of robust optimization.

Example: In autonomous driving, this is used to find "corner cases" – specific combinations of weather, lighting, and traffic that lead to system failure – rather than modeling every mundane driving situation.

Bandit Algorithms [LS20] address the problem of maximizing a cumulative reward by interacting with an unknown DGP through a set of discrete action choices, traditionally referred to as 'arms' (e.g., choosing between three distinct catalyst types).

A critical variant for AKD is *Best Arm Identification (BAI)* [Jam+14]. While standard bandits avoid losing reward during the selection process, BAI has the objective of detecting the single best arm with high statistical confidence. BAI is often described as focusing purely on exploration (since cumulative reward is not the goal), however, it still faces an exploration-exploitation dilemma regarding budget allocation: The algorithm must choose whether to 'exploit' a promising arm by collecting more data to confirm its optimality or 'explore' other arms to ensure a better candidate has not been overlooked [LS20]. Like SuMO, BAI is a local objective task; once a specific arm is statistically proven to be non-optimal, the algorithm stops collecting data for it, as it does not require a high-fidelity model of the entire DGP to satisfy its goal.

Example: In a clinical trial for a new treatment, a researcher uses BAI to determine which of five specific drug dosages is most effective. The goal is not to maximize the health of the current trial participants, but to identify the best dosage with high certainty for future approval.

Active Sensitivity Analysis [BW22; IL15; Sal08] quantifies how much the output changes when a specific input is varied, identifying which variables 'matter' most to the DGP. This is closely related to BAI, as it effectively treats inputs as arms to determine which ones dominate the system's variance. However, in contrast to BAI, it can also include selecting specific values for the selected input to characterize the local or global derivative of the response surface.

Example: In chemical manufacturing, determining whether temperature or pressure has a greater impact on final product purity. This allows engineers to prioritize which sensors or control systems require the highest precision.

Adaptive Hypothesis Detection and Estimation determines if a hypothesis is fulfilled (detection) or the degree to which it is fulfilled (estimation). *Adaptive Multi-Sample Detection* [MN14] identifies if at least one sample deviates from a shared distribution. This is related to BAI, but the focus is on the existence of a deviation rather than identifying the best performing input. *Distilled Sensing* [HCN11], for example, detects or localizes if one of many noisy sample contains a sparse signal.

Example: A biologist might adaptively sample different cell cultures to detect the presence of a rare mutation.

Adaptive Hypothesis Testing [Rob52] involves deciding on a hypothesis while maintaining Type I error control with data coming in sequentially. Which requires to account for multiple testing [How+21]. *Sequential Hypothesis Testing* [JM15; FS08] allows for early stopping as data arrives, but it typically lacks the adaptive interventions found in AKD. In AKD, the choice of the next intervention value x is based on previous results [CN08], which inherently breaks the Independent and Identically Distributed (i.i.d.) assumptions of classical frequentist statistics. Because these interventions couple past results with future sampling locations, maintaining strict statistical guarantees is difficult. Consequently, AKD practitioners often favor *Adaptive Hypothesis Detection*, sacrificing formal Type I error control for improved discovery performance.

Reinforcement Learning (RL) [SB18b] seeks to learn an optimal policy for an unknown task. While Reinforcement Learning (RL) shares the adaptive nature of AKD, it generally assumes the task itself is unknown and requires a massive volume of data. In the AKD context, the high cost of observation makes the sample-intensive nature of classical RL largely impractical.

Part III.

Drifting Data Generating Processes

3. DEAL: Data Efficient Active Learning for regression under drift

The content of this chapter bases on the following publication:

- Béla H. Böhnke, Edouard Fouché and Klemens Böhm. 'DEAL: Data-Efficient Active Learning for Regression Under Drift'. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by De-Nian Yang et al. Singapore: Springer Nature, 2024, pp. 188–200. DOI: 10.1007/978-981-97-2266-2_15

Keywords: Concept Drift; Active Learning; Regression; Stream Learning

3.1. Chapter Overview

A key element of Adaptive Knowledge Discovery (AKD) is the interaction with the Data-Generating Process (DGP). However this interaction becomes more intricate if the DGP is drifting. That is, the relation $Y_X = C_t(X)$ (in this context often named concept) of the underlying DGP changes over time t ($C_{t_1}(X) \neq C_{t_2}(X)$), leading to Concept Drift (CD). Thus, in this chapter, we will have a closer look at drifting DGPs and the implications on AKD, specifically for the task of Stream-based Active Learning (SAL).

There is no doubt that such drift poses a significant challenge for learning a statistical model, requiring frequent recalibration to maintain estimation error below a user-required threshold. While there already exist solutions for recalibrating such model in general [Zli10; Lu+19; IP19; Gam+14], they mostly assume that data is cheap and comes in the form of an infinite stream.

This assumption, however, does not hold in an AKD context where observing the output variable comes at a high cost and thus makes continuous monitoring impractical. More specifically, we require methods that can adjust for drift without continuously monitoring the drifting variables, balancing the goal of reducing expensive measurements and the need for additional measurements for drift detection and model recalibration. While such drift-correcting Active Learning (AL) methods exist for classification tasks, many real-world scenarios involve continuous target variables requiring regression models. These regression scenarios lack adequate AL methods [RJZ17; WLH19], highlighting a gap in current AL research.

Finding a solution to this gap brings multiple challenges:
(1) The first challenge follows directly from the goal of reducing expensive measurements. That is, estimating the error due to drift without actually continuously monitoring the

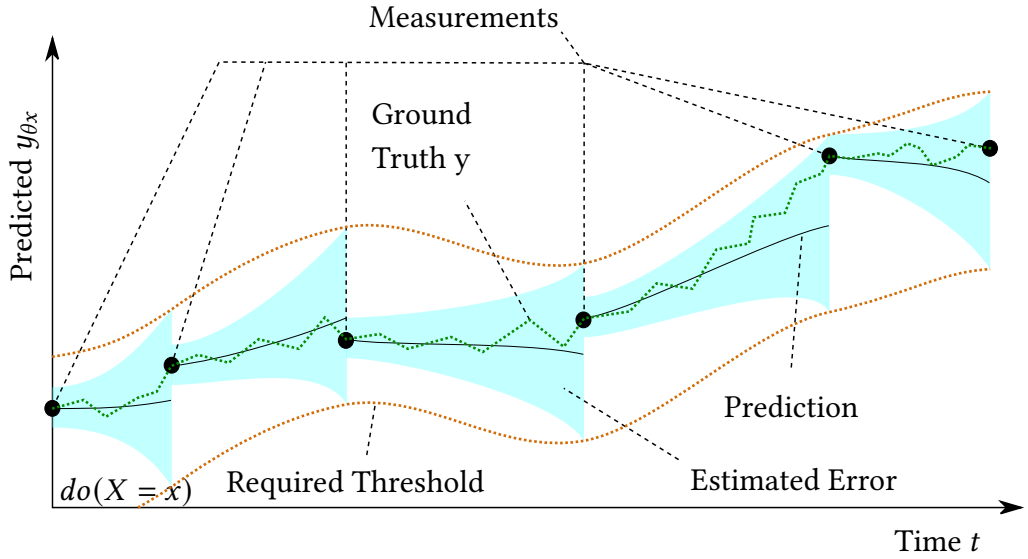


Figure 3.1.: DEAL estimates its error and measures once the error reaches the user-required threshold. The input value x is kept constant ($do(X = x)$) for better visualization, the target y_x still varies due to frequent incremental drift.

output variable.
(2) The second challenge becomes clear when comparing classification and regression tasks under drift. Classification only has to monitor a fixed number of class distributions over time. In contrast, regression has to estimate the model error due to drift for each point in a continuous input space, target space, and time.
(3) Because continuously monitoring the output variable is impractical, the final challenge is an appropriate selection of the measurement time based on the estimated error, ensuring that the actual error due to drift remains below a user-required threshold while the number of measurements is minimal.

We contribute by proposing Data Efficient Active Learning (DEAL), the first AL regression method that learns data efficiently under oracles that exhibit frequent drift while keeping the estimation error below a user-required threshold. DEAL estimates its own error due to drift, using a Gaussian Process (GP) model with a time-variant kernel. The GP learns the *drift behavior*, i.e., the time-dependent distribution of the target variable. Figure 3.1 shows that DEAL queries the oracle whenever the estimated error exceeds a user-required threshold. DEAL uses the new data to recalibrate, and combines it with the old data to update the learned drift behavior. In this way, DEAL minimizes the number of measurements required for drift correction by dynamically adjusting the sampling rate to maintain the user-required threshold. Consequently, DEAL prioritizes data acquisition during the initialization phase or during abrupt changes in drift, while significantly reducing the sampling rate for well-learned drift behavior.

We evaluate DEAL against multiple baselines, on multiple drift-affected time series, and provide the code[1] for reproduction.

3.2. Related Work

To our knowledge, existing stream-based AL methods for regression (e.g., [BST17; Iwa22; KSF17; Tur+19; SB18a; RJZ17; WLH19; YK19]) do not consider drift, i.e., the oracle is assumed to be static. Such methods stop learning once the regression model performs well and never start learning again, even if the model performance decreases due to drift. We show this undesirable behavior in our experiments.

There exist AL methods that consider drift [Zli+14; Zli+11b; Zli+11a; Zha+18; Sha+19a; Liu+21; Liu+23; PK16; MSB16; KW11; KW12; KPW18; KC19], but they are restricted to classification. Further, to apply [KPW18; Liu+23; MSB16; PK16; Zli+11a; Zli+14] a user needs to set a measurement budget and additional parameters in advance without knowing drift behavior, and thus without knowing the resulting estimation error. Improper set parameters lead to either too costly or too inaccurate models. Drift behavior that changes causes the same problem because the methods cannot adapt. Further, those methods primarily monitor the distribution of input variables per class. This is intractable for regression and impedes the transfer to the regression case.

There exist change detection methods that can adapt to changing drift behavior. For example, Active and Adaptive Incremental Learning (AAIL) [PK16] detects changes in the input variables and measures the target variable at each change. While the authors claim that AAIL adapts to Concept Drift (CD), it can only adapt to covariance shift. Covariance shift only refers to changes in the distribution of the input variables, which is cheap to observe, while CD describes a change in the relationship between the input and target variables. AAIL can be adapted to regression tasks, and we include an adaptation in our experiments. Drift detection approaches not designed for AL, as surveyed in [Zli10; Lu+19; IP19; Gam+14], typically assume that the target variables are cheap to observe, which violates a basic assumption of AL. Thus, such approaches require additional methods for strategic under-sampling of the target variable, essentially what the existing AL techniques for drifting oracles do.

In summary, the only methods viable in practice to deal with drift in AL regression is some form of under-sampling with consecutive measurements at a user-defined frequency, as in [Car+18; Car+19; Don+09]. A poor frequency choice leads to missed drift or higher measurement costs. We use such a method as one of several baselines.

3.3. Problem Statement

Active Learning (AL) refers to data collection methods aimed at estimating the relationship between input X and target Y variables under the assumption that input values x can be chosen (intervene up on X by $do(X = x)$) but measurements y_x of the target variable Y is expensive.

[1] https://github.com/bela127/alsbts-experiments

In contrast, we will investigate the setting of Stream-based Active Learning (SAL) [Set12]. SAL assumes input variables arrive as a continuous stream $X(\mathbb{T})$ over a time horizon $\mathbb{T}$, e.g., sensor data like temperature and humidity acquired for real-time environmental monitoring. A *learner* observes the stream $X(t)$ at time t, resulting in one *potential query* x_t. In contrast to AL, the learner can not decide on the value x_t, instead the learner can only decide whether to perform a *query*, i.e., to obtain a measurement y_t of the target variable Y at time t – like soil quality for agriculture at a specific time of the year – or not. While the cost of observing the stream X is negligible, measuring the target is expensive. Many scenarios share this assumption [Set12], such as industrial process control, resource allocation for environmental monitoring, or demand forecasting for energy management [Car+18].

3.3.1. Formalization

Stochastic Processes S or random processes are random functions $S : t \mapsto S(t)$ where $t \in \mathbb{T}$ is interpreted as time with domain $\mathbb{T} = \mathbb{R}_+$ [C05].

Data Streams X are treated as stochastic processes which we can observes at time t, resulting in one *potential query* x_t.
Covariance Shift is present if the distribution of the stream changes over time, that is $X(t_1) \neq X(t_2)$.

Observed Data Streams or time series are sample functions $x(\mathbb{T}) \leftarrow X(\mathbb{T})$. Here, $x(t) = x_t$ provides a distinct value x_t, drawn from $X(t)$ at time t.

Brownian Motions B are stochastic processes $B : t \mapsto B(t)$ [C05] characterized by random increments $\delta B(\delta t) = B(t + \delta t) - B(t) \forall t \in \mathbb{T}$, where these increments follow the normal distribution $\delta B(\delta t) \sim \mathcal{N}(0, \delta t)$.

A Gaussian Process (GP) is a random function defined as $\mathrm{GP}(x) \sim \mathcal{N}(m(x), v(x))$, illustrating that the random function $\mathrm{GP}(x)$ comes from a normal distribution with mean $m(x)$ and variance $v(x)$ both depending on x [LRS13]. One often uses GPs as a probabilistic prior over functions.

Kernel Functions $k(x, x')$, also called covariance functions, can define a GP instead of using a mean and variance function.
The Radial Basis Function (RBF) Kernel $k(x, x') = v \cdot \exp\left(-\frac{(x-x')^2}{2 \cdot l^2}\right)$, has two parameters: $v \in \mathbb{R}_+$ (target variance) scaling the target of the random function, and l (length scale) determining function smoothness.
The Brownian (Bridge) Kernel $k(t, t') = v_b \cdot \min(t, t')$, with points in time t, t' has a variance parameter $v_b \in \mathbb{R}_+$. A GP with a Brownian kernel results in a Brownian motion B, here variance parameter v_b translates to the drift speed, i.e., it scales $\delta B(\delta t)$ by v_b [LRS13]. One

often combines kernels to form a new kernel, which is then used to define a stochastic process. To streamline the notation of kernel composition we use the following notation:

$$\underbrace{\mathrm{GP}(x,t) = A(x) \oplus [B(x \mid \theta) \otimes C(x)]}_{\text{Process Composition}} \quad \equiv \quad \underbrace{k_{\mathrm{GP}}(\mathbf{z},\mathbf{z}') = k_A(x,x') + k_B(x,x' \mid \theta)k_C(t,t')}_{\text{Kernel Evaluation}}, \tag{3.1}$$

with $\mathbf{z} = (x,t)$ and $\mathbf{z}' = (x',t')$, where '|' signifies that θ is a kernel hyper parameter, which is given (learned beforehand) at evaluation time.

Concepts define the DGP, that is, the relations between input X and target Y variables. They are random functions $C : X \mapsto C(X) = Y_X$. AL methods often omit details about the DGP, and call this abstraction *oracle*, this is equivalent to a concept. In practice, such oracles can *drift*: Evolving, unobservable environmental parameters can affect the relationship between input and target variables over time. If a concept depends on time, it is represented as $C : (X,t) \mapsto C(X,t)$. This phenomenon of changing relationships between variables is known as *Concept Drift (CD)*. By definition, CD is present if $\exists t_1, t_2 : C(x,t_1) \neq C(x,t_2)$, where $t_1 \neq t_2$ are two points in time [Gam+14].

Queries $\mathrm{do}(X = x(t))$ are driven by the need to distinguish between the actual (unobserved) DGP output $C(X(t)) = Y_X$, and a actively measured value $C(x(t)) = y_t$. In most cases this distinction is clear from context.

Stream-based Active Learning (SAL) [Set12] iteratively observes the value x_t of the stream $X(t)$ and only performs a query if the uncertainty v_θ of a prediction $y_{\theta t}$ exceeds a certain user-required threshold v_{target}. The learner then recalibrates the model using the resultant measurement y_t. This is known as *uncertainty sampling*, which is a typical way to decide whether to query or not.

Algorithm 3.1 The Common vs Our Adapted SAL Cycle

```
1:  procedure COMMON(v_target, MODEL)
2:      while running do
3:          x_t ← X(t)
4:          y_θt :=MODEL.ESTIMATE(x_t)
5:                          ▷ y_θt only for evaluation.
6:          v_θ := MODEL.VARIANCE(x_t)
7:          if v_θ >= v_target then
8:                          ▷ Uncertainty sampling.
9:              y_t := ORACLE.QUERY(x_t)
10:             DATAPOOL.ADD(x_t, y_t)
11:             MODEL.TRAIN(DATAPOOL)
```

```
1:  procedure ADAPTED(v_target, MODEL)
2:      while running do
3:          x_t, t ← X(t),TIME()
4:          y_θt := MODEL.ESTIMATE(x_t, t)
5:                          ▷ y_θt only for evaluation.
6:          v_θ := MODEL.VARIANCE(x_t, t)
7:          if v_θ >= v_target then
8:                          ▷ Uncertainty sampling.
9:              y_t := ORACLE.QUERY(x_t, t)
10:             DATAPOOL.ADD((x_t, t), y_t)
11:             MODEL.TRAIN(DATAPOOL)
```

The research question is:
How to select the measurements and update a learned model to account for the additional uncertainty due to drift?

3.4. Our Method: DEAL

3.4.1. The Adapted SAL Cycle

We slightly modify the Stream-based active learning cycle (see Algorithm 3.1), so that in addition to observing the stream $X(t)$, we also observe the current time t. DEAL's estimation model then takes the time t into account when estimating its variance v_θ to include the additional uncertainty caused by drift (Lines 3 to 6). Whenever the uncertainty reaches the threshold, DEAL recalibrates (Line 9 to 11) with a new data point $((x_t, t), y_t)$. Here, the drifting oracle provides the measurement y_t.

To estimate the uncertainty caused by drift, we require one assumption about the drift behavior: We assume that drift occurs frequently, i.e., within a time series of length t_{end}, at least N_c changes occur. Here N_c needs to be large enough to learn sufficient statistics of the drift behavior.

3.4.2. Our Drift-Aware Estimation Model

In contrast to methods like discussed in [KPW18; Liu+23; MSB16; Zli+14], which perform measurements without modeling the drift behavior, we are the first to learn statistics about the drift behavior. These statistics enable us to measure in adaptive time intervals in which drift of a certain magnitude may occur. We derive the statistics from a Gaussian process model (GP) according to the following prior:

Definition 3.1 (The Brownian drift prior)**.** *The Brownian drift prior is given by*

$$H(x,t) = I(x) \oplus W(x) \otimes B(t),$$

with a Brownian motion prior $B(t)$ as drift behavior, and random function priors $I(x)$ and $W(x)$ independent of any drift thereby constant over t. Here, $W(x)$ is a weighting term that defines the impact of $B(t)$ on $H(x,t)$ at a position x.

The intuition behind using a Brownian prior $B(t)$ for the drift behavior is that the combination of many small, random, and independent external influences results in a combined Brownian overall drift. Further, this model can capture drift with larger changes after a random time, as long as changes occur frequently. Such frequent drift is common in practice, like: (1) Drift due to displacement of machine elements caused by vibration. (2) Drift in large networks, such as the electrical grid, where nodes can (dis)connect from the network at random times.

We instantiate a GP with a kernel composition according to the three components $I(x)$, $W(x)$, $B(t)$, from Definition 3.1, i.e., one kernel per component. We model the drift behavior $B(t)$ with the Brownian kernel [LRS13]. To model $I(x)$ and $W(x)$, we use distinct Radial Basis Function (RBF) kernels, because of their universal approximation property [MXZ06]. In general, the choice of kernels is a parameter that one can easily tune to match prior knowledge about the data or drift. For the sake of generality, we stick to our choice in this study. Further, we enforce that both RBF kernels have the same length scale and variance, which reduces the number of learnable parameters and makes learning more

stable. This reduction assumes similar smoothness of $I(x)$ and $W(x)$ and the resulting learned function. Thus, DEAL's learnable parameters are Brownian variance v_b, RBF variance v_r, RBF length scale l_r.

The only hyperparameter DEAL takes is a user-required threshold v_{target}. DEAL trains according to Algorithm 3.1. Every time the variance v_θ estimated by DEAL becomes greater than v_{target}, DEAL measures the target value (Line 9) and adds it to the training set. DEAL then estimates the most likely kernel parameters with a gradient-based maximum likelihood optimizer (Line 11). We use the standard optimizer configuration with 5 restarts, 4 times with random kernel parameters, and once with the most likely kernel parameters from the previous iteration. In the first iteration, we initialize the kernel parameters v_b, v_r, and l_r with random values and use an initial training set of measurements from the first 10 time intervals. We use this initialization for each baseline as well.

Model Complexity of Implementation: In AL, training complexity tends to be neglected, because the oracle usually is much more expensive than the active learning decision-making. We use the standard GP model from GPy [GPy12], with a complexity of $O(N^2)$, where N is the training set size. Since N is kept small in an AL setting, the actual runtime is consistently low. We note that one can reduce the complexity of DEAL down to $O(N \cdot E)$ (with learning epochs $E \ll N$), by using gradient-based GP models and batch training. Further, one can cap the number of data points used for training at a fixed size by only using a sliding window of the dataset $\mathbf{D}$, reducing the factor N to a constant. We ran preliminary experiments with a fixed window size of 100 data points showing only minor losses in accuracy.

3.5. Experimental Design

In the following, we describe the design and the components of our computer experiments.

3.5.1. Baselines

We consider the following baselines as competitors for DEAL:

Consecutive Measurement (CM): This approach carries out measurements at regular user-specified time intervals of size δt_{meas}. For this approach δt_{meas} is a sensitive hyper-parameter which is not automatically adapted to the data-generating process and has to be tuned by the user. In our experiments, we evaluated values of $\delta t_{meas} \in [1, 20]$, with logarithmic increments.

Classic Active Learning (CAL): This standard AL approach with uncertainty sampling yields an estimate and an estimation variance. It uses a Gaussian process (GP) with an RBF kernel and kernel parameters length scale l and variance v. Unlike Consecutive Measurement (CM), the GP can automatically adjust these parameters using maximum likelihood estimation in the same way DEAL does. But unlike DEAL, this approach does not model the data behavior over time, i.e., it assumes a static oracle. To obtain a strong baseline, we initialized the GP with kernel parameters identical to the parameters of the

ground truth data, see 3.5.2. Given enough data, the Root Mean Squared Error (RMSE) of such an approach approaches zero on a stream with no noise or drift.

Change Ideal (CI): This approach is an adaptation of AAIL [PK16], which uses change detection on the input stream X. Whenever Change Ideal (CI) detects a change in X, it measures y_t. To make our evaluation independent of any specific detector and obtain a strong baseline, we simulate an 'ideal' detector. It uses the (normally unobservable) ground truth to correctly and immediately report any change in X. This baseline has no configurable parameters.

Change Error (CE): To study the effect and impact a real change detector would have on CI we also implement a imperfect 'pseudo' change detector. There are three types of errors in change detectors: undetected change, detection without an actual change, and delayed detection. To study their effect, our 'pseudo' change detector lets us control each error type with three parameters: p_{wrong} is the proportion of spurious detection at any time. p_{miss} is the chance of discarding a correct detection. std_{offset} is the standard deviation of a normal distribution. For any detected change, we take the absolute value of a random point from this distribution as an offset to delay detection.

In our experiments, we vary these parameters independently according to the given interval and step size while keeping the others at the given default value:

	Interval	Step Size	Default
p_{wrong}	$[0, 0.10]$	0.005	0.015
p_{miss}	$[0, 0.80]$	0.05	0.015
std_{offset}	$[0, 15]$	1	1

3.5.2. Evaluation Data

Stream mining frameworks such as MOA [Bif+10] and River [Mon+21] focus on stream classification, as indicated by the streams they offer. We investigate regression which require continuous *Ground Truth (GT)* time series of the form: $\mathrm{GT}(t) = ((t, x(t)), c(x(t), t))$. Here $t \in \mathbb{T}$ is the time, $x(t) \leftarrow X \mid t \in \mathbb{T}$ the observed input stream, and $y_t = c(x(t), t) \mid t \in \mathbb{T}$ the observed values of the target variable. Such continuous GT is not readily available in Stream mining evaluation frameworks, thus we generate $c(x(t), t)$ with a Gaussian process (GP) according to the following priors, where $C_{sin}(x, t)$ and $C_{rbf}(x, t)$ are adaptions from classification to regression used in [Joã+04] and [VBW16], and similar to River RBF streams:

$$C_b(x,t) = RBF(x \mid l_{gr}, v_{gr}) \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes B(t \mid v_{gb}) \tag{3.2}$$

$$C_{sin}(x,t) = RBF(x \mid l_{gr}, v_{gr}) \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes Sin(t \mid s * v_{gb}) \tag{3.3}$$

$$C_{rbf}(x,t) = RBF(x \mid l_{gr}, v_{gr}) \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes RBF(t \mid s * v_{gb}) \tag{3.4}$$

$$\begin{aligned} C_{mix}(x,t) = {} & RBF(x \mid l_{gr}, v_{gr}) \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes RBF(t \mid s * v_{gb}) \\ & \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes Sin(t \mid s * v_{gb}) \\ & \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes B(t \mid v_{gb}) \end{aligned} \tag{3.5}$$

$$\begin{aligned} C_{bmix}(x,t)) = {} & RBF(x \mid l_{gr}, v_{gr}) \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes B(t \mid v_{gb}) \\ & \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes B(t \mid v_{gb}) \\ & \oplus RBF(x \mid l_{gr}, v_{gr}) \otimes B(t \mid v_{gb}) \end{aligned} \tag{3.6}$$

Here l_{gr}, v_{gr}, v_{gb} are kernel parameters, chosen as follows: $l_{gr} = 0.1$; $v_{gr} = 0.25$; $v_{gb} \in \{0, 0.005, 0.01, 0.02\}$. From these parameters v_{gb} controls the drift speed, $s = 2000$ scales v_{gb} so that the sinus prior has a similar drift speed as the other priors for a given v_{gb}. We want to note that, even with fixed parameters, these are stochastic priors over 'possible' time series. We can potentially sample infinitely many actual time series from these priors, which can severely differ in their characteristics. For our evaluation, we use this fact to guarantee generalization by generating 50 distinct time series per parameter configuration, resulting in an evaluation performed on a total of $50 \times 5 \times 4 \times 4 \times 10 = 40000$ distinct time series. We visualize example time series drawn from such priors in Figure 3.2. For evaluation one time series is $t_{end} = 1000$ simulation units (su) long. The input stream $x(t)$ changes $N_c \in \{50, 100, 200, 400\}$ times within this 1000 su time frame, with the time of a change $t_c \leftarrow U[0, t_{end}]$ and $x(t_c) \leftarrow U[-1, 1]$. While tested on the $[-1, 1]$ range our method is compatible with any value range as the underlying GP learns its scale parameters from the data and is thereby scale invariant in time as well as in space.

3.5.3. Evaluation Metrics

For DEAL and every baseline with every data generator, we perform the following: We evaluate all parameter configurations 50 times, each time on a different time series. For each time series, we compute the RMSE of the estimation $y_{\theta t}$ against the ground truth, and log the total number of measurements N_m performed by each algorithm across each time series. We plot the RMSE over the N_m for all different configurations and different time series. Additionally, we calculate the percentage of measurements a competitor (DEAL) saves compared to a baseline (the CM baseline):

Definition 3.2 (Saved data)**.** *The saved data is* $sd = (Mb(e) - Mc(e))/Mb(e)$*, where* $Mb(e)$ *and* $Mc(e)$ *are the number of measurements a baseline* b *and a competitor* c *need to reach the same mean RMSE value* e *for the first time.*

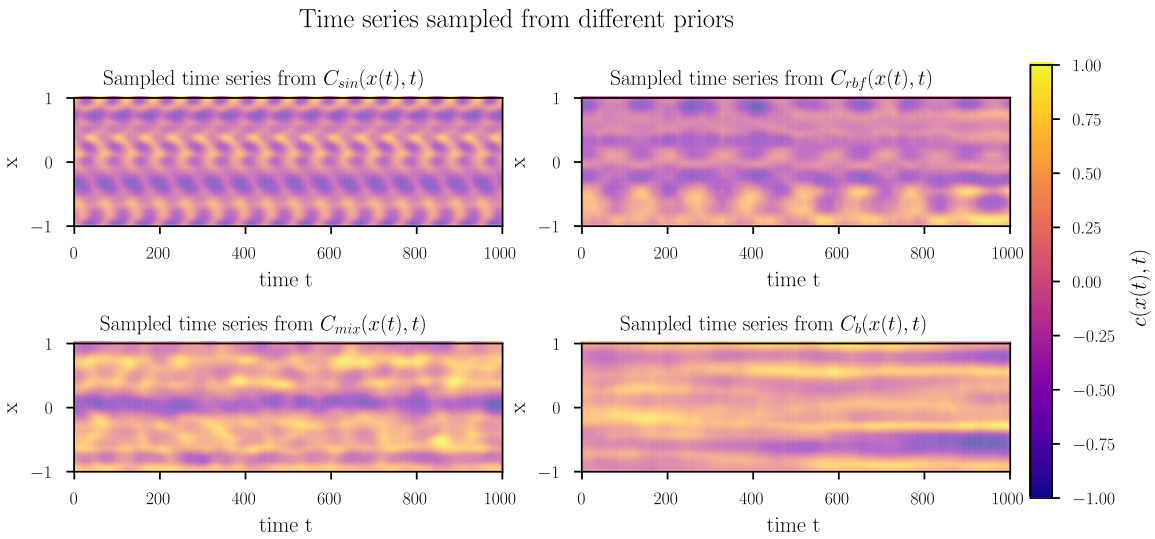
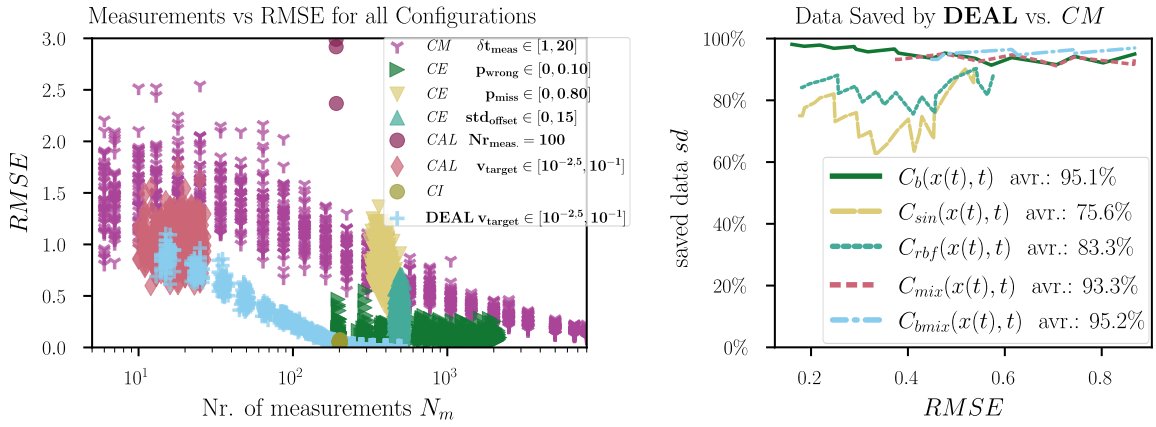


Figure 3.2.: Example time series sampled from the random function priors. Here, we show $y_t = c(x(t), t)$ normalized to $[1, -1]$ for better visualization.

3.6. Evaluation

3.6.1. Comparison of DEAL Against Baselines



(a) (b)

Figure 3.3.: Baseline comparison; 3.3a the relationship between N_m and RMSE for all parameters of the respective approach given in the legend and across different time series; 3.3b data gain against *CM* baseline.

Figure 3.3a shows the RMSE against the number of measurements N_m for DEAL and the four baselines (with two variants of *Classic Active Learning (CAL)* and three variants of *Change Error (CE)*) across a variety of parameter configurations. We can see that DEAL (+) needs on average fewer measurements to achieve lower RMSE than any configuration of any baseline (e.g., $N_m = 100$ for average $RMSE = 0.25$). Next, DEAL has less variance across the 50 time series than the baselines. This means for any fixed v_{target}, DEAL adapts

better to the individual behavior of a time series, while the baselines depend on how well their parameters match the time series behavior. The ideal change detector *CI* (●) shows a similar error as DEAL at $N_m = 200$. But, *CI*'s measurement count depends directly on the input stream's change frequency. Thus, *CI* will still carry out the same number of measurements (more than needed), even if a higher error would be acceptable. Further, as soon as the change detector is imperfect, as with *CE* (with offset ▲, missed ▼ and wrong ▶ detections) we observe a sharp increase in RMSE and its variance. The *CAL* approach (⧫) fails because once it reaches the given error threshold, it stops collecting data, never aware of any possible drift. Forcing *CAL* to collect 100 measurements regardless of the error threshold (●) causes it to learn an average value, rather than the correct relationship between the variables.

Data Gain: The *CM* baseline is the only baseline that allows users to indirectly control the RMSE by choosing the measurement time interval δt_{meas}. DEAL can directly control the user-required error thresholds v_{target}. For any other baseline, control of the resulting estimation error via a parameter is not possible. Figure 3.3b shows for a given RMSE (achieved by setting δt_{meas} or v_{target} appropriately) how much data DEAL can save compared to *CM*. In regions of low error ($RMSE \in [0.1, 0.35]$), DEAL saves over 95% data compared to *CM*. Moving to regions of higher error ($RMSE > 0.5$), we observe a slow decline in saved data. We hypothesize that this is because the larger the allowed error, the less benefit a precisely selected measurement time (based on drift behavior) has, compared to manual selection δt_{meas} of *CM*. The drop in saved data ($RMSE \in [0.3, 0.45]$) for C_{sin} and C_{rbf} is due to their periodic nature, which is why *CM* performs acceptable even without dynamic adaption.

3.6.2. Impact of the User-required Error Threshold on Estimation Error

The true estimation error RMSE should be close to the user-required standard deviation $std_{target} = \sqrt{v_{target}}$, so that the user can trust the model predictions. In Figure 3.4, the dotted line shows such proportional behavior. For small thresholds $std_{target} < 0.25$, DEAL overestimates the actual error. For higher errors $std_{target} > 0.25$, it underestimates the error. At all times, the actual error is close to the given threshold and behaves nearly proportionally as required.

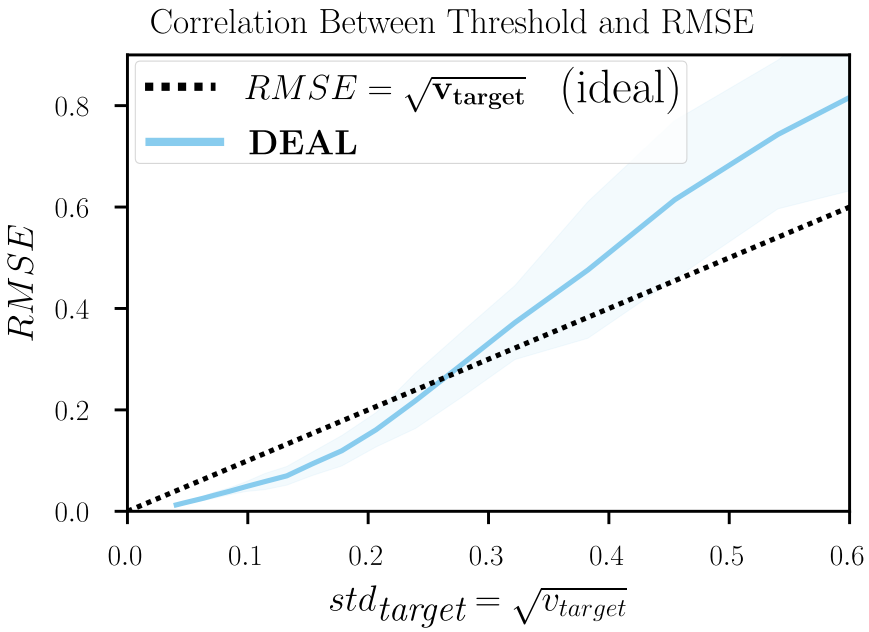


Figure 3.4.: Average RMSE for a given threshold in comparison with the user-required standard deviation $std_{target} = \sqrt{v_{target}}$.

3.6.3. Distribution of Measurements over Time

Figure 3.5 shows the number of measurements DEAL performs per time step (*simulation unit su*), depending on the user-required error threshold. In the early stages (the first 100 *su*), DEAL performs more measurements because it needs to calibrate. Roughly after $t_{conv} = 200$ *su*, it takes a constant number of measurements per *su*, just enough to reach the error threshold. This is the time DEAL has observed enough data to learn the stochastic drift behavior, thus measuring in drift matching regular intervals to ensure the required error threshold. With even more time, the variance of the number of measurements tends to decrease. Namely, the more data DEAL has seen, the closer its learned drift parameters are to the true ones.

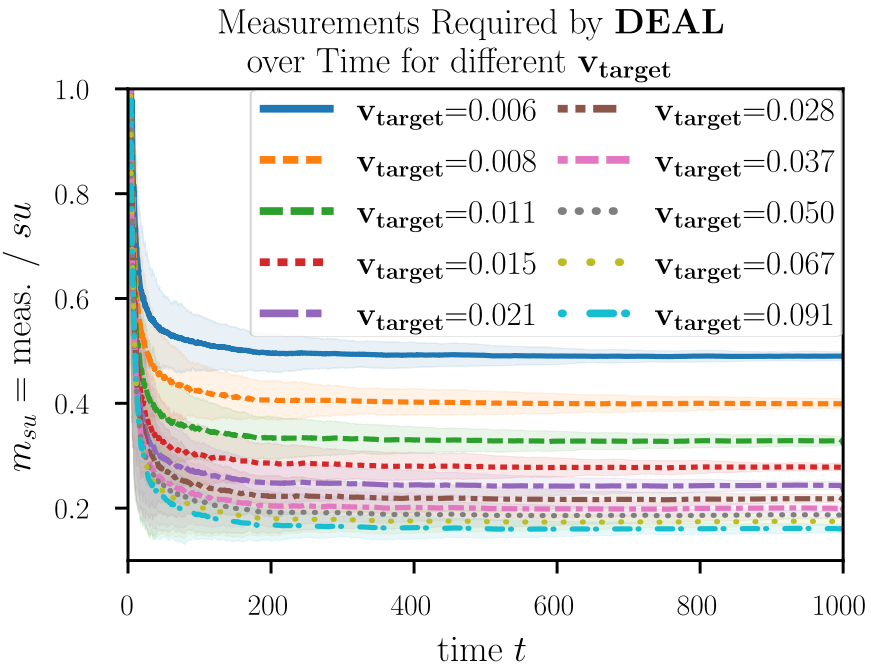


Figure 3.5.: Convergence of the average amount of performed measurements per time step (for different thresholds).

3.6.4. Relation Between User-required Error Threshold and Performed Measurements

Figure 3.6 shows how many measurements per *su* ($\bar{m}_{su}$) DEAL requires to reach a given error threshold v_{target}. As expected, if a user requires a lower estimation error, more measurements are needed to reach that error. To provide an estimation (dotted line) of this relation, we used a genetic function fitter. This relationship aids in estimating the required measurements and associated costs of reaching the user-required error threshold.

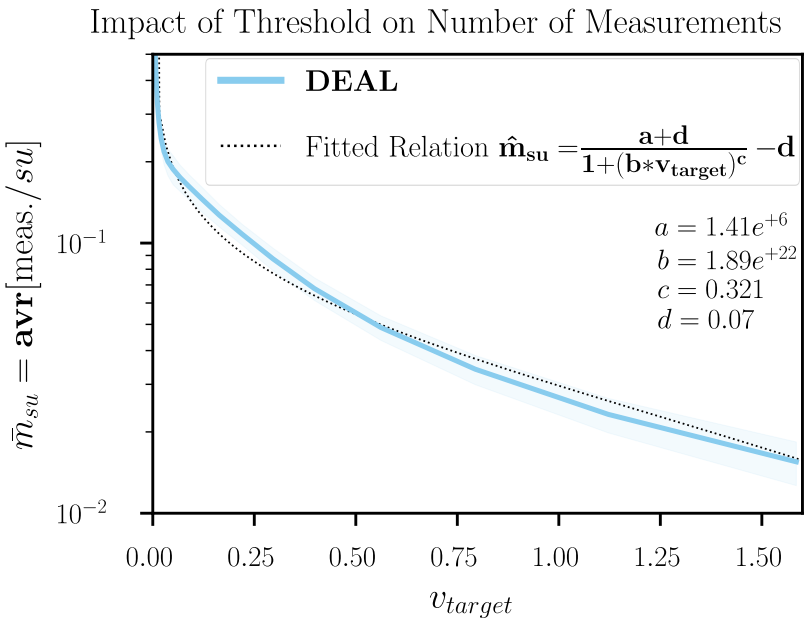


Figure 3.6.: Average amount of required measurements for a given threshold and function fit of this relation.

3.7. Chapter Conclusion

In this chapter, we investigated the challenges that arise from a drifting DGP. That is, the relationship between the input and target variables changes over time due to unobserved environmental influences.

Specifically, we addressed active learning regression with drift, i.e., predicting an expensive-to-measure continuous target variable that is affected by drift. Learning a model in such a setting is already difficult if observing the DGP is not costly, requiring constant monitoring of the target variable and recalibration in the case of drift. In an AKD setting, observing the DGP is costly; thus, monitoring the target variable continuously becomes impractical, requiring intelligent sub-sampling while maintaining a required accuracy. Current work on drift in active learning tends to focus on classification, and the resulting methods do not easily translate to regression. In contrast, active learning for regression under drift is not well covered in existing research.

We proposed DEAL, a method that adapts the frequency of measurements to the drifting relationship, to reach a given user-required error threshold. DEAL models drift by predicting the target variable and estimating the variance of that prediction at arbitrary points in time. Given a DGP which drifts constantly, DEAL requires, on average, 20 times fewer measurements over the full range of user-required error thresholds than the Consecutive Measurement (CM) baseline used in practice. Further, DEAL tunes itself to the drift to match the user-required error thresholds, while other methods require manual tuning.

Beyond its current focus on stochastic behavior, DEAL provides the modularity needed to integrate domain-informed kernels. Exploiting this to encode physical priors or structural periodicity to improve uncertainty estimates is a key direction for future research.

Part IV.

Uncertainty Quantification of Integrated Measurements

4. The Brownian Integral Kernel: A New Kernel for Modeling Integrated Brownian Motions

The content of this chapter bases on the following publication:

- Béla H. Böhnke, Edouard Fouché and Klemens Böhm. 'The Brownian Integral Kernel: A New Kernel for Modeling Integrated Brownian Motions'. In: *Data Science: Foundations and Applications*. Ed. by Xintao Wu et al. Singapore: Springer Nature, 2025, pp. 122–134. DOI: 10.1007/978-981-96-8295-9_8

Keywords: Integral Measurements; Learning from Aggregated Data; Integrated Brownian Motion; Gaussian Process Regression; Kernels

4.1. Chapter Overview

This chapter focuses on the challenge of uncertainty quantification, namely: How can we quantify uncertainty about something we cannot observe? To this end we study the problem of learning from integrated data, which inherently contains uncertainty about the actual value of the Data-Generating Process (DGP) due to integration. As there is no free lunch, quantifying the uncertainty about a unknown behavior requires additional assumptions. We make the assumption that the underling DGP behaves approximately like a Brownian motion:

Brownian motion (Figure 4.1a) is central to modeling various physical and technological processes, such as: (1) The movement of particles in a fluid [FGL15]. (2) The movement and loosening of machine elements due to vibration [LBD18]. (3) The behavior of financial and other markets [MZG16], population behavior and effects. (4) The load profile of electrical grids where producers and consumers with varying loads are plugged into or out of the grid at any time. These processes often involve stochastic uncertainties, which are effectively modeled using Gaussian processes and Brownian kernels [EMA18], enabling synthesis of process data, regression of real-world data with associated uncertainty, and the combination of both, i.e., synthesis of data from partially conditioned models.

However, in many real-world scenarios, data collection pipelines contain 'integrators' that implicitly or explicitly aggregate data over time intervals [MCO07] (Figure 4.1b). Direct observation of the quantity of interest is not possible in such scenarios. Examples include sensors with inherent integration properties (e.g., temperature sensors with heat

capacity) or practical constraints (e.g., smart meters providing 15-minute aggregated load data to value privacy). This aggregation obscures the underlying behavior and increases uncertainty, which conventional Brownian kernels cannot model accurately (Figure 4.1c). For instance, load forecasting requires precise variance estimation to manage short-term peaks in energy consumption [MO05; She+18; Gil+20]. Without the correct variance, providers risk grid instability or legal penalties due to insufficient capacity [CV98]. Similar problems arise in other domains like disease monitoring and simulation [TLK20]. As emphasized by [FTM19], the importance of integral kernels for modeling integrating processes is undisputed in many fields.

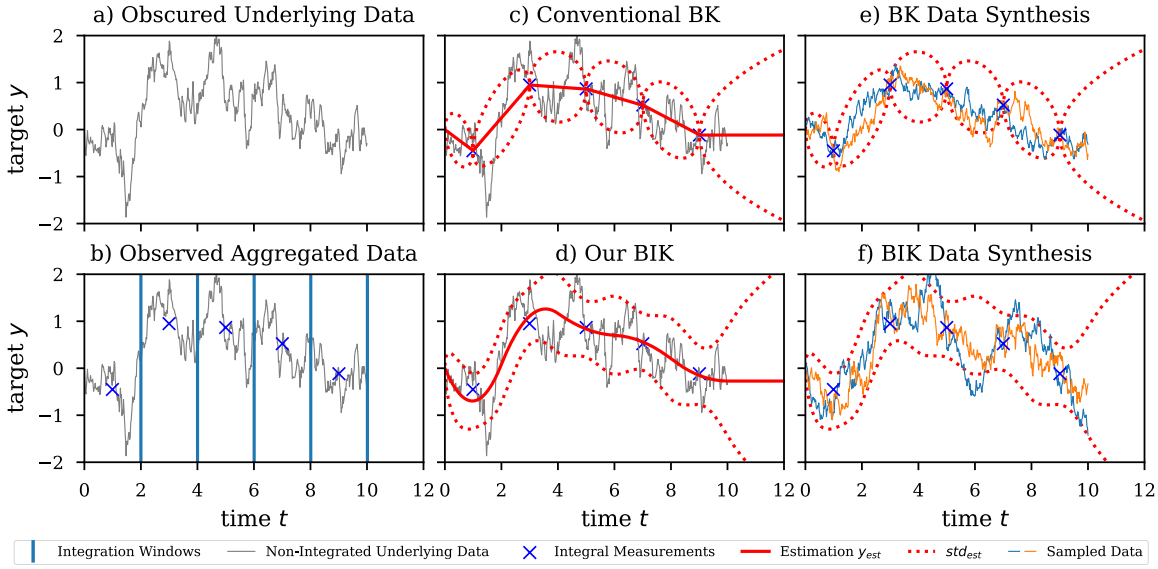


Figure 4.1.: Comparison of modelling and synthesis of integrated Brownian data with the conventional Brownian Kernel (BK) and our *Brownian Integral Kernel (BIK)*.

While a few integral kernels exist (e.g., the Radial Basis Function Integral Kernel (RBFIK) [SAL19; FTM19]), they are often computationally expensive or lack a direct connection to physical processes, making them unsuitable for integrated Brownian motion.

To address these challenges, we derive the *Brownian Integral Kernel (BIK)*, a novel analytical solution for modeling aggregated data from Brownian motions. The Brownian Integral Kernel (BIK) accurately estimates variance and supports Gaussian process regression, enabling better predictions and uncertainty quantification (Figure 4.1d), as well as data synthesis (Figure 4.1f). We validate its performance through extensive experiments on synthetic and real-world datasets. Further, to foster accessibility, we provide a Python implementation[1] of the BIK compatible with the widely used [GPy12] framework for Gaussian Process (GP) modeling.

[1] https://github.com/bela127/Brownian-Integral-Kernel

4.1.1. Fundamentals

A Gaussian Process (GP) is a standard tool for modeling continuous processes, offering flexible kernels that enable accurate regression, uncertainty quantification, and data generation [EMA18]. GPs automatically adjust their complexity based on the data, making them robust to overfitting in small datasets and scalable to larger datasets, unlike methods that require fixed parametric structures, manual tuning, or large datasets [FHF06]. A key strength of GPs is their use of kernel functions, which allow for modeling complex, non-linear relationships without explicit parametric assumptions. The general workflow for using GPs is outlined in Algorithm 4.1: In a first step (step 1) a domain expert investigates the properties of the DGP and chooses an appropriate kernel function or combination of kernel functions. After training (step 2), GPs provide a posterior distribution over functions, making it possible not only to predict outcomes (step 3) but also associated uncertainty and to generate new realistic (continuous) data (step 4). This capability is particularly valuable for simulations and uncertainty quantification.

Algorithm 4.1 Generic Use-Case of a GP

Input: Training data $\mathbf{D}_{\text{train}}$, Test data $\mathbf{X}_{\text{test}} = \{t_1, t_2, \ldots, t_n\}$
Output: Predicted mean $m(\mathbf{X}_{\text{test}})$, uncertainty $\sqrt{v(\mathbf{X}_{\text{test}})}$, generated (sampled) data $\mathbf{Y}_\theta$
Step 1: Kernel Selection
Choose a kernel function $k(\cdot,\cdot)$ (e.g., Radial Basis Function (RBF), Matérn, our BIK, etc., or combination).
Step 2: Training
Train the Gaussian process model on $\mathbf{D}_{\text{train}}$ using the kernel $k(\cdot,\cdot)$.
Learn hyperparameters θ of the kernel by maximizing the log marginal likelihood.
Step 3: Prediction and Uncertainty Estimation
For test data points $\mathbf{X}_{\text{test}}$, compute the posterior mean $m(\mathbf{X}_{\text{test}})$ and variance $v(\mathbf{X}_{\text{test}})$ using the trained model.
Step 4: Data Generation from Posterior Distribution
Sample from the posterior $\mathcal{N}(m(\mathbf{X}_{\text{test}}), v(\mathbf{X}_{\text{test}}))$ to generate new data $\mathbf{Y}_\theta$.

4.2. Related Work

Despite the strengths of GPs, conventional kernels face limitations when modeling aggregated data. The closest related kernel is the *RBFIK*, introduced by [SAL19; FTM19]. While the RBF kernel is widely used because of its universal approximation property [MXZ06], it has no relation to actual physical processes, thus lacking physical grounding. This questions the usefulness of its integral version. The Brownian Kernel (BK), on the other hand, directly relates to physical processes such as load profiles or market behavior. However, it assumes direct observations, limiting its utility for aggregated data.

Independently of [SAL19; FTM19], [Lon+20] derived a line-RBF integral kernel, which calculates the covariance between lines and measurements. While this kernel also lacks physical grounding, it further integrates over space, not time. But to quantify the uncertainty of integral measurements we require integration over time.

Numeric integration-based methods [OR11; Tan+19] and Markov Chain Monte Carlo (MCMC) approximations [TLK20] are applicable to a wide range of kernels. However they are computationally expensive compared to analytic solutions. Here, [OR11] presented a method to perform numerical integration of any kernel in an classification scenario. Similarly, in [Tan+19] a mixture of GPs and numeric integration is used to learn from spatially integrated data. However, this approach is always an approximation, and the approximation error only decreases with a quadratic number of evaluation points. Coupled with the quadratic complexity of GP regression, this becomes computationally challenging for larger datasets. [TLK20] uses MCMC to approximate the integral during training. However, such a method requires an adaptive training strategy, preventing one from using existing GP models and standard training algorithms. Some work [Hen+18] tried to reduce the computationally complexity by solving a part of the integration for the line-RBF kernel and solving the missing part with numeric integration. The authors claimed that there is no analytic solution to the line-RBF integral kernel, which was disproven by [Lon+20]. Fully analytic solutions, such as ours, are superior because they only require constant calculation time and provide an exact solution to the integral [Lon+20].

For smooth kernels, it is feasible to approximate the integral kernel by a finite spectral approximation on a Hilbert space [SS19; Jid+17]. [Pur+19] uses this approximation to solve kernel line integrals associated with the problem of x-ray tomographic reconstruction. However, such spectral approximations fail for discontinuous kernels like the Brownian Kernel (BK) [Pur+19].

Efforts to adapt Kalman filters [Sun+19; XD19; Tag+17; Liu+20; KL19; SH12; Tod+20; RR10] and integrate numeric methods [OR11; Lon+20] have resulted in problem-specific or approximate solutions. For example, the extensions of Kalman filters for time-continuous modeling [Tag+17; Liu+20] requires a time-transition functions which is challenging to drive and often problem-specific. This is why the work of [KL19] goes so far as using GPs as a noise model for Kalman filters. Which would, in turn, require a kernel like we proposed. Further, [SH12; Tod+20] have shown that such Kalman filters are equivalent to GPs under certain assumptions, i.e., it is possible to transform the kernel of a GP to a transition function for a Kalman filter or even use the kernel directly in an adapted Kalman filter implementation [RR10]. Such methods would directly benefit from our newly proposed kernel.

Some approaches are out of scope for this work. Time-discrete methods, such as Kalman filters, lack the continuous modeling required for integrated data, while extensions for time-continuous modeling [Tag+17; Liu+20] require problem-specific transition functions. Methods like a Support Vector Machine (SVM) do not support uncertainty estimation, and neural networks lack efficient mechanisms for data generation and uncertainty quantification. These limitations make them unsuitable as baselines for this study.

This paper introduces the *Brownian Integral Kernel (BIK)*, an analytical solution for modeling aggregated data. The BIK directly relates to common physical processes and provides exact covariance estimation for integrated Brownian motion, enabling efficient computation within standard GP frameworks while maintaining a direct connection to physical processes.

4.3. Problem Statement

We assume a data-generating process $C(t) := B(t)$ that behaves like a Brownian motion. Some real processes that behave like this do not allow observation of their actual value $b(t)$ (realization) at time t. One can only observe average or integral measurements of $B(t)$, while the true value $b(t)$ remains unknown.

Definition 4.1 (Integral measurement)**.** *An integral measurement $\mathcal{B}(s, e)$ is the integration over time t of measurements from $b(t)$ from start time s to end time e:*

$$\mathcal{B}(s, e) = \int_s^e b(t)\,\mathrm{d}t.$$

Even if $b(t)$ follows a Brownian motion B which is per definition erratic, integral measurements $\mathcal{B}(s, e)$ behave differently, this is because integration smoothes the values leading to a smoother function. In consequence the measured value $\mathcal{B}(s, e)$ is most certainly not the true value of the underlying process $B(t)$. However, a Brownian motion model, e.g., a Gaussian process with Brownian kernel $k_{ff'}(t, t') = v_b \cdot min(t, t')$ assumes that the observed data points are the true values. Fitting such a model on integral measurement $\mathcal{B}(s, e)$ gives the wrong predictive variance of zero at measurement locations. To obtain the correct covariance, i.e., to correctly model the additional variance due to integration, a new kernel is necessary.

The research question is:
How can we incorporate the knowledge about the integration into a kernel to account for the additional variance, and correctly reflect the inherent uncertainty of integral measurements?

4.3.1. Notation

Stochastic Processes S or random processes are random functions $S : t \mapsto S(t)$ where $t \in \mathbb{T}$ is interpreted as time with domain $\mathbb{T} = \mathbb{R}_+$ [C05].

Brownian Motions $B : t \mapsto B(t)$ are stochastic processes characterized by random increments: $\delta B(\delta t) = B(t + \delta t) - B(t)$, these increments follow the normal distribution $\delta B(\delta t) \sim \mathcal{N}(0, \delta t)$. Brownian motion occurs in many scenarios where many small, random, and independent changes lead to a random overall change [LRS13].

A Gaussian Process (GP) is a random function such as:
$\mathrm{GP}(x) \sim \mathcal{N}\left(m(x), v(x)\right)$, illustrating that the process $\mathrm{GP}(x)$ comes from a normal distribution with mean $m(x)$ and variance $v(x)$ depending on x [LRS13]. One often uses GPs for applications other than time series modeling, in the case of time series modeling one uses t as a parameter instead of x, in such cases a GP is a stochastic processes. For instance, [Kar+20] uses a GP as a random function, i.e., as a probabilistic prior over smooth functions, and to model functions with associated uncertainty.

Kernel Functions $k(x, x')$, also called covariance functions, can define a GP instead of using a mean and variance function. Here, x, x' are two points from the regime of the GP, and $k(x, x')$ returns the covariance of the GP according to these given points.

The Brownian Kernel $k(t, t') = v_b \cdot min(t, t')$, with points in time t and t' has a variance parameter v_b. It translates to the drift speed of the resulting Brownian motion [LRS13], i.e., how fast a time series diverges from a given starting point.

Primitive Functions , also called antiderivative, commonly use upper case letters. However, to avoid overlap in notation with random functions, we will use calligraphic letters $\mathcal{F}$ to denote primitive functions. In addition, we will use index notation to denote partly integrated functions, which is required for multivariate functions. For example: $k(t, t') = k_{ff'}(t, t')$ is the original function, $k_{\mathcal{F}f'}((s, e), t') = \int_s^e k_{ff'}(t, t')\,\mathrm{d}t$ is the one time integration and $\mathcal{F}((s, e), (s', e')) = k_{\mathcal{F}\mathcal{F}'}((s, e), (s', e')) = \int_s^e \int_{s'}^{e'} k_{ff'}(t, t')\,\mathrm{d}t'\,\mathrm{d}t$ the full (two-time) integration. Indexes of a primitive function $\mathcal{F}_{t,t'}$ (denoted with a calligraphic letter) are simply used for naming and do not have any special meaning if not otherwise specified in the text.

4.4. The Brownian Integral Kernel

We propose to model the additional variance (described in the previous section) directly in a new kernel to solve the problem of mis-estimating the variance. Thus, this calls for a kernel that yields the covariance between two integrated time intervals of a Brownian motion. Following [FTM19] we can derive this kernel by integrating the Brownian Kernel (BK):

Definition 4.2 (The Brownian Integral Kernel (BIK))**.** *The Brownian Integral Kernel (BIK) is the two-time integration over time intervals* (s, e) *and* (s', e') *of the Brownian Kernel:*

$$k_{\mathcal{F}\mathcal{F}'}((s, e), (s', e')) = v_b \cdot \int_{s'}^{e'} \int_s^e min(t, t')\,\mathrm{d}t\,\mathrm{d}t',$$

where s, s' *and* e, e' *are the integration start and end times respectively.*

This kernel generalizes the BK to integral measurements by modeling the higher variance due to integration, which the BK neglects. For intuition and proof of how integrating a kernel is equivalent to integrating the underlying process, please refer to Appendix A.0.2. Further, a kernel needs to fulfill some properties, namely, it needs to be positive-semidefinite [SS02]. This property follows directly from its derivation as an integral of a kernel and we provide proof in Appendix A.0.1.

Theorem 4.1 (The solution to the Brownian Integral Kernel (BIK))**.** *The Brownian Integral Kernel (BIK) admits the following solution:*

$$
k_{\mathcal{F}\mathcal{F}'}((s,e),(s',e')) =
$$
$$
v_b \cdot \begin{cases}
\textit{if } s \le s' < e' \le e: \\
\quad \mathcal{F}_{tt}((e',t),(s',e')) + \mathcal{F}_{tt'}(s',e') + \mathcal{F}_{t't'}((s,s'),(s',e')) \\
\textit{if } s' < e' \le s < e: \\
\quad \mathcal{F}_{tt}((s,e),(s',e')) \\
\textit{if } s' \le s < e' \le e: \\
\quad \mathcal{F}_{tt}((s,e),(s',s)) + \mathcal{F}_{tt}((e',e),(s,e')) + \mathcal{F}_{tt'}(s,t') \\
\textit{if } s \le s' < e \le e': \\
\quad \mathcal{F}_{t't'}((s,s'),(s',e')) + \mathcal{F}_{t't'}((s',e),(e,e')) + \mathcal{F}_{tt'}(s',e) \\
\textit{if } s < e \le s' < e': \\
\quad \mathcal{F}_{t't'}((s,e),(s',e')) \\
\textit{if } s' \le s < e \le e': \\
\quad \mathcal{F}_{tt}((s,e),(s',s)) + \mathcal{F}_{tt'}(s,e) + \mathcal{F}_{t't'}((s,e),(e,e'))
\end{cases}
$$

, with $\mathcal{F}_{tt}, \mathcal{F}_{t't'}, \mathcal{F}_{tt'}$ being sub-parts of the primitive function:

$$\mathcal{F}_{tt}((l',u'),(l,u)) = \frac{1}{2}u^2 \cdot u' - \frac{1}{2}u^2 \cdot l' - \frac{1}{2}l^2 \cdot u' + \frac{1}{2}l^2 \cdot l', \tag{4.1}$$

$$\mathcal{F}_{t't'}((l',u'),(l,u)) = \frac{1}{2}u'^2 \cdot u - \frac{1}{2}l'^2 \cdot u - \frac{1}{2}u'^2 \cdot l + \frac{1}{2}l'^2 \cdot l, \textit{ and} \tag{4.2}$$

$$\mathcal{F}_{tt'}(l,u) = \frac{1}{3}(u-l)^3 + (u-l)^2 \cdot l, \tag{4.3}$$

with (u,u'),(l,l') being the upper and lower integration bounds.

The multiple occurrences and symmetries of the sub-parts stems from the underlying kernel geometry and will become clear in the detailed proof, which we provide in Section 4.5.

The resulting kernel $k_{\mathcal{F}\mathcal{F}'}$ calculates the covariance between training data intervals, i.e., between two integrated time intervals. However, during inference, most users are interested in the predictive variance ${k_{ff'}}^{post}$.

Definition 4.3 (The predictive variance of a GP)**.** *The predictive variance ${k_{ff'}}^{post}$ of a GP is a variance within the original space of the non-integrated Brownian motion. For a given point t, ${k_{ff'}}^{post}$ quantifies how likely a prediction $y_{\theta t}$ is equal to the unobserved ground truth y_t.*

The predictive variance ${k_{ff'}}^{post}$ is often used to quantify the uncertainty associated with predictions $y_{\theta t}$ [Set12]. To calculate it, we need to calculate the *partly integrated* covariance $k_{\mathcal{F}f'}$.

Definition 4.4 (The *partly integrated* covariance)**.** *The* partly integrated *covariance $k_{\mathcal{F}f'}$ is the covariance between an inference point t' (point in the original space of non-integrated Brownian motion) and an integration time interval (s,e). It is calculated as a one-time integration of $k_{ff'}$:*

$$k_{\mathcal{F}f'}((s,e),t') = v_b \cdot \int_s^e min(t,t')\,\mathrm{d}t.$$

Theorem 4.2 (The *partly integrated* kernel)**.** *The* partly integrated *kernel admits the following solution:*

$$k_{\mathcal{F}f'}((s,e),t') = v_b \cdot \begin{cases} if\, t' \le s < e: \\ \quad \mathcal{F}_t(t',s,e) \\ if\, s < t' < e: \\ \quad \mathcal{F}_{t'}(s,t') + \mathcal{F}_t(t',t',e) \\ if\, s < e \le t': \\ \quad \mathcal{F}_{t'}(s,e) \end{cases}$$

with $\mathcal{F}_t, \mathcal{F}_{t'}$ *as follows:*

$$\mathcal{F}_t(t,l',u') = t \cdot u' - t \cdot l', \tag{4.4}$$

$$\mathcal{F}_{t'}(l',u') = \frac{1}{2}u'^2 - \frac{1}{2}l'^2. \tag{4.5}$$

We obtain this partly integrated covariance by one-time integration of $k_{ff'}(t,t')$. The proof follows directly from the proof of Theorem 4.1, see Section 4.5. Here, Equations 4.4 and 4.5 are an intermediate result (compare with Equations 4.6 and 4.7) from Section 4.5.

We can now calculate the predictive variance ${k_{ff'}}^{post}$ of the Gaussian process, according to [RW06] with the standard formula for a GP. Here we use matrix notation, where $K_{ff'}, K_{\mathcal{F}f'}, K_{\mathcal{F}\mathcal{F}'}$ is the matrix obtained by using the appropriate kernel $k_{ff'}(*,*), k_{\mathcal{F}f'}(*,*), k_{\mathcal{F}\mathcal{F}'}(*,*)$, $K_{f\mathcal{F}'}$ is obtained by transposing $K_{\mathcal{F}f'}$, and ${K_{ff'}}^{post}$ is the result matrix that corresponds to ${k_{ff'}}^{post}$:

Corollary 4.1 (Posterior covariance matrix)**.**

$${K_{ff'}}^{post} = K_{ff'} - K_{f\mathcal{F}'}{K_{\mathcal{F}\mathcal{F}'}}^{-1}K_{\mathcal{F}f'}.$$

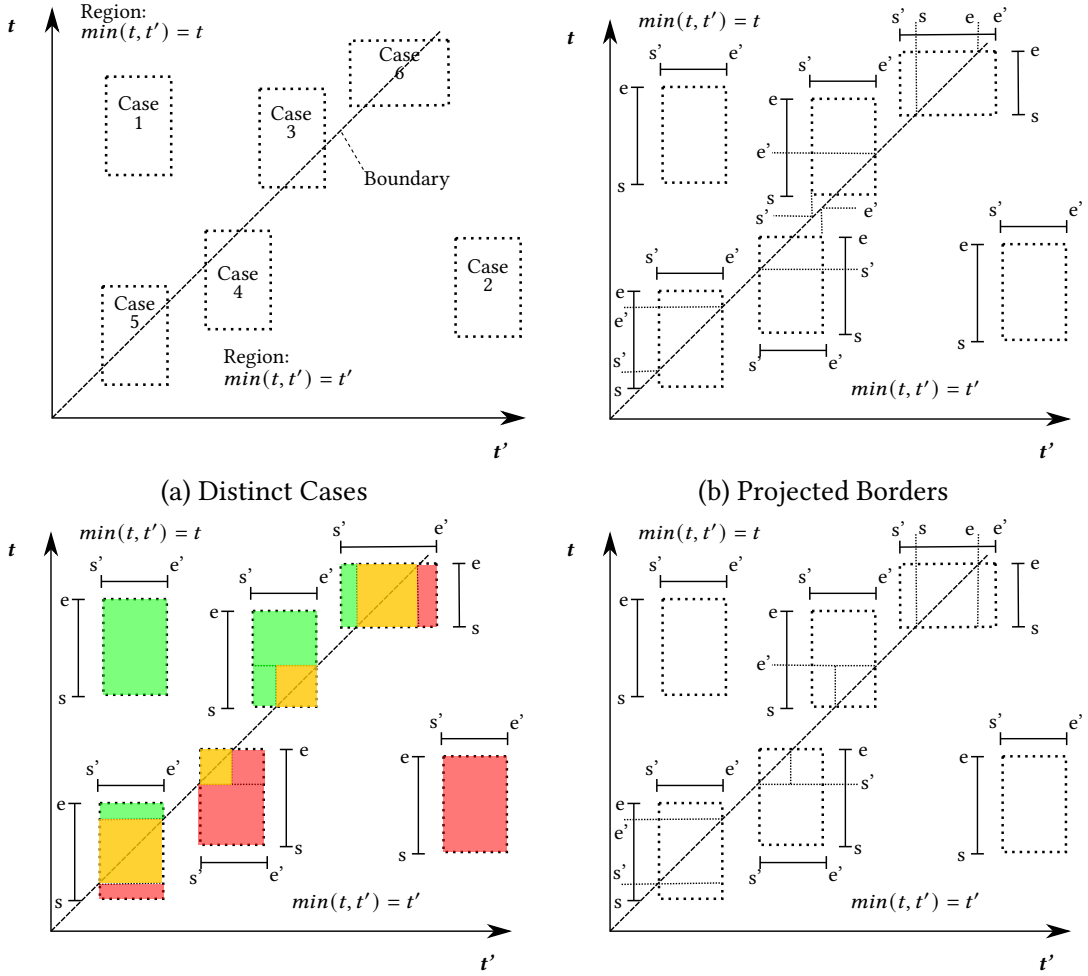
4.4.1. Computational Complexity of the BIK

Concerning computational complexity, one has to distinguish between the complexity of the training algorithm without kernel and the complexity of the stand-alone kernel. We perform all experiments (for our kernel as well as the baseline kernels) with the standard GP model from the GPy framework [GPy12], which has a complexity of $O(N^2)$. Depending on the use case and the needed accuracy, other less complex models can be employed to approximate GP. For example, one can use gradient-based models that allow batch training with a complexity of $O(N \cdot E)$, where E is the number of training epochs. Further, new work [RR10; SH12; Tod+20] has shown how to use GP kernels directly in an adapted Kalman filter implementation, thereby reducing the runtime to $O(N)$ as long as certain assumptions are met. Similarly, methods for training on time series can be used, like sliding windows that discard old measurements to keep N small, thereby losing information from old measurements. This is possible for all kernels, including ours.

If an exact solution is needed, the standard GP model with complexity $O(N^2)$ must be used. For each of these N^2 calculations, the kernel is invoked once, which is why kernel complexity can become an issue when too large. Our analytically derived BIK achieves

the lowest possible complexity of $O(1)$, which makes it ideal. In comparison, the work [Hen+18] has a complexity of $O(m)$ where m are the number of Monte Carlo iterations used to approximate the integral. Here, m needs to be high to obtain good approximations. While full numerical integration like in [OR11] and [Tan+19] is possible for any kernel, it comes with a complexity of $O(m^2)$ because of the nested Monte Carlo iterations. Coupled with the quadratic complexity of GP regression, this becomes computationally challenging for larger datasets.

4.5. Derivative Proof for the BIK



(a) Distinct Cases

(b) Projected Borders

(c) Sub-Parts

(d) Integral Bounds

Figure 4.2.: Integral case distinction; 4.2a The six distinct cases; 4.2b The projected integral borders; 4.2c Division of cases into (toned) sub-parts; 4.2d Bounds of sub-integrals

Proof. The function $min(t, t')$ is discontinuous, so the solution of the integral:

$$k_{\mathcal{F}\mathcal{F}'}((s, e), (s', e')) = v_b \cdot \int_{s'}^{e'} \int_{s}^{e} min(t, t') \, \mathrm{d}t \, \mathrm{d}t',$$

depends on the integration bounds s, e, s', e'. To prove Theorem 4.1, we need a case distinction to cover the different bound configurations. We visualize each bound configuration in Figure 4.2a. We do so by drawing a boundary into one quadrant where $min(t, t')$ switches from yielding t to yielding t'. This boundary is the linear function $t' = t$. In the region above the boundary, the function $min(t, t')$ yields t. In the region below, it yields t'. We then see that there are six cases of possible bound configurations (dotted boxes). Because

$min(t, t')$ is symmetric to the boundary, we can project s, e, s', e' onto the other axis, receiving an ordering of the integral borders. This ordering uniquely identifies each of the six cases. See if-conditions of Theorem 4.1 and Figure 4.2b (thin lines).

We further divide these six cases into smaller partitions and find that three common sub-parts, highlighted in Figure 4.2c, give solutions for each partition. While the solution for the sub-parts is structurally similar, the integral bounds are different. This is why we solve each sub-part independently of the specific bounds by substituting the upper bounds with u, u' and the lower bounds with l, l'. We name the resulting three sub-integrals after the regions they are part of: Integrals with integration bounds u, u', l, l' in one region $u, u', l, l' \in \{t \mid t < t'; t, t' \in \mathbb{T}\}$ are named $\mathcal{F}_{tt}$ and in the other region $u, u', l, l' \in \{t' \mid t' < t; t, t' \in \mathbb{T}\}$ are named $\mathcal{F}_{t't'}$ respectively. Integrals with bounds in both regions are named $\mathcal{F}_{tt'}$. In Figure 4.2c the sub-integral $\mathcal{F}_{tt}$ has color green, $\mathcal{F}_{t't'}$ red and $\mathcal{F}_{tt'}$ orange.

Sub-parts $\mathcal{F}_{tt}$ and $\mathcal{F}_{t't'}$: The function $min(t, t')$ is continuous for these sub-parts. This allows us to directly integrate once, leading to:

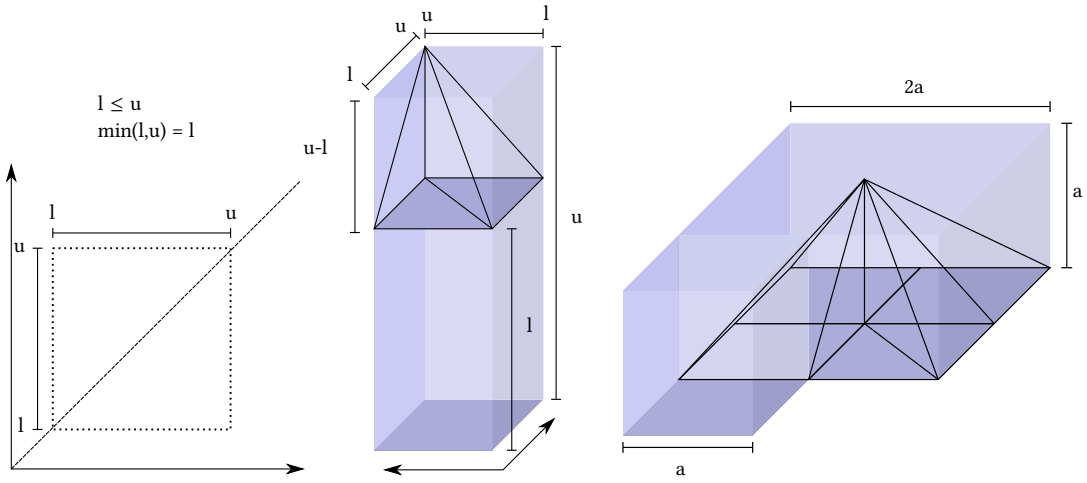
$$
\begin{aligned}
t < t' : \mathcal{F}_t(t, l', u') &= \int_{l'}^{u'} min(t, t')\,\mathrm{d}t' = \int_{l'}^{u'} t\,\mathrm{d}t' = \\
&\quad |_{t'=l'}^{u'}\ t \cdot t' = t \cdot u' - t \cdot l' \qquad (4.6) \\
t' < t : \mathcal{F}_{t'}(l', u') &= \int_{l'}^{u'} min(t, t')\,\mathrm{d}t' = \int_{l'}^{u'} t'\,\mathrm{d}t' = \\
&\quad |_{t'=l'}^{u'}\ \frac{1}{2}t'^2 = \frac{1}{2}u'^2 - \frac{1}{2}l'^2 \qquad (4.7)
\end{aligned}
$$

We finish the first part of the proof by integrating twice, which gives us:

$$
\begin{aligned}
t < t' : \mathcal{F}_{tt}((l', u'), (l, u)) &= \int_{l}^{u}\int_{l'}^{u'} t\,\mathrm{d}t'\,\mathrm{d}t = \\
&\quad |_{t=l}^{u}\ \frac{1}{2}t^2 \cdot u' - \frac{1}{2}t^2 \cdot l' = \frac{1}{2}u^2 \cdot u' - \frac{1}{2}u^2 \cdot l' - \frac{1}{2}l^2 \cdot u' + \frac{1}{2}l^2 \cdot l', \\
t' < t : \mathcal{F}_{t't'}((l', u'), (l, u)) &= \int_{l}^{u}\int_{l'}^{u'} t'\,\mathrm{d}t'\,\mathrm{d}t = \\
&\quad |_{t=l}^{u}\ \frac{1}{2}u'^2 \cdot t - \frac{1}{2}l'^2 \cdot t = \frac{1}{2}u'^2 \cdot u - \frac{1}{2}l'^2 \cdot u - \frac{1}{2}u'^2 \cdot l + \frac{1}{2}l'^2 \cdot l.
\end{aligned}
$$

These are the Equations 4.1 and 4.2 of Theorem 4.1.

Sub-part $\mathcal{F}_{tt'}$: The third integral $\mathcal{F}_{tt'}$ needs a separate solution because it intercepts the region border and, thus, is discontinuous. We derive Equation 4.3 by looking at $min(t, t')$ geometrically, see Figure 4.3. The volume of the geometry is equivalent to the two-times integration. We decompose the geometry into two subvolumes (sv), one cuboid (cub), and one more complex subvolume (com). We calculate the volume of the cuboid directly, see Figure 4.3a. By merging the geometry of the complex subvolume four times with itself,



(a) Total Volume (b) Pyramid

Figure 4.3.: Geometric view on integrating; 4.3a Total volume; 4.3b subvolumes combined to pyramid.

we obtain a pyramid, see Figure 4.3b. We now work out the integral bounds and obtain the solution to the double integral with the following system of equations:

$$\begin{cases} \mathrm{sv}_{com} = \frac{1}{4} V_{pyr} \\ V_{pyr} = \frac{1}{3} V_{cub} \\ V_{cub} = h \cdot w \cdot d \end{cases} \begin{cases} a{=}u{-}l \\ h{=}a \\ w{=}2a \\ d{=}2a \end{cases} \Longrightarrow \begin{cases} \mathrm{sv}_{com} = \frac{1}{3}(u-l)^3 \\ \mathrm{sv}_{cub} = (u-l)^2 \cdot l \\ \mathcal{F}_{tt'}(l,u) = \mathrm{sv}_{com} + \mathrm{sv}_{cub} \end{cases},$$

which results in Equation 4.3 of Theorem 4.1:

$$\mathcal{F}_{tt'}(l,u) = \frac{1}{3}(u-l)^3 + (u-l)^2 \cdot l.$$

To combine the three subintegrals into the six original cases, we still need the integral bounds of the subintegrals corresponding to the original cases. Using the symmetric property of $min(t,t')$, we can obtain these bounds, see Figure 4.2d. With the matching integral bounds, we obtain the solution to the BIK, Theorem 4.1. □

4.6. Experimental Design

We derived the BIK and highlighted its advantages over existing kernels. To validate our findings, we compare the BIK in various scenarios against the Brownian Kernel (BK), the RBFIK from [FTM19], and the Brownian Kernel with added white noise (BNK) [Boy07; Li+20]. The comparisons assess prediction quality, variance estimation, and the plausibility of data generated by a GP conditioned on integral measurements.

4.6.1. Used Data

For our evaluation, we use data from multiple sources: **Synthetic data:** *Synth* from a Gaussian process with a Brownian Kernel and subsequent numeric integration. **Simulated data:** *Load* generated with a realistic and widely used data generator [Pfl+22]. **Real-world data:** *Real* provided by a private household, *HIPE* using the HIPE data set [Bis+18], and *Stock* using daily stock market prices from [Kum24]. In the following we give more details on the datasets:

- For *Synth*, we draw data from a GP with a BK to obtain true Brownian motion behavior. To obtain integral measurements, we integrate this data numerically by summing over given equidistant time intervals. We generate 20 such time series, each with a length of 2.5k, 5k, 10k, and 20k time steps, and integrate over a window size $w \in [25, 50, 100, 200]$ steps, respectively. This results in the constant number of 1000 integral measurements for training.
- For *Load*, we use 17 consumer load profiles with one-minute resolution and a length of 7 days (time series of the electricity consumption of households) generated with a widely used data generator [Pfl+22]. This generator simulates real-world households according to 400+ parameters, such as household size, gender and age of household members, employment, photovoltaic, and many more, and is steadily improved. It is accepted as close to real-world data in the electricity community as witnessed by the citations listed in [Pfl+22]. Like smart meters, we aggregate the resulting one-minute profiles in 15-minute intervals.
- For *Real*, we use real-world load profiles provided by a private household, that are already aggregated in 15-minute intervals. Here no ground truth is available, and thus we only use it for evaluating the quality of the time series generated by the GP within Section 4.7.2.
- For *HIPE*, we use the High-resolution Industrial Production Energy (HIPE) data set [Bis+18]. It contains readings of ten machines and the main terminal of a power-electronics production plant.
- For *Stock*, we use daily stock market prices from [Kum24]. Weekly aggregation provides us with integrated training data.

We provide all used datasets which are otherwise not easily accessible, together with the experiment code within our GitHub repository to facilitate reproducibility.

It is important to note that, contrary to the intuition of the name *load profile*, load profiles can have both positive and negative values due to generating nodes such as photovoltaic systems. Especially for short-term modeling of such load profiles, they behave Brownean-like because of randomly moving cloud cover [Xu+09]. For long-term modeling, one almost always has additional periodicity (day and night). However, the periods are usually much longer than 15 minutes, which is why they are not relevant for modeling uncertainty due to integration. Our kernel can be used for long-term modeling with standard kernel composition by just adding a periodic kernel. Yet such an extension is not necessary for evaluating the short-term uncertainty effects we are targeting.

4.6.2. Metrics

We evaluate multiple application-relevant aspects of the kernels. One is often interested in the prediction error. However, in our scenario (where one can only observe integrals of the ground truth but not the ground truth itself) predictions will always be close to the mean within the observed integration interval. Therefore, standard metrics such as Mean Square Error (MSE) are not meaningful when comparing the kernels. Instead, this scenario requires an evaluation that combines prediction and prediction uncertainty. For this, we use the Weighted Mean Absolute Error (WMAE):

Definition 4.5 (The WMAE)**.** *The WMAE quantifies the estimation quality considering the estimated likelihood $p_{GP}(t)$ of an estimation $y_{\theta t}$ vs the Ground Truth (GT) y_t:*

$$WMAE = \frac{1}{|\mathbf{T}|} \sum_{t \in \mathbf{T}} p_{GP}(t) \cdot |y_{\theta t} - y_t|$$

with $\mathbf{T}$ *being the set of 'time' test points.*

The WMAE is an intuitive measure: The estimated likelihood $p_{GP}(t)$ quantifies for a prediction $y_{\theta t}$ how likely this prediction is. If the model is confident ($p_{GP}(t) \approx 1$) and the prediction is accurate, the WMAE is low. WMAE penalizes prediction errors more when the model is (wrongly) confident, and less when it is (correctly) uncertain. Note that we are interested in the likelihood per data point, thus it is not required to normalize the likelihood across all data points.

While the WMAE is relevant for the prediction accuracy, we can also directly evaluate the estimated uncertainty. For this, we calculate the variance of the ground truth within an integration interval and compare it with the estimated variance, which should be similar. We use the Mean Square Error MSE_{var} between true and estimated variance. Please note, that this metric is only meaningful in combination with the prediction error.

Finally, we evaluate whether the generated data (which is partially conditioned on observed integral measurements) matches the process assumptions: We know that the integral of the ground truth results in the observed integral measurement. Therefore, the integral of generated data should also result in the observed integral. Here, we calculate the Mean Absolute Error MAE_{int} between the integral measurement and the integral value of generated data. We calculate the integral value of generated data by sampling data at the same points in time used to calculate the ground truth integrals and then apply the same

windowing procedure, i.e., summing up all the resultant values within the integration time interval.

4.6.3. Experiment Procedure and Model

We evaluate each kernel using the same Gaussian process model. As a model, we use the standard GPy implementation [GPy12]. We also train all configurations in the same way, using the GPy gradient-based maximum likelihood optimizer in its standard configuration [GPy12]. In this configuration, the optimizer performs 5 independent optimization runs, each time with random start kernel parameters. The model then uses the parameters from the best run.

For each dataset, we repeat this procedure 20 times, each time with a different ground truth time series. We then calculate the mean and the variance across the runs of the respective metrics.

4.7. Evaluation

We provide a quantitative comparison between our BIK and multiple baselines across multiple datasets in Section 4.7.1. Further, we feature additional qualitative kernel comparisons and visualizations in Section 4.7.2. These are especially useful for domain experts who like to attain an intuitive/visual understanding of the superiority of the BIK against other kernels. Further, we provide an ablation study of process and kernel parameters in Section 4.7.3.

4.7.1. A Quantitative Comparison of BIK and Baselines

A comparison of the different approaches, in Table 4.1, shows that BIK is superior to its competitors in every regard evaluated: The integral of generated data has at least 10 times lower MAE_{int} compared to all baselines. (The RBFIK from [FTM19] cannot generate data.) For our BIK, we provide a more detailed analysis of MAE_{int} in Section 4.7.3.

Regarding the prediction with uncertainty, BIK has at least 2 times better WMAE on *Load* and *Synth* $w = 25$ than with all competitors. On *Synth* $w = 50, 100, 200$ the baselines catch up a bit, but BIK still beats them by 30% less error. Also, the WMAE variance suggests that the baselines are very unstable across data sets.

The MSE_{var} between predicted variance and actual variance during measurement on *Synth*, *HIPE*, *Stock* is 2.6 times better for BIK than any baseline. On *Load* BIK is still 10% better.

Note that the results of BK and BNK are similar on synthetic data. This similarity arises because BNK learns a noise of zero. Indeed, BNK only gives good uncertainty estimates when trained on several different observations of the same data point (or with a smaller time step). With integral measurements, such data cannot be observed, leading to BNK learning incorrect small variances.

One may also notice that the variance in predictive variance (measured with MSE_{var}) is high across the experimental runs for all kernels. This is to be expected, since by chance the ground truth is sometimes simply close to the average within the integration interval, leading to a high deviation. Even with this high deviation, MSE_{var} is still meaningful for comparing the quality of the uncertainty quantification as long as the prediction error is similar. Even though BNK estimates a similar uncertainty as BIK, the metrics MAE_{int} and WMAE show that the uncertainty estimation of BNK is meaningless because of its much higher prediction error.

Table 4.1.: Metrics for different kernels and data sets. Bold entries mark the best result within one metric.

Data	Kernel	Metrics MSE_{var}	MAE_{int}	WMAE
Synth w=25	BIK	**5.8 ± 12.**	**7.6 ± .45**	**3.8 ± 2.9**
	RBFIK	24.9 ± 25.	—	44.2 ± 60.
	BK	15.6 ± 20.	72.8 ± 4.6	6.9 ± 133
	BNK	15.6 ± 20.	73.0 ± 6.0	6.9 ± 133
Synth w=50	BIK	**5.9 ± 14.**	**3.67 ± .26**	**2.3 ± 2.0**
	RBFIK	24.6 ± 27.	—	13.5 ± 19.
	BK	20.8 ± 25.	71.1 ± 5.8	3.38 ± 42.
	BNK	20.8 ± 25.	72.4 ± 4.3	3.38 ± 42.
Synth w=100	BIK	**4.7 ± 13.**	**2.0 ± .14**	**4.3 ± 3.3**
	RBFIK	20.6 ± 25.	—	44.8 ± 62.
	BK	17.8 ± 23.	68.5 ± 5.2	5.95 ± 89.
	BNK	17.8 ± 23.	70.0 ± 4.7	5.95 ± 89.
Synth w=200	BIK	**5.4 ± 9.2**	**1.12 ± .09**	**4.3 ± 3.5**
	RBFIK	22.4 ± 19.	—	50.9 ± 71.
	BK	20.6 ± 18.	63.2 ± 5.4	5.50 ± 51.
	BNK	20.6 ± 18.	62.9 ± 4.2	5.50 ± 51.
Load	BIK	**.50 ± 1.2**	**.39 ± .006**	**.35 ± .74**
	RBFIK	.64 ± 1.6	—	3.5 ± 7.4
	BK	.66 ± 1.6	3.3 ± .07	12. ± 26.
	BNK	.56 ± 1.4	3.7 ± .08	.80 ± 1.7
HIPE	BIK	**12.5 ± 26.**	**.69 ± .03**	**0.4 ± 0.3**
	RBFIK	43.1 ± 43.	—	8.02 ± 6.5
	BK	45.5 ± 44.	5.84 ± .21	518. ± 429
	BNK	45.5 ± 44.	5.69 ± .19	476. ± 395
Stock	BIK	**1.1 ± 2.9**	**14.8 ± 1.1**	**0.3 ± 0.3**
	RBFIK	3.7 ± 5.1	—	4.48 ± 3.9
	BK	4.0 ± 5.3	63.1 ± 3.2	46k ± 38k
	BNK	4.0 ± 5.3	60.9 ± 2.6	45k ± 38k

4.7.2. Providing Intuition about the BIK by A Comparative Visualization with BK

For domain experts, it is interesting to understand in which scenarios our BIK brings benefits. In this regard, we provide intuition by visualizing and discussing the similarities and differences between a Gaussian process model (*Brown*) with the BK and a model (*Int_Brown*) with our BIK. For brevity, we omit BNK from this visual comparison because the quantitative evaluation in Section 4.7.1 has shown that this kernel is nearly equivalent to the BK for integral measurements.

We examine the estimates and estimated variances as well as the synthesized data. For this, we visualize the fitted relation and its variance together with the integrated measurements and true underlying data, as well as generated load profiles that satisfy the conditions imposed by the observed integral measurements and the used kernel.

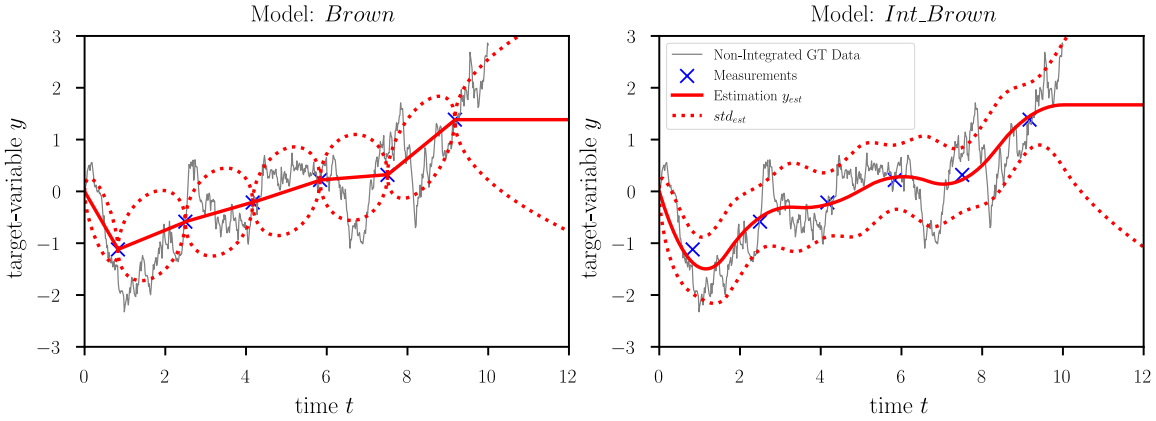


Figure 4.4.: Comparison between estimated variance v_{est} of the two kernels; 4.4a the standard BK; 4.4b our new BIK.

Figure 4.4 illustrates the differences between the two models in estimation and estimation variance v_{est} (shown as a ▬ dotted line). We integrate Brownian motion data (*Synth* data), which gives us integral measurements as by definition 4.1 (marked as ×). These measurements are inherently imprecise due to integration.

With integral measurements, the most likely function should be a smooth function. This is because integrating over time intervals acts like a sliding window smother. Comparing the fitted functions (depicted by the ▬ line) in terms of smoothness, the *Brown* model exhibits roughness, whereas the function modeled by *Int_Brown* is smooth as expected.

Furthermore, in Figure 4.4a, the *Brown* model erroneously estimates a variance of zero at measurement locations, failing to account for the fact that the measurements are integrated. In contrast, *Int_Brown* with our BIK in Figure 4.4b acknowledges the drift that occurs during measurement periods. It incorporates this influence by modeling the resulting inaccuracy with a higher variance. The maximum variance between measurement locations remains roughly consistent for both models. This is expected for points far away from actual measurements because the underlying Brownian motion is the same.

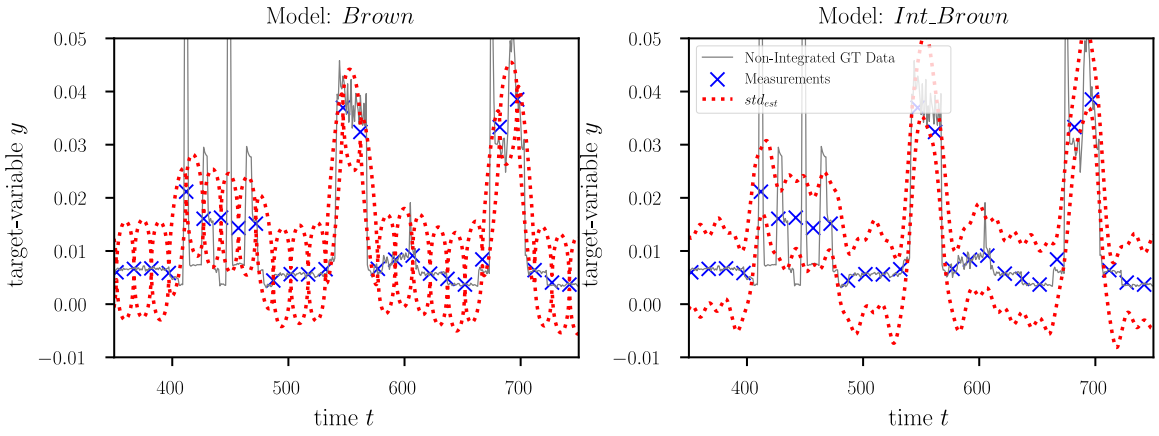


Figure 4.5.: Comparison using *Load* Data; 4.5a the BK; 4.5b our new BIK.

In Figure 4.5, we observe similar patterns in the synthetic load profiles, which are integrated over 15-minute intervals (marked as ×). Additionally, we display the high-resolution GT load profile with a one-minute resolution (depicted by the ▬ line). Upon comparing the actual load profile with the predicted one, it is evident that the true profile seldom aligns precisely with the integral measurement points. The *Int_Brown* model accurately captures this, exhibiting high variance in the vicinity of these locations.

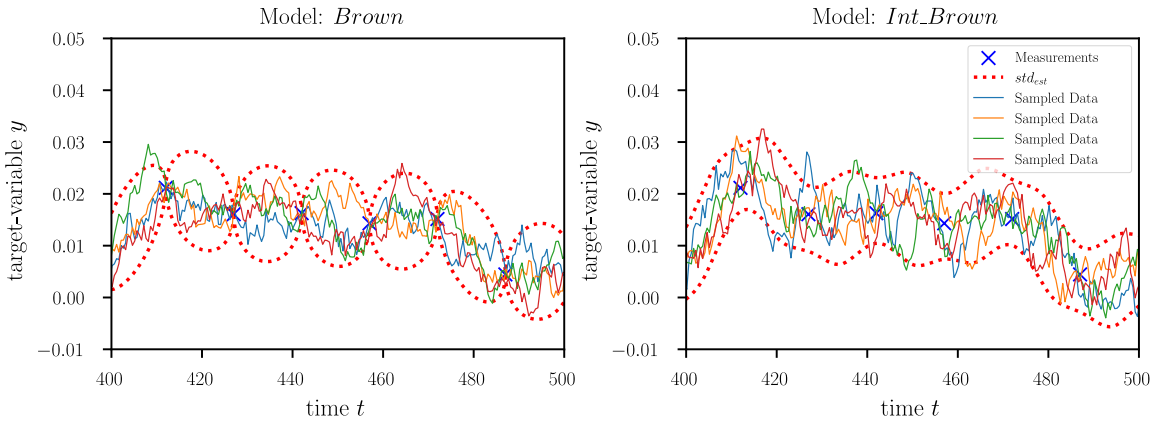
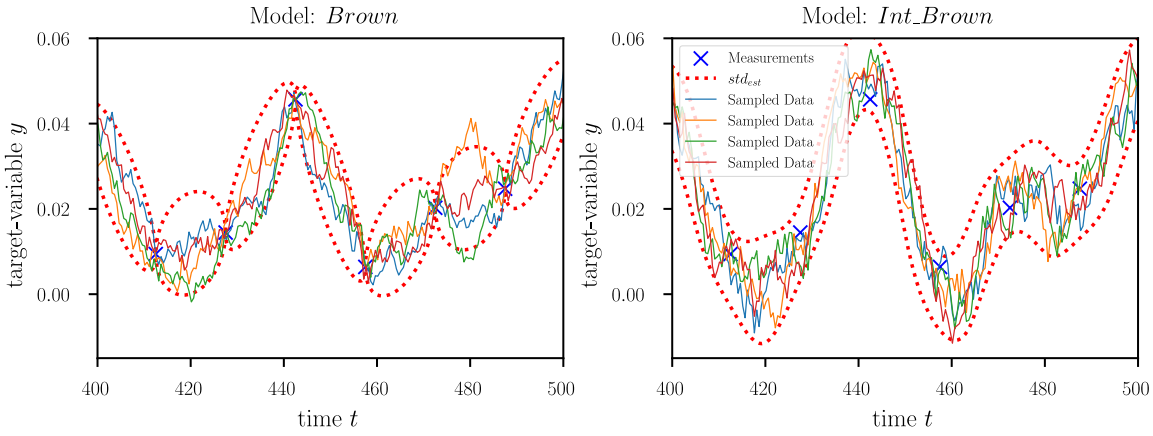


Figure 4.6.: Data generation from prior conditioned on *Load*; 4.6a data generated with BK passes through measurement locations which is wrong; 4.6b data generated with our BIK. It does not need to go through measurement locations, but its integral is guaranteed to be the same as what was measured.

In Figure 4.6, we see the ability of the GPs to synthesize data partially conditioned on measurements. Both models (*Brown* Figure 4.6a, and *Int_Brown* Figure 4.6b) are conditioned on the data presented in Figure 4.5. With *Brown*, all the generated time series pass through the measurement points (marked as ×), which is incorrect. Integrating the generated data gives random values that are not equal to the observed measurements. In

contrast, *Int_Brown* correctly generates data that does not necessarily intersect with the measurement locations. The integration of this data, however, yields the correct measured value. Therefore, the data generated with our BIK reflects reality more accurately.

In Figure 4.7, we trained the models on real-world data (*Real*) and generated potential load profiles. The behavior of load profiles produced by our *Int_Brown* model, as shown in Figure 4.7b, is plausible, as integrating them again yields the measured data, this is not the case for data generated with *Brown*.



(a) (b)

Figure 4.7.: Data generated from a model conditioned on real-world data (*Real*); 4.7a integrating data from the BK does not reflect real behavior; 4.7b data from our BIK is much more likely in the real world, and integrating the data results in the measured data.

4.7.3. Ablation Study of Integral Window Size and Data Variance

In the following, we will examine how properties of the underlying physical process impact the estimation and the estimated uncertainty. First, we will discuss how the integral window size, i.e., the time constant of the integrator of a physical process, reflects on estimations. Secondly, we will examine the difference between Brownian motions with different variances – equivalent to different drift speeds of the Brownian motion.

4.7.3.1. Impact of the Integral Window Size

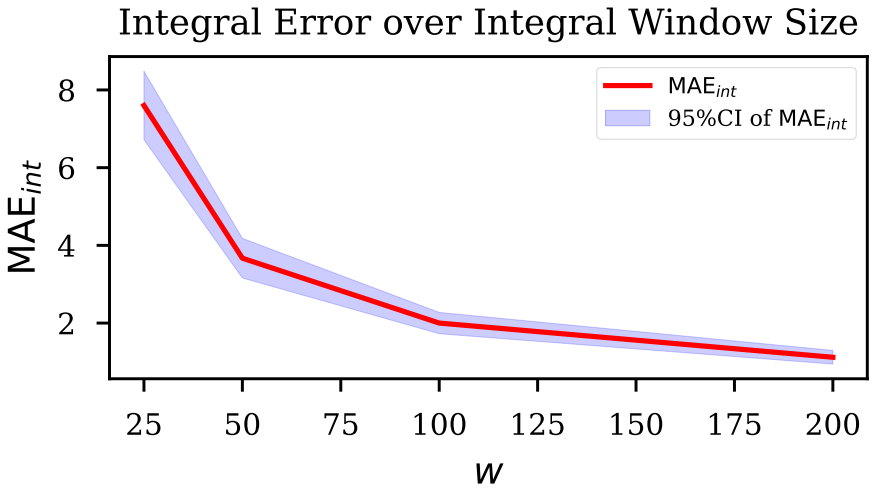


Figure 4.8.: Comparison of how changing the integral window size changes the integration error of generated data.

An observant reader may notice that the MAE_{int} decreases with increasing window size w. See Figure 4.8. We examined this behavior in depth and found that it is an artifact of the metric calculation rather than of our kernel:

We calculate the metric over a fixed number of generated points in time, equal to the points provided by the original GT data, i.e., w points. We then sum up the values at these points to approximate the integral measurement and calculate the error in relation to the ground-truth integral. However, as the generated data is continuous, using a fixed number of points will always introduce an error in the integral approximation. The smaller the number of points, the greater the error and the larger the approximation variance (refer to the blue region in Figure 4.8). Indeed, in our experiments, we found that maintaining the same start and end times for integration while increasing the number of points within the interval reduces the metric error. The error approaches zero for large w, which aligns with our expectations for an exact kernel like ours – integrating the data generated with the kernel yields the ground-truth integral value.

4.7.3.2. Impact of Data Variance

The previous observation in Section 4.7.3.1 has shown us that our kernel gives exact solutions for the same ground truth but with different integration intervals. Now we will

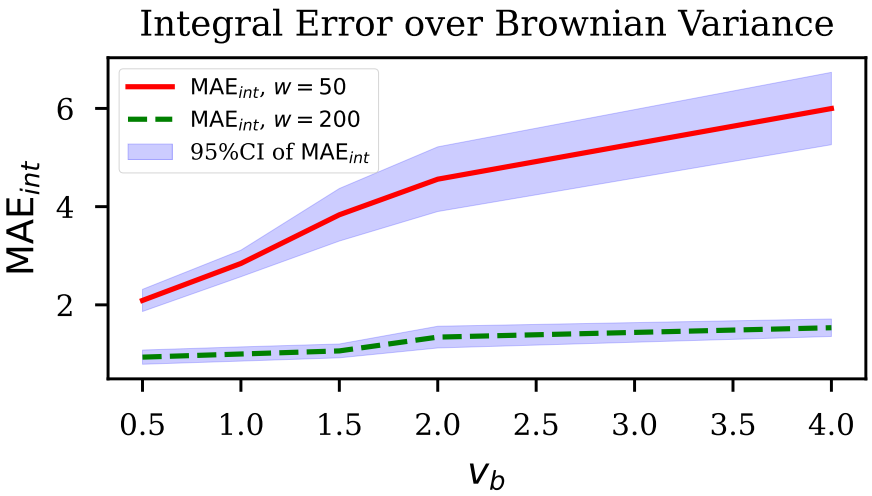


Figure 4.9.: Comparison of how Brownian GT data with different drift speeds v_b changes the integration error of generated data.

examine how changing the speed of the Brownian motion, i.e., the Brownian variance v_b will impact MAE_{int}. In Figure 4.9 we see that increasing v_b while keeping $w = 50$ constant increases MAE_{int}. This behavior is again caused by the metric calculation. The Brownian motion within the same interval now behaves more erratically, requiring more evaluation points to calculate the integral with the same precision. Increasing the evaluation points ($w = 200$) (while keeping training points the same) again leads to a declining error approaching zero for any v_b, which is the expected behavior for an exact kernel.

4.8. Chapter Conclusion

In this chapter we investigated the uncertainty introduced by integrated measurements of a Brownian DGP. Such integrated Brownian motions are crucial in many physical processes and data collection techniques. The conventional BK, while effective for modeling Brownian motion, falls short in capturing the uncertainties associated with integrated data.

This study has tackled modeling integrated Brownian motions precisely, by providing an analytical solution to the novel Brownian Integral Kernel (BIK). The BIK enables precise estimation of variance associated with the underlying quantity of interest. Further, the BIK is a valuable tool for tasks like regression with uncertainty estimation and for data synthesis partially conditioned on measurements, as shown in our experiments.

Our contributions bridge a significant gap in modeling integrated processes. Our BIK enhances the accuracy and reliability of GP modeling in such uncertain and dynamic environments by a factor of at least 2 on every dataset and against every baseline. Data synthesis with our integral kernel is better by a factor of at least 10 compared to all baselines.

While our BIK is a substantial advancement, there are avenues for further exploration: Investigating the properties of other integrated processes besides the Brownian integrated

process could result in additional new kernels useful for modeling such processes. Further, using our kernel in applied research could lead to advances in several directions, such as estimating privacy violations and disaggregating load data from smart meters. Finally, our kernel could bring better uncertainty estimates for the challenge of concept drift, which we investigated in the previous chapter (Chapter 3).

Part V.

Domain Knowledge Integration

5. How Domain Knowledge Can Improve Machine Learning Surrogates

The content of this chapter bases on the following publication:

- Bela H. Böhnke et al. 'How Domain Knowledge Can Improve Machine Learning Surrogates for Manufacturing Process Optimization – a Comparative Study'. In: *Procedia CIRP*. 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024) 130 (2024), pp. 145–153. DOI: 10.1016/j.procir.2024.10.069

Keywords: Surrogate Modelling; Manufacturing Process Optimization; Domain-informed Machine Learning; Finite-Element-Simulation; Domain Knowledge Integration; Physics-guided Machine Learning; Theory-guided Data Science

5.1. Chapter Overview

In the previous chapter (Chapter 4) we needed some initial assumptions to quantify the uncertainty. In this chapter we investigate how additional domain knowledge can be included into a model, and how the additional knowledge can improve the generalization capability of a model thereby decreasing uncertainty and increasing data efficiency. We do this in the context of Surrogate Model-based Optimization (SuMO) for the task of industrial manufacturing processes optimization:

Industrial manufacturing processes require careful parametrization for optimal operation in terms of part quality, throughput or efficiency. In current practice, identifying optimal parameters often involves lengthy trial-error campaigns and significant rework for fault correction. High-fidelity process simulations, e.g., based on the Finite Element Method (FEM), allow to assess manufacturability at the earliest stages of part development [Mou20]. Besides rigorous analysis of process dynamics, their inherently virtual nature also allows for coupling with optimization algorithms, often referred to as *virtual process optimization*. While such a coupling enables automated search for optimal parameters, the computational demands of iterative optimization often makes it impractical in real-world applications [Zim+21].

One option to reduce the computational burden in virtual process optimization is SuMO [KL13]. Multiple variants of SuMO exist, which all share the idea of constructing a computationally efficient, data-driven approximation of the expensive simulation – a *surrogate*. This process is referred to as *training* and relies on a priori sampled observations. The resultant surrogate then guides the optimizer in the parameter space.

The more accurate the surrogate, the more efficient the optimization process will be. Since accuracy generally improves with training data, a naïve idea could be to supply more sample simulations. However, the available computational resources usually limit the number of simulations. This in turn makes data-efficiency the key for real-world applicability.

In this study, our objective is to enhance surrogate accuracy by leveraging readily available engineering knowledge while maintaining a constant number of simulation samples. We stepwise introduce additional knowledge by domain-agnostic and domain-informed methods and compare the impact on surrogate accuracy. Additionally, we categorize different types of additional knowledge regarding knowledge complexity and assess the difficulty of incorporating the knowledge. Further, we discuss the transferability of our methods to other domains. We provide our implementation together with additional experiments online[1] to facilitate reproduction.

5.2. Related Work

Data-driven surrogate models can be used to guide the optimization and to identify promising candidate solutions at low computational effort [Gor+10a]. However, due to their statistical nature, they always deviate from the original process, and thus, these candidates may differ from the true optimum of the original process. SuMO tries to sequentially eliminate this bias of the surrogate model by iteratively refining the surrogate model with new observations over the course of optimization [Bou+19]. Over the last decades, research on surrogate modeling mostly studied the suitability of different models ranging from simple polynomials and regression trees [Sma+18] to stochastic processes [RL21; HKZ20] and artificial neural networks [Sim+01]. Irrespective of the actual model, the studies tend to view surrogates as a phenomenological *input-output*-relation, where adjustable process parameters (*input*), e.g. temperature or pressure, are mapped to a scalar or low-dimensional quality metric (*output*) [KL13].

However, spurred by advances in Machine Learning (ML), attempts have been made to process – and also predict – more complex information with data-driven models. For instance, data preprocessing steps like principal component analysis have been introduced to find an information-rich input-space representation [Lia+18]. Alternatively, techniques have been studied that do not just output a scalar value but instead predict multi-dimensional quantities, i.e., a full-field estimation of the quality. For applications in material forming, see, e.g., [Pfr+18; Zim+21; Goo+21].

Overall, the literature shows that the introduction of additional information increases surrogate accuracy. However, most works view this from a methodological perspective, i.e., seek to improve accuracy by algorithmic improvements but tend to disregard other information sources. Such sources can be domain knowledge [Kar+17a; Wil+20] about material behavior or spatial dependencies, but as of now, no systematic investigations for materials science have been reported.

[1] https://github.com/bela127/knowledge-surrogate-opt

5.3. Use Case: Textile Forming Optimization

This chapter considers optimization of the manufacturing process of composites, specifically Continuous-Fiber Reinforced Plastics (CoFRP). CoFRP offer unparalleled weight-specific mechanical properties and are thus increasingly applied across industries. However, their superior properties usually come at a substantial cost: Not only are the materials themselves expensive, also their complex behavior during manufacture entails considerable optimization effort to produce high-quality products. CoFRP-processes generally comprise a process chain with multiple steps [Kär+15]. While process parameters need to be optimized for all steps, this chapter focuses on forming (*draping*) engineering textiles – specifically woven fabrics.

Figure 5.1 visualizes the process. Initially, the fabric is cut, stacked, and transferred to a press tool. Upon tool closure, the textile takes up the desired three-dimensional shape. For subsequent handling, an binder material stabilizes the textile's shape. Then the restraining forces are released, the tool opens, and the formed textile is transferred to a resin injection tool for infiltration, curing, and demolding.

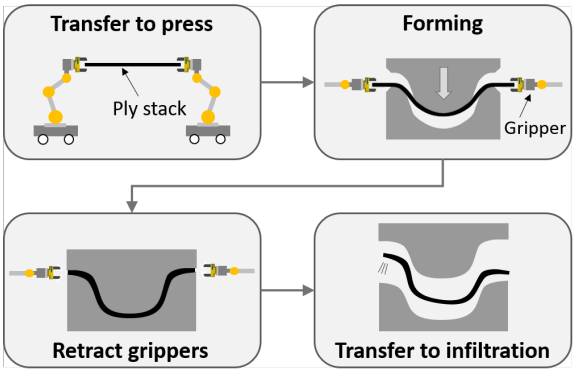


Figure 5.1.: Visualization of the draping process. [Zim+21]

This work revisits the virtual forming optimization problem from [Zim+21]. It studies an FEM-based forming simulation model of a double-dome geometry, a common benchmark geometry in textile forming. To control the process, 60 spring-guided grippers clamp the textile along its perimeter, as schematically shown in Figure 5.2. The grippers locally exert restraining forces onto the textile and thereby manipulate its draw-in into the mold. The optimizer can choose gripper spring stiffnesses c_j ($j = 1 \dots 60$) between $0.01 \dots 1.0\,\mathrm{N/mm}$.

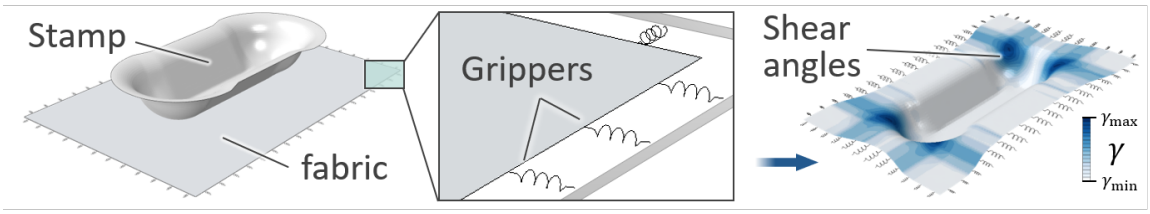


Figure 5.2.: Left: Forming simulation setup with 60 grippers along the textile perimeter, visualized by springs. Right: Example shear angle distribution after forming. Some springs stretch and locally introduce a restraining force. [Kär+18; Zim+21]

Due to their textile architecture, woven fabrics have a low shear stiffness compared to their tensile stiffness in warp and weft direction. This makes in-plane shear the domin-

ant deformation mechanism. One can quantify it by the shear angle γ, as visualized in Figure 5.3.

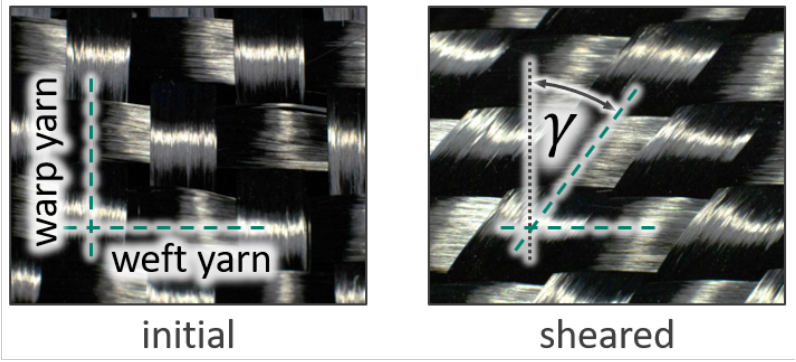


Figure 5.3.: Shear deformation of a woven fabric measured by the shear angle γ. [Zim23]

However, fabrics cannot undergo arbitrarily large shear deformations but show a forming limit, the *locking angle* γ_{lock} [Boi+18]. Shearing beyond γ_{lock} increases the likelihood of defects like wrinkling and textile folding. Also, high shear angles impede resin infiltration and may lead to uninfiltrated regions (*dry spots*) which can compromise the structural performance [EE04; Kär+18]. Therefore, the shear angle is a crucial quality indicator during the forming process, and it is typically minimized by finding the optimal gripper spring stiffness combination. Early work on virtual forming optimization uses the maximum shear angle $o(\boldsymbol{\gamma}) = \gamma_{\max} = \max \boldsymbol{\gamma}$ as the optimization objective [Che+15].

To ensure domain-specific legibility for materials science experts, we retain the domain-specific variable names $\boldsymbol{c}, \boldsymbol{\gamma}$, as replacing such standardized conventions with our global notation x, y would obscure the physical interpretation of the underlying draping process. The relation to the Data-Generating Process (DGP) is as follows: Interventional value or model input $x = \boldsymbol{c}$ is the vector of spring stiffnesses c_j ($j = 1 \dots 60$), and the observed output variable $C(\boldsymbol{c}) = y = \boldsymbol{\gamma}$ is a set of shear angles. For a given gripper configuration n, the shear strain is pixel-wise encoded in an image $\boldsymbol{\gamma}_n = (\gamma_{n1}, \dots, \gamma_{nP})$ with P being the pixel count. In this setup we have a deterministic FEM-based forming simulation, thus we do not have random variables as inputs or outputs.

5.4. Considered Domain Knowledge and Inclusion Methodes

We investigate the effect of domain knowledge on surrogate accuracy. As Table 5.1 summarizes, we study three different approaches to include domain knowledge. We selected our approaches to cover typical levels of complexity regarding: (1) Contained domain knowledge, (2) required effort to include the knowledge, and (3) transferability to other manufacturing processes. Our approaches are outlined in the following.

Table 5.1.: Investigated domain knowledge; categorized according to our findings

Example	Knowledge	Inclusion	Transferability
Geometry-strain relation	simple	simple	general
Gripper-tensile-force relation	complex	complex	specific
Objective Alignment (OA)	complex	simple	general

5.4.1. Geometry-Strain Relation

In manufacturing and general engineering, we can typically expect a complex relation between component geometry, manufacturability and structural performance.

Knowledge One such relation stems from the deformation mechanism of woven fabrics: Shear deformation will mainly form in doubly-curved geometry regions. That is, a close spatial relation between geometry and high shear angles can be expected. We deem this rather broad and qualitative statement a comparably simple form of domain knowledge. It 'only' requires a description of the geometry and the part quality distribution, here the set $\boldsymbol{\gamma}$ of shear angles.

Inclusion For textile forming, this has proven comparably straightforward: As proposed in [Zim+19; Zim+22] and later confirmed in [Vii+23], images are well-suited to describe such spatial relations in textile forming: One can encode the tool geometry and material shear as a grayscale image using the local elevation from the tool separation plane and, likewise, the local shear angles as shown in Figure 5.4. At the same time, images are suitable data formats for ML techniques, which makes incorporation straightforward.

Transferability We hypothesize that a spatial-aware surrogate model achieves better generalization performance than its classical *input-output* counterpart. This is because the spatial-aware model requires less training data to achieve the same accuracy. Such qualitative geometry-process-relations are widespread in manufacturing and general engineering: Consider, for instance, fiber reorientation along the flow paths in molding processes or stress concentrations at geometrical notches. Thus, we expect good transferability to other domains.

5.4.2. Gripper-Tensile-Force Relation

We further utilize domain knowledge to encode the positions and the area of influence of the grippers. The grippers actuate the textile locally, and thus, we again expect a close spatial relation between grippers and the textile response.

Knowledge A major (1) and a minor (2) mechanism is assumed to govern the effect of grippers on the fabric: (1) The continuous fibers can transmit large tensile loads along their axis, and thus, each gripper will actuate the fabric yarns connected to its point of attack in a line-wise manner. (2) Shear forces are transmitted to neighboring yarns via

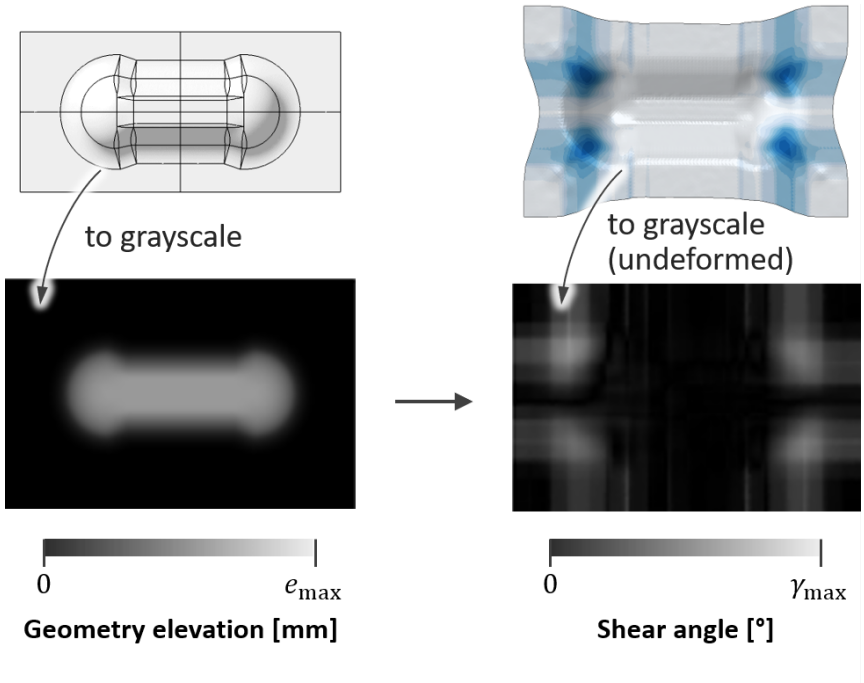


Figure 5.4.: Encoding the tool geometry (top left) and material shear (top right) as grayscale images (bottom).

friction at the warp-weft-crosspoints, although on a much lower scale. However, as the frictional forces at the crosspoints add up along the fiber axes, more and more neighboring yarns are actuated. As a result, the grippers' area of influence will spread to a certain extent along its line of action. We deem these material-specific mechanisms a complex form of knowledge.

Inclusion We resort again to image representations and consider two possible approaches: a *stiffness*-encoding and, extending it, a *force*-encoding. Both encodings seek to approximately represent the two material mechanisms. Since the non-linear, multi-scale behavior of woven fabrics defies a rigorous mechanics-based, closed-form model of the gripper effects, we lean on simplified elasticity theory for orthotropic materials. Please note that the following is not a rigorous derivation based on continuum mechanics but a pragmatic adaptation to comply with engineering understanding. We assume that the gripper influence behaves roughly similarly to the stiffness of a unidirectional fiber under rotation. That is, it is maximal along the fiber axis (reduced plane-stress stiffness Q_{11}) but diminishes with relative rotation α to the fiber axis (Figure 5.5). See, e.g., [Öch23] for the transformation equations of the reduced stiffnesses Q_{ab} under rotation, where a, b are the indexes of the conventionally used Voigt notation.

In our study, we are more focused on relative values than absolute ones. Therefore, we normalize the stiffnesses relative to the stiffness maximum Q_{11} between 0 and 1 via $q_{\text{rel}} = k\, Q_{ab}/Q_{11}$. This normalization process employs an attenuation factor k, which reduces the signal's intensity exponentially in further distance h from the fiber direction and reflects the lower friction forces. Specifically, we set $k = \exp\left(-\left(h/l_{\text{aff}}\right)^2\right)$. l_{aff} represents a length which

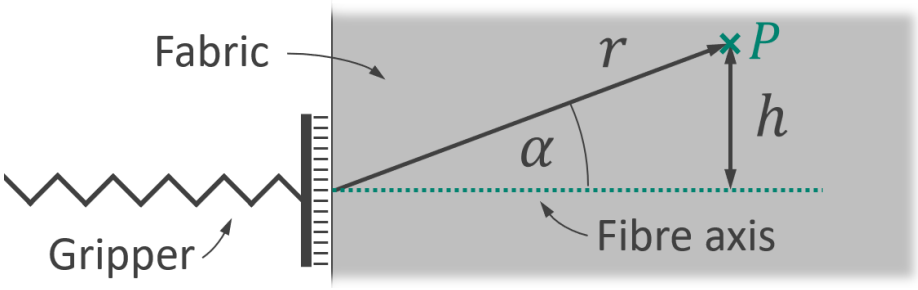


Figure 5.5.: Relative rotation angle of an affected material point P and fiber axis.

increases in proportion to the distance r: $l_{\text{aff}} = [1 - \exp(-r/r_{\max})] \cdot [l_{\text{aff}\,\infty} - l_{\text{aff}\,0}] + l_{\text{aff}\,0}$ with $l_{\text{aff}\,\infty} = 30\,\text{mm}$, $l_{\text{aff}\,0} = 10\,\text{mm}$ and $r_{\max} = 200\,\text{mm}$. The relative stiffness q_{rel} models the grippers' areas of influence while the spring stiffnesses c_j scale them up and down so that we obtain the *stiffness*-encoding I_{C} via $I_{\text{C}} = c_j \cdot q_{\text{rel}}$.

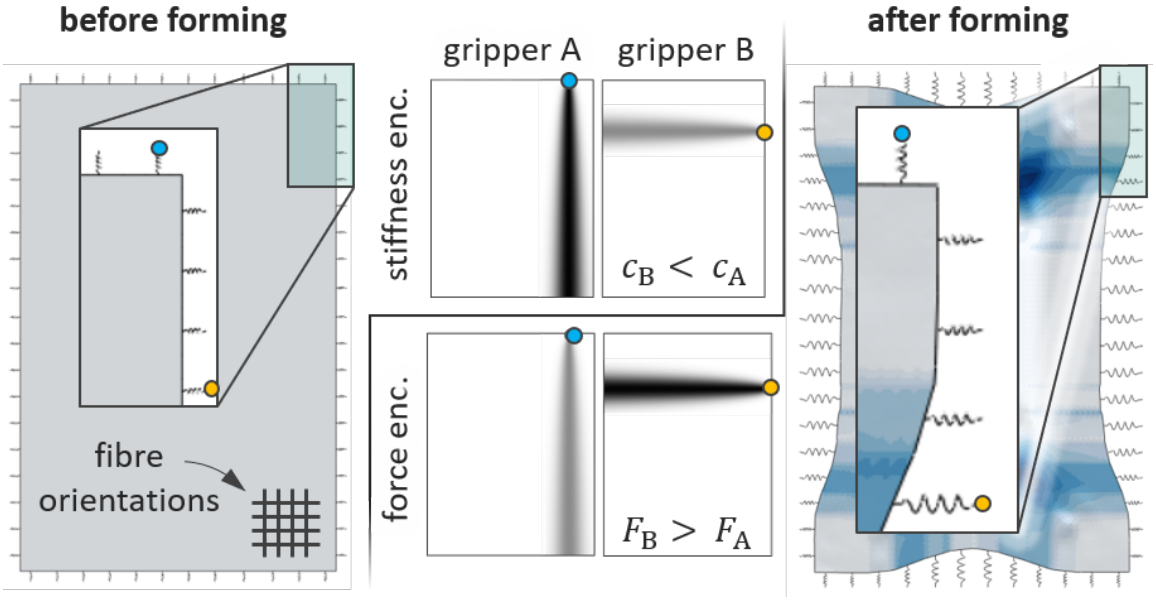


Figure 5.6.: Gripper stiffness and force impact distribution I_{C} and I_{F}. Gripper forces F_j computed from stiffness c_j and reference-displacement u_{ref} by $F_j = c_j \cdot u_{\text{ref}}$.

We give two encoding examples for I_{C} in Figure 5.6 (center top), one with a high stiffness (dark) and one with a lower stiffness (bright). Clearly, the distributions reproduce the engineering understanding: From the grippers' points of attack (blue and yellow markers) I_{C} is maximal in warp or weft direction, respectively, and gradually widens in perpendicular direction.

To obtain the *force*-encoding, we extend the stiffness encoding. Consider the situation shown on the right of Figure 5.6. Spring A (blue marker) is barely stretched, while spring B (yellow marker) experiences considerable stretch, i.e., $u_{\text{B}} >> u_{\text{A}}$. Since the exerted force F of a gripper depends not only on its spring stiffness c_j but also on its stretch u ($F_j = c_j \cdot u$), the stiffness of each griper contributes to the forming process to different extents. Thus,

we obtain the force-encoding $I_F = I_C \cdot u_{ref}$ by multiplying the gripper stiffness I_C with a reference-displacement u_{ref} to factor in the expected spring stretch during forming. Here, u_{ref} comes from an additional forming simulation, where all gripper springs have a uniform stiffness of $c_j = 0.5\mathrm{N\,mm^{-1}} \forall j \in 1 \dots 60$. This involves the assumption that the gripper-displacements remain approximately constant across simulations. By comparing the encodings of I_F (cf. Figure 5.6 center bottom) and I_C (cf. Figure 5.6 center top) it becomes apparent that the force-encoding of the gripper-tensile-force relation is different depending on how much the fabric is locally drawn into the mold: Although spring B has a lower stiffness than spring A ($c_A > c_B$), it stretches much more ($u_A << u_B$) and thus exerts a higher force during forming. Consequently, in the force-based encoding, spring B is more pronounced.

As with the geometry-strain relation, including the image-based encoding into the surrogate is easy. However, we deem the incorporation approaches complex because of the mechanical understanding required to obtain the encoding.

Transferability These image-based encodings are highly process-specific, because of the material-specific deformation mechanisms. Thus, they may not be directly applicable to other processes, although it is certainly conceivable to devise different representations for other processes.

5.4.3. Alignment of training and optimization objective

Over the last decades, surrogate models were mainly used to construct phenomenological *input-output*-relations, where adjustable design variables (*input*) are mapped to a scalar merit function (*output*), often an objective function for optimization [KL13]. Optimization amounts to minimization of this scalar objective function value. Historically, surrogate training aims at predicting this scalar objective function value accurately. However, an accurate prediction of the objective function is important only near minima. Thus, predictions close to minima are more important and require highest-possible accuracy. Classical surrogate training does not reflect this difference in importance, though, but weighs all data equally. Recent work has shown that full-field predictions are beneficial for accuracy [Goo+21; Zim+21] because they allow learning relations between neighboring regions. However, full-field predictions instead of scalar objective functions even compound this difference in importance. This is, because in a full-field many elements contribute only little to the objective function, making their accurate prediction less important. However, current work weighs them equally to the – often few – elements that contribute considerably to the objective function. We name this issue *training-objective-bias*.

Knowledge The objective function already contains well-formalized domain knowledge about what constitutes part quality and possibly defect allowables. Specifically, the objective function quantifies the importance of different elements with respect to the overall part quality. For the n-th gripper configuration, the shear strain is pixel-wise encoded in an image $\boldsymbol{\gamma}_n = (\gamma_{n1}, \dots, \gamma_{nP})$ with P being the pixel count. In accord with [Pfr+18], we assume the norm $o(\boldsymbol{\gamma}_n) = \|\boldsymbol{\gamma}_n\|_k = (\sum |\gamma_{np}|^k)^{1/k}$ with $k = 4$ as the objective which balances

suppression of maximum shear and general shear formation. By incorporating this – often complex – knowledge into the surrogate, we expect to identify important regions and reflect their importance during training.

Inclusion We propose a novel method – *Objective Alignment (OA)* – that makes use of this knowledge during model training. OA seeks to align the training objective with the optimization objective, thereby counteracting the training-objective-bias. During training, the pixel-wise loss of the shear strain field (Figure 5.4, right) is weighted by its pixel-wise importance with respect to the objective function. We argue that the importance of a ground truth value γ_{np} of image n and pixel p is quantified by the influence W_{np} the value γ_{np} has on the objective function $o(\boldsymbol{\gamma}_n)$, calculated over the whole ground truth strain field $\boldsymbol{\gamma}_n$. We quantify this influence W_{np} via backpropagation as the gradient and then normalize the overall importance matrix W_n with a min-max normalization to the interval $[0.1, 1]$ to obtain pixel weights w_{np}:

$$W_{np} = \frac{\delta o(\boldsymbol{\gamma}_n)}{\delta \gamma_{np}}, \quad w_{np} = \left|_{0.1}^{1} W_{np} \right|_{\min(W_n)}^{\max(W_n)} \tag{5.1}$$

. Note that we do not normalize to 0 so that any pixel has at least some influence to avoid random predictions $\hat{\gamma}_{np}$. The obtained weights then quantify the contribution of each pixel to the overall objective-aligned loss. The Mean Absolute Error (MAE) (per image i) with OA correction reads:

$$\mathrm{MAE}_{\mathrm{OA},n} = \sum_{p=1}^{P} w_{np} \left| \gamma_{np} - \hat{\gamma}_{np} \right| \tag{5.2}$$

and analogously for other losses such as Mean Square Error (MSE), see Section 5.5.1.1.

Overall, we expect OA to reduce the necessary amount of calibration data, i.e., surrogate refinement iterations, and allow for faster identification of optimal parameters. OA is applicable to any differentiable objective function without further engineering effort. Thus, we deem OA a simple incorporation method.

Transferability Unlike our domain encoding (stiffness-/force-encoding), the inherent domain knowledge about the optimization objective is readily available in any automated parameter optimization context such as SuMO, and already formalized within the objective function. Thus, OA ought to be excellently applicable to objective functions from other domains. This makes it generally useful across disciplines.

5.5. Ablation Studies and Results

Having outlined the envisaged domain knowledge and suitable domain-based inclusion methods, this section presents the setup of our numerical studies and the observed effects on surrogate accuracy and performance for SuMO. First, we introduce the common study setup. We then discuss various surrogate architectures and existing domain-independent knowledge inclusion methods, which we use as baselines in our studies.

5.5.1. General Study Setup

We perform studies for all three types of domain knowledge similarly: We scale the data to an $[0, 1]$ interval with respect to the physically possible shear minimum (0°) and maximum values (90°). We train each investigated surrogate model with the Adam optimizer [KB15] with an initial learning rate of 0.001 and a batch size of 8. We perform each study on training sets of sizes between 100 and 900 to investigate how well the surrogate can generalize from different amounts of data. The generalization capability is especially important for SuMO, where each data point is costly, which is why our discussion of results will focus on smaller training set sizes up to 500. For each training set size, we perform a 5-fold cross-validation with a separate test set of constant size 100. For each validation we take the end results after early stopping with a 60-epoch patience period or after a maximum of 300 epochs. We then report their mean and 95 % confidence interval per training set size. Note: The results should not be confused with a learning curve, for which only the number of epochs varies, but the training set size stays the same.

5.5.1.1. Loss Functions for Neural Networks

For evaluation we use MAE, MSE, and Root Mean Squared Error (RMSE). These metrics are commonly used as loss functions and evaluation metrics to assess the quality of a trained network. This study further uses the RMSE on the objective function ($RMSE_{\text{obj}}$) and the *Area Under the Curve (AUC)* which not only quantifies the end result but also the convergence speed of the optimization to reach this result, defined as:

$$\text{MAE}_n = \sum_{p=1}^{P} \left|\gamma_{np} - \hat{\gamma}_{np}\right|, \tag{5.3}$$

$$\text{MSE}_n = \sum_{p=1}^{P} \left(\gamma_{np} - \hat{\gamma}_{np}\right)^2, \tag{5.4}$$

$$\text{RMSE}^2_{\text{obj},n} = \left(o(\boldsymbol{\gamma}_n) - o(\hat{\boldsymbol{\gamma}}_n)\right)^2 \tag{5.5}$$

$$\text{AUC} = \frac{1}{2} \sum_{i=1}^{I-1} (o(\hat{\boldsymbol{\gamma}}^*_{i+1}) + o(\hat{\boldsymbol{\gamma}}^*_{i+1})) \tag{5.6}$$

Therein, P is the number of pixels p in an image n, with γ_{np} denoting the true and $\gamma_{\theta np}$ the predicted value of the p-th pixel. Further, I is the number of optimization steps (iterations), and $\boldsymbol{\gamma}^*_{\theta i}$ is the strain field of the best product found up to the current iteration i. Averaging the per-image loss across all N images gives the total loss. In addition, $\text{RMSE}_{\text{obj}} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \text{RMSE}^2_{\text{obj},n}}$ requires taking the square root.

5.5.2. Geometry-Strain Relationship

The first kind of additional information we include in the surrogate model is the *geometry-strain-relation*, cf. Section 5.4.1. There we represent the geometry and the shear strain field by grayscale-images (Figure 5.4). This provides root-cause information (doubly-curved geometry regions) towards the formation of the shear field, which the grippers then manipulate. We accordingly select image-processing architectures for the surrogate, namely Convolutional Neural Network (CNN).

5.5.2.1. Surrogate Architectures

We use the Multi-Layer Perceptron (MLP) from the original paper [Zim+21] as our baseline (Figure 5.7). It directly takes the 60 spring stiffnesses c_j, $j = 1 \dots 60$ as input and estimates the full strain field. It neither has knowledge of spatial relationships or stamp geometry nor does it feature convolutional layers.

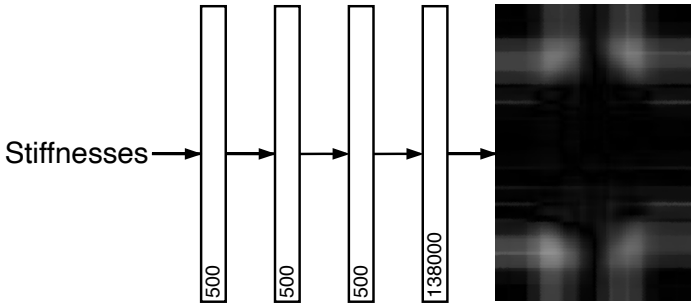


Figure 5.7.: The baseline MLP architecture from [Zim+21] without additional information.

We compare this baseline (MLP) to three CNN architectures for image-to-image tasks (cf. Figure 5.8): A classical encoder-decoder-architecture from [Zim+19; Zim23] for geometry-to-strain prediction, a *U-Net* architecture [RFB15], and the state-of-the-art *CFPNet-M* architecture [LGL23]. Note that the U-Net and the CFPNet use skip connections, which allow information to flow from previous layers to subsequent layers to prevent the problem of vanishing gradients in deep networks. The CFPNet additionally introduces the new *Channel-wise Feature Pyramid (CFP)* -modules, which facilitate learning of features of varying sizes.

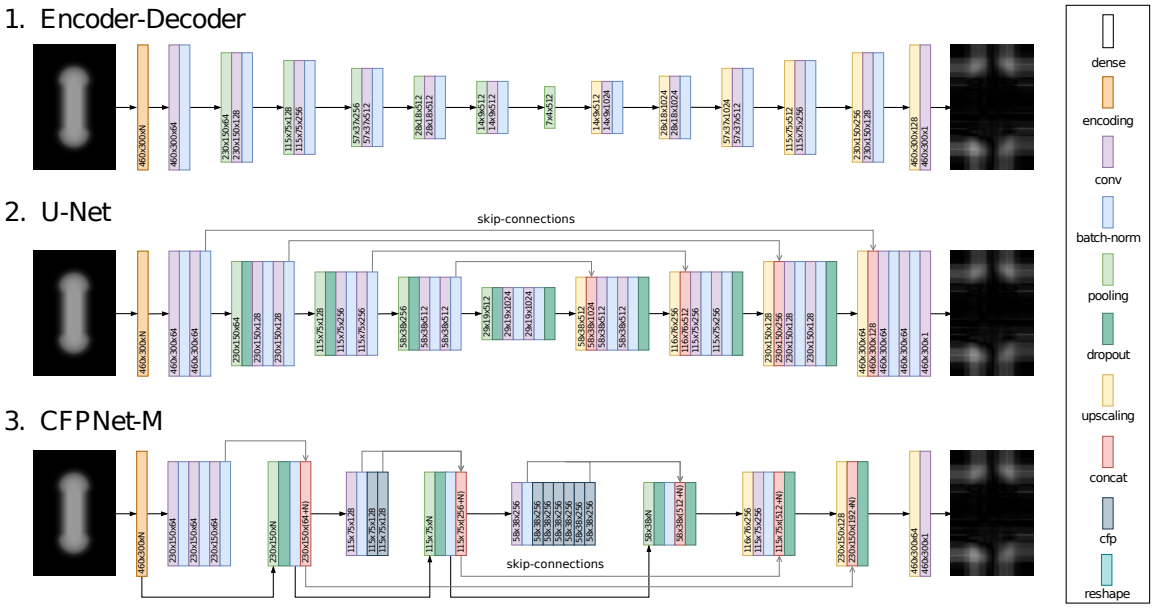


Figure 5.8.: Architectures of the models used for evaluation. The CFPNet architecture is the surrogate model of our choice because of its better overall performance compared to all other investigated architectures.

A regular multi-path architecture includes the non-image spring stiffnesses into the CNNs. As Figure 5.9 shows, they are fed through a separate network which is concatenated to the bottleneck stage of the CNNs. Specifically, we reuse the MLP from [Zim+21] for the non-image path.

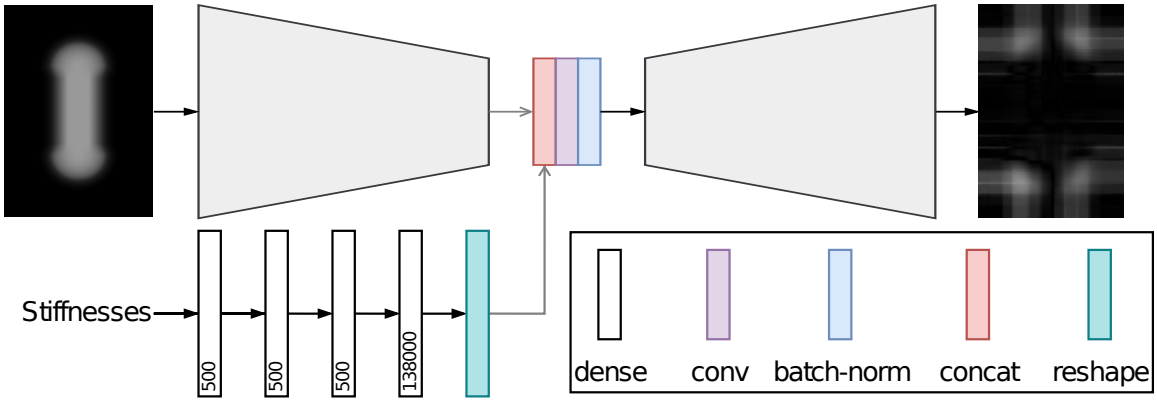


Figure 5.9.: The multi-path architecture for merging non-image data with image data.

We train all models with the MSE-loss and evaluate them after training in terms of MAE and $RMSE_{obj}$. See 5.5.1.1 for their definition. The $RMSE_{obj}$ considers the objective function and, thereby, is especially informative regarding optimization performance.

5.5.2.2. Results

Figure 5.10 visualizes the results. Clearly, U-Net and CFPNet outperform all other architectures when training on small data, i.e., ≤ 250 samples. For 250 data points, the MAE decreases by ≈ 65% and the RMSE_{obj} even by 75% of the MLP (baseline) with practically negligible scatter. The U-Net performs exceptionally well and almost reaches its maximum performance with only 100 data points.

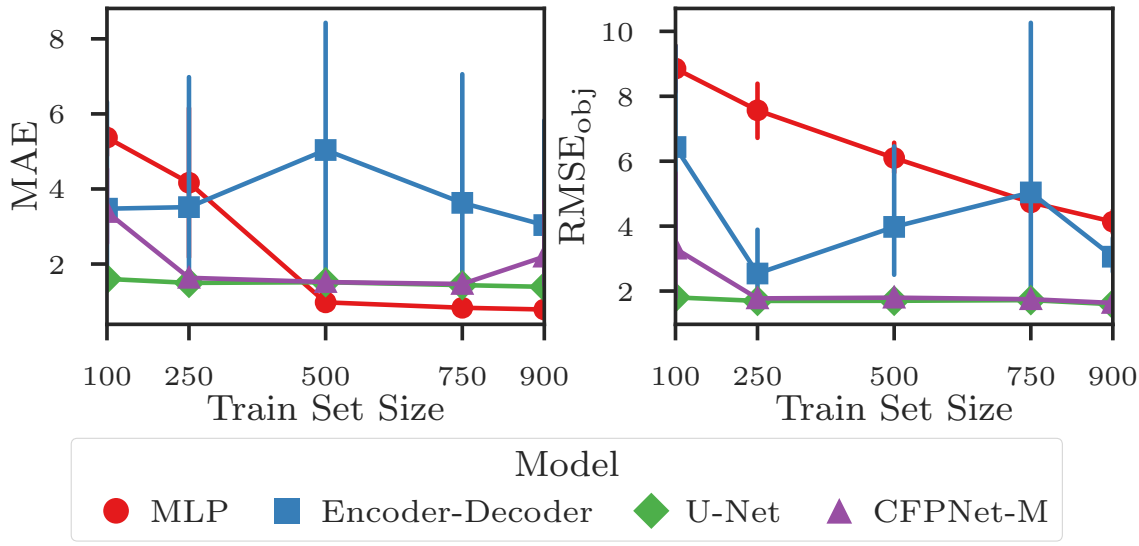


Figure 5.10.: (left) model comparison of MAE between predicted strain fields; (right) model comparison in RMSE based on the objective function. Small values imply better performance.

Interestingly, the MLP is able to catch up from 500 data points onwards and even outperforms the CNNs regarding the MAE. However, we expect the performance on RMSE_{obj} to be more informative regarding optimization performance, where the CNNs consistently keep their superiority. The encoder-decoder performance proves entirely unstable – presumably due to missing skip connections – and we exclude it from further investigations. Nonetheless, we can observe that every architecture that makes use of domain knowledge performs significantly better for small training set sizes than the MLP without domain knowledge.

5.5.3. Encoding of Grippers

We further study the effect of different gripper representations on surrogate accuracy. Specifically, we want to assess the suitability of our domain-informed gripper-tensile-force encodings I_{C} and I_{F} (Figure 5.6), respectively, compared to domain-independent encodings.

5.5.3.1. Gripper Encoding Methods Investigated

We study two CNN architectures, U-Net and CFPNet. Our baseline uses a vector-valued gripper encoding, which is fed into the CNNs via the multi-path architecture, cf. Figure 5.9.

We compare these baselines to models with 'image-only' gripper encoding. Specifically, we use two domain-agnostic encodings and our domain-informed encoding I_C and I_F. As domain-agnostic methods we use *naïve copy* and *DeepInsight+* to transform the vector-data to image-data without using domain knowledge. All tested models now make use of the geometry-strain-relation from Figure 5.4.1.

Naïve copy populates for each value of a vector a matrix of the desired image input size. In our case, this amounts to 60 matrices for the 60 gripper stiffnesses and to $2 \cdot 60 = 120$ matrices for their positions, i.e., an input shape $H \times W \times 3 \cdot 60$, with image height H and width W. While simple and easy to implement, this method can lead to excessive memory consumption, as it replicates values for each vector element.

DeepInsight+ is a combination of two state-of-the-art domain-independent techniques to transform non-image data into image data. The first technique is based on the finding that combining multiple data representations generally benefits learning. While the original work [SK22] proposes to combine three simple encoding schemes – *Row-Wise Copy*, *Distance Matrix*, and *Equidistant Bars* – we add the DeepInsight encoding taken from [Sha+19b]. Notwithstanding algorithmic details, DeepInsight utilizes the property that CNNs compute neighboring pixels together and that such pixels share information. The DeepInsight methodology automatically places similar features close together while distancing dissimilar ones. The placement of features involves a dimensionality reduction on the whole training set and projects high-dimensional feature vectors onto a two-dimensional image plane. The feature values of a specific input vector are then mapped to these locations producing one specific image for one specific vector.

Note that DeepInsight requires a fixed training data set. Sequential data acquisition schemes such as SuMO require recalculating the DeepInsight mapping and retraining the entire network at each iteration, increasing the computational effort.

5.5.3.2. Results

Figure 5.11 shows the results obtained with U-Net and CFP-Net. Our domain-driven encodings I_C and I_F almost consistently outperform the domain-independent encoding schemes (cf. Section 5.5.3.1), especially for small training sets. While increasing the training set size benefits all encodings, the multi-path architecture with vector-valued gripper encoding exhibits performance plateaus between 250 and 500 data points. Interestingly, there is practically no difference between gripper stiffness and gripper-force encoding. This might imply that the spatial distribution of the gripper influence is more important than the signal intensity. For the optimization-relevant sparse-data situations (100 samples), our domain encoding performs best on the CFPNet-architecture and reduces the MAE by $\approx 120\%$ and the RMSE_{obj} by $\approx 20\%$. We observe inferior performance of U-Net throughout our studies. Thus, we select CFPNet as the model of our choice and for discussion in the following.

5.5.4. Alignment of Training and Optimization Objective

The last kind of additional information we study is Objective Alignment (OA), cf. Equation 5.2. The objective function of an engineering problem contains complex and well-

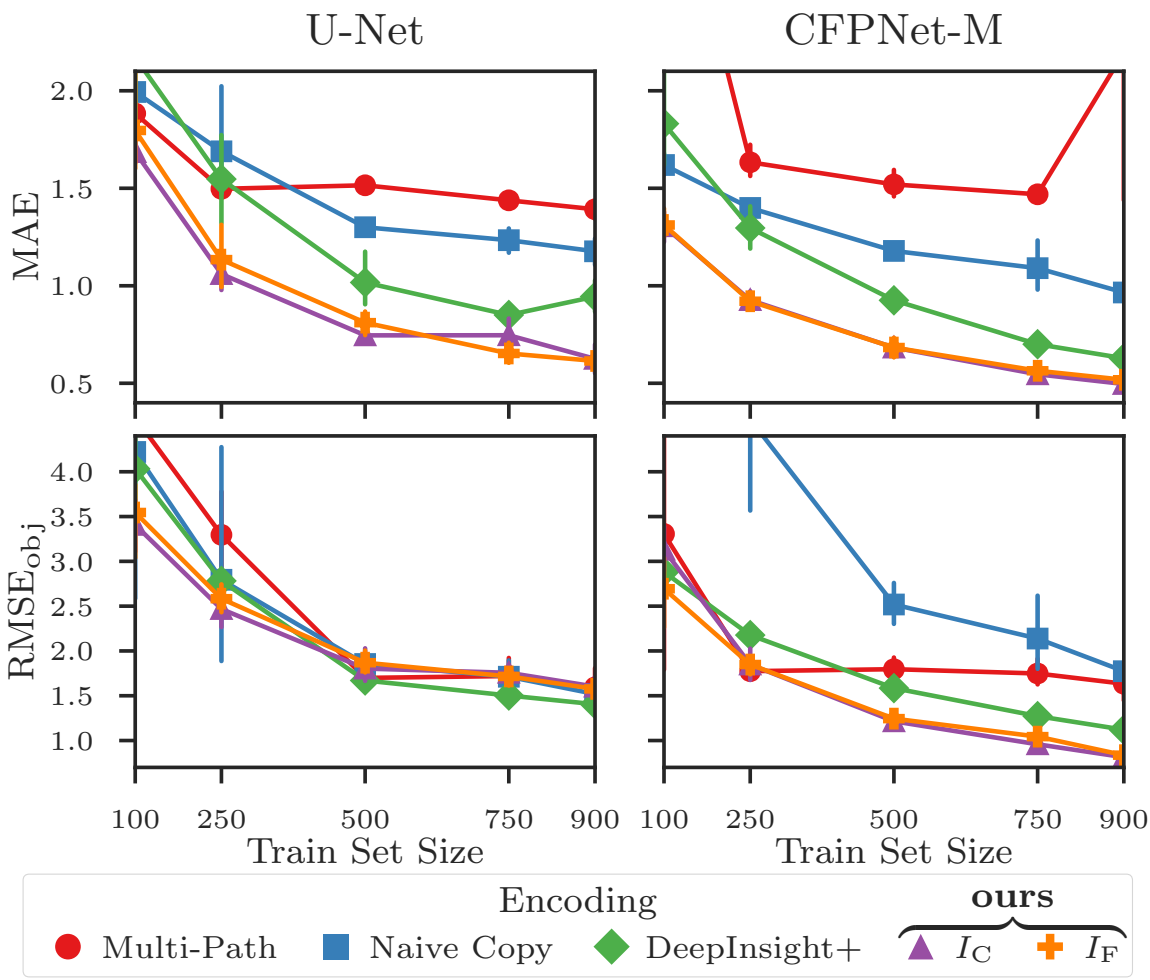


Figure 5.11.: Comparison of gripper-encoding approaches for CFP-Net regarding MAE and $RMSE_{obj}$.

formalized domain knowledge, which OA seeks to leverage. Although this knowledge is case-specific, it is always available in any automated parameter optimization context across disciplines. However, a key challenge remains: developing a method to effectively exploit this information in the surrogate model.

5.5.4.1. Investigated Loss Functions

To evaluate the suitability of OA, we compare the resulting surrogate accuracy obtained by OA to the accuracy obtained with other loss functions. Specifically, we compare our $\mathrm{MAE}_{\mathrm{OA}}$ (Equation 5.2) to the commonly used MAE, MSE, and also the *structural similarity* (SSIM) loss [Wan+04] and a *domain-regularized* loss (DRL) [Das+22].

The SSIM-loss considers changes in *structural information*, *luminance*, and *contrast* of an image. Here, *structural information* refers to spatially adjacent pixels, *luminance* to the reduced visibility of image distortion at the edges, and *contrast* to less apparent distortions within textured regions. Similar to CNNs, SSIM operates on a number M of image windows rather than individual pixels. SSIM reads per image n:

$$\mathrm{SSIM}_n = 1 - \frac{1}{M}\sum_{m=1}^{M}\frac{(2\mu_{\boldsymbol{\gamma}_{nm}}\mu_{\hat{\boldsymbol{\gamma}}_{nm}} + c_1)(2\sigma_{\boldsymbol{\gamma}_{nm},\hat{\boldsymbol{\gamma}}_{nm}} + c_2)}{(\mu^2_{\boldsymbol{\gamma}_{nm}} + \mu^2_{\hat{\boldsymbol{\gamma}}_{nm}} + c_1)(\sigma^2_{\boldsymbol{\gamma}_{nm}} + \sigma^2_{\hat{\boldsymbol{\gamma}}_{nm}} + c_2)} \tag{5.7}$$

where $\boldsymbol{\gamma}_{nm}, \boldsymbol{\gamma}_{\theta nm}$ are windows in the original $\boldsymbol{\gamma}_n$ and the predicted $\boldsymbol{\gamma}_{\theta n}$ image, and μ_* and σ_* are the mean and the variance of such a window, while $\sigma_{*,*}$ is the covariance between windows. The constants c_1, c_2 are added to avoid instability. We use the standard hyperparameters, except for the window size, which we set to 5; for more details on SSIM and its parameters, see [Wan+04].

The DRL adjusts the loss function with penalty terms to reflect domain constraints and is used to include domain knowledge in several works [KGC17; Kar+17b; Hoe+22]. It consists of two parts: a *label-based* part for the training data and a *knowledge-based* part for the domain knowledge. During model training, the optimizer tries to satisfy both parts simultaneously using a tradeoff hyperparameter α. We use the difference of weights $w_{np}, w_{\theta np}$ for pixel p in image n, based on the predicted shear strain $\gamma_{\theta np}$ and the ground truth shear strain γ_{np} as in Equation 5.1 as knowledge-based part:

$$\mathrm{MAE}_{\mathrm{DRL},n} = \underbrace{(1-\alpha)\sum_{p=1}^{P}\left|\gamma_{np} - \gamma_{\theta np}\right|}_{\text{Label-based}} + \underbrace{\alpha\sum_{p=1}^{P}\left|w_{np} - w_{\theta np}\right|}_{\text{Knowledge-based}}. \tag{5.8}$$

The intuition behind this formulation is to punish pixels that should be important for the objective but whose predictions are not, and vice versa. DRL can be combined with any standard loss function. We did this in our evaluation with MAE, MSE, and SSIM. To select the α hyperparameter, we conducted a grid search in the range $[0, 1]$ with a step size of 0.1 with a fixed training set size of 500. The optimal value for α was determined to be 0.5 and used for all subsequent numerical studies.

5.5.4.2. Results

Table 5.2 summarizes the average accuracy and the 95%-confidence interval for models trained with a given loss and evaluated on different evaluation metrics. The following observations were consistent across all sample sizes. If the loss function is similar to the evaluation metric, i.e., training and evaluation objectives are aligned, performance is best. This supports our hypothesis that objective alignment is important. To estimate product quality, we deem the $RMSE_{obj}$ metric most relevant. For $RMSE_{obj}$ we see that MAE_{OA} and MSE_{OA} outperforms every other loss function. It is noteworthy that for all evaluation metrics, some version of the MAE performs best. We conclude that MAE is the most suitable loss for general strain field prediction and MAE_{OA} for product quality prediction.

Table 5.2.: Evaluation metrics for different loss functions and train sizes. Bold entries mark the best result within one metric

		Evaluation Metrics		
Samples	**Loss**	**RMSE**	**MAE**	**$RMSE_{obj}$**
100	MAE	**1.37 ± 0.05**	**1.01 ± 0.03**	3.00±0.46
	MSE	1.74 ± 0.12	1.31 ± 0.10	3.13±0.33
	SSIM	1.74 ± 0.23	1.03 ± 0.03	15.78±7.55
	DRL MAE	1.94 ± 0.40	1.41 ± 0.29	3.43±0.67
	DRL MSE	1.91 ± 0.05	1.42 ± 0.04	3.06±0.27
	DRL SSIM	1.68 ± 0.11	1.24 ± 0.09	5.26±1.58
	OA MAE	1.63 ± 0.10	1.18 ± 0.08	**2.36 ± 0.20**
	OA MSE	1.77 ± 0.08	1.34 ± 0.06	2.44±0.25
250	MAE	**1.02 ± 0.04**	**0.74 ± 0.03**	1.66±0.18
	MSE	1.25 ± 0.09	0.93 ± 0.06	1.85±0.23
	SSIM	1.03 ± 0.08	0.75 ± 0.04	4.83±4.73
	DRL MAE	1.28 ± 0.22	0.93 ± 0.14	1.75±0.18
	DRL MSE	1.42 ± 0.10	1.05 ± 0.06	1.92±0.20
	DRL SSIM	1.26 ± 0.19	0.93 ± 0.13	2.55±0.23
	OA MAE	1.17 ± 0.07	0.85 ± 0.05	**1.27 ± 0.06**
	OA MSE	1.33 ± 0.06	1.00 ± 0.07	1.62±0.16
500	MAE	**0.75 ± 0.03**	**0.55 ± 0.02**	1.28±0.10
	MSE	0.91 ± 0.06	0.68 ± 0.05	1.21±0.09
	SSIM	0.79 ± 0.07	0.56 ± 0.02	5.36±5.74
	DRL MAE	0.83 ± 0.03	0.60 ± 0.03	1.23±0.10
	DRL MSE	0.97 ± 0.06	0.72 ± 0.04	1.39±0.07
	DRL SSIM	0.86 ± 0.07	0.64 ± 0.05	1.92±0.15
	OA MAE	0.87 ± 0.03	0.62 ± 0.03	**0.88 ± 0.11**
	OA MSE	1.02 ± 0.10	0.75 ± 0.08	1.10±0.09
900	MAE	**0.62 ± 0.02**	**0.45 ± 0.01**	0.87±0.16
	MSE	0.67 ± 0.04	0.50 ± 0.03	0.81±0.04
	SSIM	**0.62 ± 0.06**	0.46 ± 0.02	3.76±2.56
	DRL MAE	0.72 ± 0.05	0.52 ± 0.04	1.04±0.11
	DRL MSE	0.84 ± 0.06	0.63 ± 0.05	1.09±0.13
	DRL SSIM	0.64 ± 0.06	0.47 ± 0.05	1.60±0.11
	OA MAE	0.71 ± 0.02	0.51 ± 0.02	**0.65 ± 0.05**
	OA MSE	0.81 ± 0.05	0.60 ± 0.04	0.73±0.03

5.6. Evaluating SuMO with Domain Knowledge

After evaluating the surrogate performance for each knowledge type separately, we assess their suitability for SuMO with all knowledge included. We evaluate four distinct optimization strategies, a non-surrogate approach, and three different SuMO strategies: I) This approach is a classical Evolutionary Algorithm (EA) without surrogate as in [Pfr+18]. II) The second approach (MLP) aligns with [Pfr+18; Zim+21]. It uses the MLP model from [Zim+21] as a surrogate and employs an EA to minimize the objective function while iteratively refining the surrogate. III) The third approach (MLP-MC) extends the MLP model by integrating *MC-Dropout*[1] for uncertainty estimation and Bayesian Optimization. No domain knowledge is involved so far. IV) Finally, our approach (CFPNet-M-MC) substitutes the MLP-MC surrogate with the CFPNet-M and includes all domain knowledge. CFPNet-M-MC also uses MC-Dropout to enable Bayesian Optimization.

For the Bayesian Optimization, we use Monte Carlo Dropout to estimate the point-wise prediction variance using it to calculate the Expected Improvement (EI). To prevent getting stuck in local optima, we use a combination of random and local search to maximize EI. EI is first calculated for 5000 random candidates, then, for the 10 best candidates, a local search is performed. The optimization was implemented using the Sequential Model-Based Optimization (SMAC3) framework [Lin+22], which was used for the winning solution to the BBO challenge [Awa+20].

5.6.1. Numerical Study Setup

Since the optimization approaches are stochastic, performance evaluations on a single optimization run are not meaningful. Thus, we reevaluate each optimization strategy three times. To validate our domain-informed SuMO approach, we need to run the entire SuMO procedure assuming that we have no initial data. This requires a ground truth simulator. Since a single FE forming simulation takes up to 2 hours, repeated optimization runs with hundreds of simulations are not feasible. Hence, we replace the FE simulator with a more efficient oracle model based on our best-performing model: CFPNet, with all domain knowledge, trained with MAE loss on the entirety of the available dataset of 900 simulations. In our study, we treat data from the oracle as ground truth data. Since we only have to train this model once, we use a model with higher learning capacity, i.e., it has 128 convolutional channels [LGL23].

We initiate all SuMO strategies with a design of 100 simulations generated via a Sobol sequence. The EA-method, i.e. strategy I), starts without simulations. We restrict each method's access to the oracle to a total of 1000 simulations, which includes the initial 100 simulations.

[1] Monte Carlo Dropout randomly switches off a certain number of neurons during model training and evaluation to calculate estimation uncertainty.

5.6.2. Results

Table 5.3.: Final optimization results for different metrics. Bold values mark the best results.

Surrogate	Start Results Quality $\bar{o}(\boldsymbol{\gamma}^*_{\theta s})$	Start Results $\gamma_{\max}$ in °	End Results Quality $\bar{o}(\boldsymbol{\gamma}^*_{\theta e})$	End Results $\gamma_{\max}$ in °	AUC
EA	385.24	47.29	380.18	43.73	342473
MLP	382.61	43.92	379.29	43.70	341716
MLP - MC	382.61	43.92	377.70	42.72	341251
CFPNet-M-MC	382.61	43.92	**373.77**	**41.03**	**339381**

Table 5.3 compares the start and end results of the optimization. Our CFPNet-M-MC method outperforms every other method in every metric. Compared to the start shear angle of EA, CFPNet-M-MC improves on the maximum shear angle $\gamma_{\max}$ by 6.3° or 13.24 %, respectively. We further evaluate the area under the curve (AUC, see 5.5.1.1). Besides the optimization result, it factors in how fast the optimization converges. Again, our method outperforms every other method, indicating that it performs better almost everywhere during the optimization.

Figure 5.12 shows optimization convergence in terms of *Relative Improvement* (RI) in iteration i according to:

$$\mathrm{RI}_i = (\bar{o}(\boldsymbol{\gamma}^*_{\theta i}) - \bar{o}(\boldsymbol{\gamma}^*_{\theta s})_{\mathrm{EA}})/\bar{o}(\boldsymbol{\gamma}^*_{\theta e})_{\mathrm{EA}} - \bar{o}(\boldsymbol{\gamma}^*_{\theta s})_{\mathrm{EA}}),$$

where $\bar{o}(\boldsymbol{\gamma}^*_{\theta s})_{\mathrm{EA}}$ and $\bar{o}(\boldsymbol{\gamma}^*_{\theta e})_{\mathrm{EA}}$ is the best part quality of the EA optimization found in start s and end i iteration, respectively, averaged for three runs. Likewise, $\bar{o}(\boldsymbol{\gamma}^*_{\theta i})$ is the part quality of the corresponding optimization strategy in iteration i. In loose terms, RI quantifies how much total optimization potential a strategy has exhausted in iteration i relative to the total EA-performance.

The graphs in Figure 5.12 confirm the AUC result: In every iteration, our approach finds a process configuration that, on average, outperforms any of its competitors. With an average improvement in end quality of 400% compared to the EA-baseline, and 200% compared to the second best competitor MLP-MC. While the results of our method scatter more than the baselines, they scatter in the direction of even better results. Even including the lower 95% confidence interval, our method's results are consistently better than the mean performance of any of its competitors. We assume the increased scatter stems from network initialization effects which may be larger for CNNs than for small MLPs.

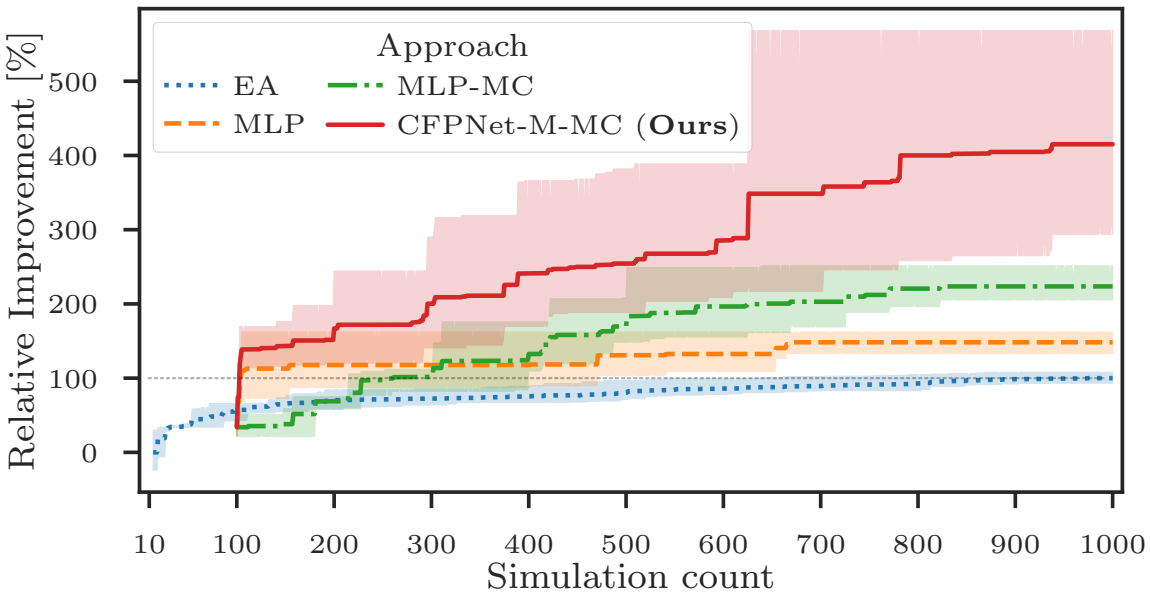


Figure 5.12.: Improvement during optimization with 95% confidence interval.

5.7. Chapter Conclusion

In this chapter we investigated the benefits of including additional domain knowledge into a surrogate model. We showed how this facilitates the data efficient solution of manufacturing process optimization. We investigated different methods of including domain knowledge and proposed Objective Alignment (OA), a new method that is generally useful across domains for including complex domain knowledge in surrogate models for SuMO. We investigated the different knowledge inclusion methods in isolation, showing that each one significantly improves the surrogate model. Finally, we have shown that the combination of all domain knowledge inclusion methods for SuMO outperforms every state-of-the-art model by a significant margin, with an average improvement of 200% even against the best-performing baseline.

Within the broader context of Adaptive Knowledge Discovery (AKD), these results demonstrate that the efficiency of the interaction loop – that is the effectiveness of a sampling strategy – is strongly dependent on how well the model can generalize and estimate the uncertainty. By anchoring the model to the underlying physical constraints of the DGP, and guiding its predictions to align with our knowledge discovery task, we can effectively increase the information gained from each expensive DGP measurement (FEM-simulation).

In particular, domain-independent methods such as OA are directly applicable and promise to improve the surrogate across domains. That is, one should strive to align the underlying models and sampling strategies with the discovery task if applying AKD to new tasks. We envision that OA opens multiple avenues for further research in the applied setting. In addition, future research on domain-specific types of knowledge, such as the geometry-process-relations, will lead to model improvements. However, such domain-

specific knowledge may require adaptations of the incorporation methods and different network structures for representation, e.g., graph neural networks.

In the next chapter (Chapter 6) we will specifically focus on developing a task aligned sampling strategy for a (in the context of AKD) new knowledge discovery task.

Part VI.

The Adaptive Knowledge Discovery Task of Multi-sample Testing

6. AMT: Data-Efficient Adaptive Multi-sample Testing for Binomial Data

The content of this chapter bases on:

- Béla H. Böhnke and Nadja Klein. 'AMT: Adaptive Multi-Sample Testing for Data Efficiency on Binomial Data'. In: *Data Science: Foundations and Applications*. Singapore: Springer Nature, under review

Keywords: Hypothesis Testing; Multi-sample Testing; Adaptive Sampling
The publication is still under review.

6.1. Chapter Overview

In the previous chapters we investigated each of the different building blocks of Adaptive Knowledge Discovery (AKD): The data-generating process, uncertainty quantification, and data-efficient learning by incorporating domain knowledge. We evaluated these concepts within the established frameworks of stream-based active learning, gaussian process regression, and surrogate model-based optimization.

In this chapter, we address a knowledge discovery task that has received little attention within the AKD context: The task of Multi-sample Testing (MT). To provide an adaptive solution to this task, we leverage our understanding of the data-generating process to perform Bayesian uncertainty quantification. Based on this foundation, we develop a new task-aligned sampling strategy designed to solve the MT problem with high data efficiency.

In many scientific fields, such as biology [e.g., microarray studies; NG94], medicine [e.g., clinical trials; LDA05; Jia+15], and materials science [e.g., superconductivity classification; Tal19], a common task is that of MT, i.e., to determine whether multiple samples taken under different conditions originate from a shared distribution. However, a major challenge is that data collection is often costly and time-consuming, making it impractical to acquire the large sample sizes typically required for classical statistical methods. In materials science, for example, expensive materials are frequently lost during destructive measurements. In medicine and psychology, for example, conducting studies on humans is time-consuming and expensive, and experiments may come with ethical concerns. In addition, non-adaptive classical sampling methods, such as random or space-filling sampling, collect all data upfront. This often results in valuable resources being wasted on regions of the data space that offer little information to distinguish between distributions, in turn reducing test power.

A promising direction for improving data efficiency is adaptive sampling. However, its application to MT remains largely unexplored, which can be attributed to three primary

challenges:
(P1) *Transferring adaptive sampling strategies to MT is not straightforward, because MT requires the non-trivial decision of which sample to add additional (new) data to, which differs from existing adaptive sampling tasks.*
(P2) *Because adaptive sampling introduces statistical dependence between observations, it violates the Independent and Identically Distributed (i.i.d.) assumption required for many classical test statistics.*
(P3) *Test statistic agnostic permutation tests cannot be employed in their existing form since they also require i.i.d. data, or involve computationally demanding re-simulation of the entire adaptive sampling loop – as exemplified in Appendix 6.4.4.3.*

Our work addresses these challenges in the context of Bernoulli-distributed data, which is common in many MT applications. For example, in biology, one might use this to efficiently test whether multiple protein samples exhibit the same distribution of known protein sequences. Building upon Bernoulli-distributed data, our contributions to overcome the aforementioned three primary challenges are:

(C1) We propose the Adaptive Multi-sample Testing (AMT) framework, schematically depicted in Figure 6.1, which consists of the following phases:
(0) AMT begins with $|\mathcal{X}|$ initial samples $\mathbf{Y}_{x0}$ containing N_0 observations each.
(1) AMT checks a stopping criterion, e.g., a given budget of iterations I.
(2) If the criterion is not met in iteration i, AMT estimates the difference between the outcome variables Y_x based on the samples.
(3) AMT then selects the two variables Y_{x_1}, Y_{x_2} with index x_1, x_2 with the highest estimated distributional difference,
(4) observes the outcomes y_{1i}, y_{2i},
and (5) adds them to the corresponding samples $\mathbf{Y}_{x_1(i-1)}, \mathbf{Y}_{x_2(i-1)}$ to form the expanded samples $\mathbf{Y}_{x_1 i}, \mathbf{Y}_{x_2 i}$ respectively.
(6) Once the iteration limit is met, AMT computes an MT using all $\mathbf{Y}_{xi} \mid x \in \mathcal{X}$ samples.

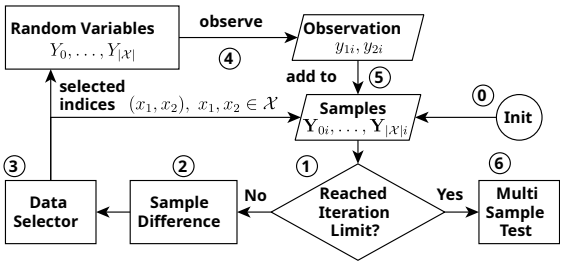


Figure 6.1.: Flow chart of our Adaptive Multi-sample Testing (AMT) approach.

(C2) We derive an adaptive sampling strategy that does not require hyperparameter tuning. It is based on a Bayesian upper confidence bound (BUCB), that is, the posterior probability that a sample differs from the other samples. By selecting data that increases this probability, we can significantly reduce the amount of data required to achieve similar test power.

(C3) We derive the null distribution of each sample, conditioned on the sample size, to correct for the non-i.i.d. samples generated by our sampling strategy. This enables us to obtain a test that maintains theoretical guarantees for the type I error, while being

computationally feasible. We use this for sample-wise subtests, combining the subtest results to form a Bonferroni-corrected omnibus test.

We benchmark AMT in extensive numerical experiments against a range of adaptive and non-adaptive sampling strategies and test statistics. To demonstrate the flexibility and robustness of AMT, we also evaluate permutation-based variants of AMT. Our code[1] is available online to facilitate reproduction.

While AMT is tailored to Bernoulli-distributed data, it is also conceptually applicable to settings with observations from metric variables. This can be achieved by transforming the data into discrete 'pseudo'-samples of binary observations, similarly to Mood's-Median test [Moo63]. We tested this transformation on real-world data and provide details in Section 6.4.4.1.

6.1.1. Fundamentals

To differentiate MT from other related task, like Best Arm Identification (BAI) or Multi-sample Detection (MD), we will start by defining the task. In this chapter, to maintain consistency with the overarching AKD framework, we utilize our unified notation.

In our Data-Generating Process (DGP) formulation $Y_X = C(X)$, we select (through an intervention $do(X = x)$) a sample index $x \in \mathcal{X} \subset \mathbb{N}$ and observe the outcome y_x from random outcome variable Y_X. This is in contrast to the standard in testing and bandit literature where the random outcome variable is typically named X_n (or X_k) and n (or k) is used as index.

Multi-sample Testing (MT), as the name suggests, requires multiple given samples $\mathbf{Y}_x$ of size N_x. MT assumes that a sample is a set of observations $\mathbf{Y}_x = \{y_{x1}, y_{x2}, \ldots, y_{xN_x}\}$ from a random variable Y_x. Obtaining a single observation y_x from the random variable Y_x is denoted as $y_x \leftarrow Y_x$, while obtaining a whole sample $\mathbf{Y}_x$ of size N_x is denoted as $\mathbf{Y}_x \underset{N_x}{\leftarrow} Y_x$. MT generally has the goal of determining whether all samples (and thus all random variables) originate from a shared distribution $Y_1, \ldots, Y_{|\mathcal{X}|} \sim P(Y|\theta)$ with parameter $\theta \in \Theta$ or not. In practice, for example in experimental science, θ often refers to the mean, and one is interested in testing whether the sample means differ. More formally, this leads to the following definition:

Definition 6.1 (Multi-sample Testing for the Difference in Means)**.** *Let $\mu_1, \mu_2, \ldots, \mu_{|\mathcal{X}|}$ denote the true means of the $|\mathcal{X}|$ random variables $Y_x \sim P(Y|\mu_x)$, $x \in \mathcal{X}$. The goal is to test:*

$$H_0 : \ \mu_1 = \mu_2 = \cdots = \mu_{|\mathcal{X}|} \quad \textit{vs.}$$
$$H_1 : \ \exists\, x_1 \neq x_2,\ x_1, x_2 \in \mathcal{X},\ \textit{such that}\ \mu_{x_1} \neq \mu_{x_2}\ .$$

MT does not aim to identify which specific samples differ (which would be a localization and not a detection task), but rather to detect whether any deviation exists. Post-hoc analysis may be used to localize differences once H_0 is rejected. Section 6.1.1.2 gives some practical guidelines to perform MT.

In this framework, two types of statistical errors can occur during the decision-making process which are used as standard metrics used to evaluate the performance of a test.

[1] Git: https://github.com/bela127/AMT_Adaptive-Multi-Sample-Testing

Definition 6.2 (Type I Error (Alpha Error))**.** *The type I error, or alpha error (α), is the probability of incorrectly rejecting the null hypothesis H_0 when H_0 is, in fact, true. In the context of MT, this means concluding that a difference exists between the population means when they are all equal. The significance level of a test, is also often denoted by α. It is the maximum acceptable probability of making a type I error.*

$$\alpha = P(\textit{Reject } H_0 \mid H_0 \textit{ is true})$$

Definition 6.3 (Type II Error (Beta Error))**.** *The type II error, or beta error (β), is the probability of failing to reject the null hypothesis H_0 when the alternative hypothesis H_1 is true. It represents the case where a difference exists but the test fails to detect it. It is related to power by Power $= 1 - \beta$.*

$$\beta = P(\textit{Fail to reject } H_0 \mid H_1 \textit{ is true})$$

Definition 6.4 (Power of Multi-sample Testing)**.** *The power of a Multi-sample Testing is the probability of correctly rejecting the null hypothesis H_0 when the alternative hypothesis H_1 is true. For a given test and a specific alternative, the power is defined as:*

$$\textit{power} = P(\textit{Reject } H_0 \mid H_1 \textit{ is true} = 1 - \beta)$$

It is a measure of a test's ability to detect an actual difference among the populations.

The relationship between these errors and the true underlying scenario is summarized in Table 6.1.

Table 6.1.: Decision matrix for Multi-sample Testing.

Test Result	**Actual Scenario**	
	H_0 is True	H_1 is True
Accept H_0	Correct Decision ($1 - \alpha$)	Type II Error (β) (False Negative)
Reject H_0	Type I Error (α) (False Positive)	Correct Decision (Power $1 - \beta$)

6.1.1.1. Differentiation to Other Tasks

While similar to other tasks MT has some important distinctions to other tasks.

Best Arm Identification (BAI) in the multi-armed bandit setting operates under a different set of assumptions regarding the underlying state. In the bandit setting, a type I error is defined as the probability that the selected arm x_{sel} is a non-optimal arm x_{bad}. This formulation makes the implicit assumption that there is an optimal arm x_{opt} to be selected [Hil+13].

In the context of Hypothesis Testing, BAI is therefore situated permanently in the H_1 setting, assuming from the outset that at least one arm differs from the rest. Formally, the bandit type I error is:

$$\text{TypeI}_{\text{bandit}} = P(x_{\text{sel}} = x_{\text{bad}} \mid H_1) \tag{6.1}$$

It is important to note that assuming H_1 will not give us the type I (α) error of classical Hypothesis Testing; instead, it yields the type II (β) error and its inverse, the power of the bandit. BAI focuses on identifying the specific source of variation rather than establishing whether variation exists at all.

Indeed, if the null hypothesis H_0 were to hold and all arms were identical, any selected arm x_{sel} would be considered optimal by definition, making it impossible to commit a type I (α) error in the Hypothesis Testing sense. Consequently, the BAI framework lacks the mechanism to control for the classical type I (α) error, as it does not account for the possibility that no difference exists.

Multi-sample Detection (MD) assumes the same H_0 as MT, aiming to determine if any sample deviates from a baseline. However, while MT strictly enforces a guarantee for the type I (α) error to bound the probability of false positives, MD often prioritizes detection sensitivity without providing a formal statistical guarantee for α.

Independence Testing (IT) determines if variables are statistically independent by testing the null hypothesis $H_0 : P(X_1, X_2) = P(X_1)P(X_2)$. Unlike MT, which compares marginal distributions across $|\mathcal{X}|$ distinct groups to see if they originate from the same population, Independence Testing (IT) evaluates the relationship between variables within a single population X without conditioning on any groups. That is, MT treats group membership as a discrete independent variable to identify distributional differences. IT treats exactly two (often continuous) distinct variables symmetrically, focusing on the existence of a statistical relationship rather than assigning specific independent or dependent roles, focusing on their association.

6.1.1.2. A Short Guide to Multi-sample Testing

In past years experimental scientists repeatedly discussed how to correctly perform hypothesis tests. Often finding that they are used incorrectly, not fulfilling the requirements of Neyman and Pearson [NP33b], resulting in an increase of false positive results (type I errors) [Gig93; Gig04]. Thus the *American Psychological Association (APA)* assembled the *Task Force on Statistical Inference (TFSI)* to establish a guideline for performing hypothesis test [Wil99].

According to this guideline performing a hypothesis test should typically involve four steps [EGS26; Gig04]: (1) Defining the H_0 and H_1 hypothesis. In this step, it is essential to define an explicit H_1, which involves assuming an effect size preferably based on theory or using a standard value calculated by convention (e.g., low, medium, large [Coh88]). Explicitly defining the H_1 is important, because otherwise in the case of a non-reject, one would not be able to interpret the H_0 [NP33b; NP33a], i.e., one does not know if the sample size was too small or if there was no effect. (2) This brings us directly to the next step – power analysis and sample size planning. To be able to interpret both H_0 and H_1, one needs to decide on an appropriate significance (e.g. $\alpha = 5\%$) and also needs to select an appropriate power, typically over 80%. With α, H_1, and power fixed, one can now perform a power analysis and obtain a total sample size $N = \sum_{x \in \mathcal{X}} N_x$ that is at least required to reach the required power. (3) Only now can one start a meaningful data collection, and

collect as much data as calculated by the power analysis. (4) Finally, one can calculate the multi-sample test statistic and interpret its results.

6.2. Related Work

In this section, we review related work on MT, adaptivity in hypothesis testing, and discuss data-efficient decision-making techniques based on adaptive sampling through feedback loops from other disciplines.

6.2.1. Multi-sample Testing

Classical Approaches include ANOVA [Fis92], the Kruskal–Wallis Test (KW) [a non-parametric alternative based on ranks; KW52], the multi-sample chi-square test [Pea92], and the median test [based on a chi-square test after data transformation; Moo63]. All these MT assume iid data meaning they require a non-adaptive sampling process.
Kernel-based and distribution-free tests are applicable to higher dimensions, usually also requiring iid data. Examples are tests based on the maximum mean discrepancy [Gre+12] or its known equivalent, the energy distance [SR13].
Permutation tests are a general class of non-parametric tests, which generate a null distribution for virtually any chosen test statistic by randomly permuting the observations [Ern04]. This idea relies on the exchangeability of observations under H_0, which is explicitly broken by adaptive sampling. Therefore, attaining a correct distribution under H_0 requires a non-standard permutation involving computationally intensive re-simulation of the adaptive sampling loop.

6.2.2. Adaptivity in Hypothesis Testing

Adaptive methods for hypothesis testing have their origins in sequential experimental design [Rob52]. Unlike classical approaches that use a fixed sample size, sequential designs adapt to the data as it is collected to improve efficiency. More recent adaptive strategies, such as α-investing [FS08; JM15], have focused on adaptively allocating an error budget across a series of sequential hypothesis tests. However, these methods are designed for multiple individual tests rather than MT, which involves a single global hypothesis test. Similar recent work on e-values [RW25], while useful for combining multiple individual tests, is not focused on sequential testing.

Adaptive sampling has shown promise in various areas, but adapting it to MT has proven challenging as it disrupts classical type I error control. Thus, most works focus on Multi-sample Detection (MD), i.e., essentially MT but without type I error control. [HCN11] is such a work that proposes an iterative sampling strategy to detect differences between Gaussian variables, efficiently reducing the required sample size for detection. The missing type I error control makes it unsuitable for hypothesis testing.

6.2.3. Adaptive Sampling Techniques

Adaptive sampling aims to obtain (e.g., measure, label) data useful for data-efficiently solving a knowledge discovery task (e.g., model learning, optimum search). It has been used in the following contexts:
Multi-Armed Bandit Algorithms sequentially select among multiple options (arms) to maximize a cumulative reward [LS20]. Some works address best-arm identification [ABM10; KCG16], which is related to localizing a variable with stochastically different behavior. However, as discussed in Section 6.1.1.1 they can not provide type I error control in the hypothesis testing sense.
Surrogate Model-based Optimization aims to optimize expensive black-box functions with minimal data, rather than to perform statistical testing [Gor+10b]. Bayesian optimization is a prominent example [Spr+16; Wan+16] that adaptively explores the input space.
Active Learning seeks to train accurate predictive models with minimal labeled data by selecting the most informative samples [Set12]. Although it shares the principle of adaptive data selection, its core focus is on prediction tasks, not hypothesis testing.

In contrast to these tasks, we propose AMT, which utilizes adaptive sampling for MT, aiming to obtain data that aids the test decision while minimizing the sample size required for acceptable power. This requires a new and aligned understanding of the task-dependent concept of 'data usefulness'.

6.2.4. Summary and Contributions

To the best of our knowledge, no existing methods perform MT via adaptive data collection. The closest works using adaptive sampling [HCN11; KCG16] focus on a form of Multi-sample Detection and do not provide type I error control.

In contrast, our work presents a novel adaptive method tailored to MT and a new test statistic with an analytically derived null distribution. Thereby, we directly address the lack of strict type I error control in an adaptive setting, with increased test power, and offer a robust solution for MT in data-limited scenarios.

6.3. Problem Statement

In this section, we introduce some general notation, and MT for Bernoulli-distributed variables.

Bernoulli Multi-sample Testing (BMT) adapts MT to the case of Bernoulli-distributed random variables $Y_x \sim \text{Bernoulli}(p_x)$, with success probability p_x. A sample $\mathbf{Y}_x \underset{N_x}{\leftarrow} Y_x$ is a set $\mathbf{Y}_x = \{y_{x1}, y_{x2}, \ldots, y_{xN_x}\}$ of binary observations $y_{xn} \in \{0, 1\}$, $n \in \{1, \ldots, N_x\}$.

Definition 6.5 (Bernoulli Multi-sample Testing (BMT)). *Let $p_1, \ldots, p_{|\mathcal{X}|}$ be the true success probabilities of $|\mathcal{X}|$ independent Bernoulli-distributed random variables Y_x with $x \in \mathcal{X}$. The goal is to test*

$$H_0 : \; p_1 = p_2 = \cdots = p_{|\mathcal{X}|} = p \quad \textit{vs.}$$
$$H_1 : \; \exists\, x \in \mathcal{X}, \; \textit{such that } p_x \neq p,$$
$$\textit{where } p = \frac{1}{|\mathcal{X}|} \sum_{k \in \mathcal{X}} p_k \textit{ is the mean of } p_1, \ldots, p_{|\mathcal{X}|}.$$

To ground the abstract formulation of Bernoulli Multi-sample Testing (BMT) in an intuitive mental model, we interpret this problem as **Fake Coin Detection (FCD)**. Here, each random variable Y_x is viewed as a physical coin that can be tossed to observe a head $y_x = 1$ or tail $y_x = 0$. Under this interpretation, BMT tests whether all coins in a collection are identical or if at least one coin is "plated" (deviant). By adopting this terminology, we aim to replace abstract binary observations with the more accessible concept of coin flips, allowing the reader to visualize the sampling process as an agent deciding which coin to flip next to most efficiently identify the 'fake'.

The research question then is:
Which coins should we flip next, given the previous samples from all coins, to detect a fake coin with as little data as possible while still guaranteeing a required type I error?

6.4. Our Method: Adaptive Multi-sample Testing (AMT)

This section will first provide a high level overview over our AMT Algorithm, in Section 6.4.1, discussing all the building blocks of the resulting Algorithm 6.1, including sample initialization, adaptive coin selection, the coin difference metric, and our test statistic. We will then give implementation details for our new coin selection in Section 6.4.2. And discuss our new test statistic, which corrects for adaptive sampling in Section 6.4.3. We close with a discussion on the properties and the application of our AMT Algorithm in Section 6.4.4.

6.4.1. High Level Overview

Let $|\mathcal{X}|$ be the number of Bernoulli coins Y_x with $x \in \mathcal{X}$, I be the maximum number of sampling iterations i, and $N_0 = N_{x0} \mid x \in \mathcal{X}$ be the number of initial observations per sample $\mathbf{Y}_{x0}$ such that $|\mathbf{Y}_{x0}| = N_{x0} = N_0$ after initialization. For a sample $\mathbf{Y}_{xi}$ of coin Y_x at iteration i we denote by $h_{xi} = \sum_{y_n \in \mathbf{Y}_{xi}} y_n$ the number of heads and by $t_{xi} = N_{xi} - h_{xi}$ the number of tails within sample $\mathbf{Y}_{xi}$.

The steps of AMT in Algorithm 6.1 are as follows:

(1) Initialization: Each sample $\mathbf{Y}_{x0}$ is initialized with N_0 observations y_{xn} from coin Y_x: $\mathbf{Y}_{x0} \underset{N_0}{\leftarrow} Y_x \equiv \mathbf{Y}_{x0} = \{y_{xn} \leftarrow Y_x | n \in \{1, \ldots, N_0\}\}$ (Lines 2 to 3).

Algorithm 6.1 AMT: Adaptive Multi-sample Testing

1: **procedure** AMT(Inputs: $\mathcal{X}$, I, N_0)
2: **for** $x \in \mathcal{X}$ **do**
3: $\mathbf{Y}_{x0} \underset{N_0}{\leftarrow} Y_x$ ▷ *Sample Initialization (Step 1)*
4: **for** $i \in \{1, \dots, I\}$ **do** ▷ *Stopping Criterion (Step 2)*
5: $(x_1, x_2) := \arg\max_{x_1, x_2} d(\mathbf{Y}_{x_1(i-1)}, \mathbf{Y}_{x_2(i-1)})$ ▷ *Coin Selection (Step 3)*
6: $y_{1i} \leftarrow Y_{x_1}$ ▷ *Coin Sampling (Step 4.1)*
7: $y_{2i} \leftarrow Y_{x_2}$
8: $\mathbf{Y}_{x_1 i} := \mathbf{Y}_{x_1(i-1)} \cup \{y_{1i}\}$ ▷ *Sample Update (Step 4.2)*
9: $\mathbf{Y}_{x_2 i} := \mathbf{Y}_{x_2(i-1)} \cup \{y_{2i}\}$
10: MSTest($\{\mathbf{Y}_{xI} \mid x \in \mathcal{X}\}$) ▷ *Multi-sample Test (Step 5)*

(2) Stopping Criterion: At the beginning of each iteration (Line 4) AMT checks if the given iteration limit I has been reached, and will continue sampling if not. Here, I is a proxy for the maximum combined sample size $N = \sum_{x \in \mathcal{X}} N_{xI} = N_0 \cdot |\mathcal{X}| + 2I$. An appropriate stopping iteration I depends on the problem and needs to be determined prior through a power analysis (see Section 6.1.1.2).

(3) Coin Selection: A crucial aspect of AMT is the adaptive selection of coins. In Line 5, AMT selects two coins Y_{x_1} and Y_{x_2} that maximize a predefined *coin difference* $d(\mathbf{Y}_{x_1(i-1)}, \mathbf{Y}_{x_2(i-1)})$ between their current respective samples $\mathbf{Y}_{x_1(i-1)}$ and $\mathbf{Y}_{x_2(i-1)}$. This selection is formally expressed as:

$$(x_1, x_2) := \arg\max_{x_1, x_2} d(\mathbf{Y}_{x_1(i-1)}, \mathbf{Y}_{x_2(i-1)}) \tag{6.2}$$

We construct $d(\cdot, \cdot)$ such that it estimates the probability of two coins having different success probabilities, making it a variant of Bayesian confidence bounds; see Section 6.4.2 for details. Sampling according to these bounds automatically balances exploration and exploitation, eliminating the need for manual tuning. We note that our algorithm is flexible in the sense that it can utilize different choices for the difference $d(\cdot, \cdot)$, resulting in different variants of AMT. We explore such variants in our experiments in Section 6.5.

(4) Coin Sampling: Subsequently, AMT observes the two selected coins Y_{x_1} and Y_{x_2} (Lines 6 to 9), adding the outcomes y_{1i} and y_{2i} to their respective sample sets $\mathbf{Y}_{x_1 i} := \mathbf{Y}_{x_1(i-1)} \cup \{y_{1i}\}$ and $\mathbf{Y}_{x_2 i} := \mathbf{Y}_{x_2(i-1)} \cup \{y_{2i}\}$.

(5) Multi-sample Testing: Upon reaching the iteration limit, a multi-sample test is performed on the collected samples (Line 10). Although AMT can use various existing test statistics (as demonstrated in Section 6.5.1), adaptive sampling results in non-iid data, which invalidates the null distribution of classical tests. To correct for this, we propose a novel test statistic, based on the Beta-Binomial distribution, that maintains comparable power while allowing for the analytical derivation of the null distribution and, thereby, the critical values for controlled type I error. We give the details in Section 6.4.3. Compared to

permutation testing, our proposal is much faster and has less computational complexity, as exemplified in Section 6.4.4.3.

Remark 6.1 (Relevance and Constraints)**.** *AMT shares the same goals as MT, with the added objective of arriving at a meaningful test decision with as little data as possible. However, by doing so, AMT is constrained to applications where additional data can be collected on request (in contrast to only having observational data). This constraint is often fulfilled in experimental science, for which AMT is also most relevant, because here data collection is often expensive.*

6.4.2. Coin Selection

In Step *(3)* of our algorithm we select coins according to their difference. Because the true difference between coins is unknown, we quantify the probability of a difference by using a BUCB based on appropriate posterior distributions under H_0 and H_1:

Per-sample posterior under $\mathbf{H_1}$: We consider the posterior distributions $B_x \equiv P(p_x \mid \mathbf{Y}_x)$, $x \in \mathcal{X}$. We assign Beta priors $\text{Beta}(a, b)$ to each success probability p_x, setting $a = b = 1$ to reflect uniform priors on $(0, 1)$. The resulting posterior distributions are again Beta distributions $B_x = \text{Beta}(a_x, b_x)$ with parameters $a_x = 1 + h_x$, $b_x = 1 + t_x$, where h_x, t_x are the number of heads and tails in sample $\mathbf{Y}_x$, respectively.

Posterior under $\mathbf{H_0}$: We consider the posterior distribution $B_{\text{med}} \equiv P(p \mid \mathbf{Y}_{\text{med}})$ as a reference distribution. Here p is the average success probability across all samples as follows naturally from H_0 in Definition 6.5, and $\mathbf{Y}_{\text{med}}$ reflects an 'average sample'. Assuming the same (uniform) Beta priors $\text{Beta}(a, b)$ for p as previously for p_x, we arrive at the Beta posterior $B_{\text{med}} = \text{Beta}(a_{\text{med}}, b_{\text{med}})$ with $a_{\text{med}} = 1 + h_{\text{med}}$ and $b_{\text{med}} = 1 + t_{\text{med}}$. Here h_{med} and t_{med} are the number of heads and tales in the 'average sample'.

Because we do not have an 'average sample' we need to estimate its parameters h_{med} and t_{med} from the individual samples $\mathbf{Y}_x$. We do this by first estimating the average success probability p as the median $p_{\text{med}} = \text{med}(\bar{p}_1, \ldots, \bar{p}_{|\mathcal{X}|})$ of the estimated per-sample success probability $\bar{p}_x = \frac{h_{xi}}{N_x}$, $h_{xi} = \sum_{n \in \{1,\ldots,N_x\}} y_{xn}$. Here, we use the median sample because it is a more robust estimator under H_0 in the case of outliers. We then calculate the average sample size $N_{\text{med}} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} N_x$, and finally obtain our posterior parameters h_{med} and t_{med} as $h_{\text{med}} = p_{\text{med}} \cdot N_{\text{med}}$ and $t_{\text{med}} = N_{\text{med}} - h_{\text{med}}$, respectively.

Calculating a credibility region: Before we can calculate a BUCB we require a *credibility region* under H_0. Such a credibility region, in contrast to a hypothesis testing *confidence interval*, has the interpretation that: 'It contains the true parameter value p with a probability of $1 - \alpha_c$ under H_0'. A hypothesis testing *confidence interval*, on the other hand, is interpreted as: 'Under H_0 it is overlapping the true parameter value p in $1 - \alpha$ times of the cases'. To calculate a credibility region one needs to decide on a coverage probability α_c, not to be confused with the type I error α of a hypothesis test. We choose a standard value of $\alpha_c = 95\%$. The choice of α_c is known to be robust [KCG12], which we verified in a preliminary study. Thus for our final approach α_c needs no tuning and can be treated as a

constant. We can now calculate a two-sided $1 - \alpha_c$ credibility region $[c_\uparrow, c_\downarrow]$ using our H_0 posterior B_{med} as follows:

$$\begin{aligned} c_\uparrow &= \inf\{p' : P_{B_{med}}(p < p') \geq 1 - \alpha_c/2\} \\ c_\downarrow &= \sup\{p' : P_{B_{med}}(p < p') \leq \alpha_c/2\} \ . \end{aligned} \tag{6.3}$$

Bayesian upper confidence bound (BUCB): Finally, we can calculate a BUCB $p_{\updownarrow x}$ for each coin, using the credibility region together with the per-sample posterior distribution B_x. Here the BUCB is the probability under H_1 that the true parameter value p does not fall into the credibility region of p under H_0. $p_{\updownarrow x}$ is directly given by inserting the credibility region into the per-sample posterior distribution B_x:

$$\begin{aligned} p_{\updownarrow x} &= p_{\uparrow x} + p_{\downarrow x} \\ p_{\uparrow x} &= P_{B_x}(p_x > c_\uparrow) \text{ and} \\ p_{\downarrow x} &= P_{B_x}(p_x < c_\downarrow) \end{aligned} \tag{6.4}$$

Intuitively, selecting coins according to such a BUCB will focus on coins for which we are fairly certain that they will differ from H_0 when we observe more data.

Selecting the coins: We are now ready to select the two coins with index x_1, x_2 which are likely to differ the most, see Eq. (6.2) of our algorithm. We can simply find these two coins by maximizing the probabilities $p_{\uparrow x}, p_{\downarrow x}$:

$$x_1 = \arg\max_{x'} p_{\uparrow x'} \ , \ x_2 = \arg\max_{x'} p_{\downarrow x'} \tag{6.5}$$

This takes only linear time in respect to $|\mathcal{X}|$.

6.4.3. Multi-sample Test Statistic

To perform a MT in Step (5) of our AMT algorithm we require a test statistic which corrects for the adaptive sampling procedure. A closed-form solution for the distribution under H_0 proved to be intractable. Hence, we first perform separate single coin tests for each $x \in \mathcal{X}$. Formally, these tests are defined as follows.

Definition 6.6 (Single Coin Test)**.** *Let p_x be the success probability of coin Y_x, and $p_{\neg x} = \frac{1}{|\mathcal{X}|-1} \sum_{k \in (\mathcal{X} \setminus \{x\})} p_k$. A single coin test considers*

$$H_{0x} : \ p_x = p_{\neg x} \quad \textit{vs.} \quad H_{1x} : \ p_x \neq p_{\neg x}.$$

Here, H_{0x} is derived from H_0, where $p_x = p_{\neg x} = p$. Aggregating the single coin test results allows us in the second step to construct an omnibus test for H_0, where we reject H_0 if and only if any of the H_{0x} were rejected. To control the overall significance level under H_0, we apply the Bonferroni correction, setting $\alpha_x = \alpha/|\mathcal{X}|$ for each single coin test [Gar23]. As the Bonferroni correction is known to be conservative, we refer to the resulting reduction in power in our experiments and investigate it more closely in Appendix B.2.3.

Single coin test H_{0x} distribution: To perform the single coin test we require the distribution of h_x under H_{0x} (and thereby under H_0). Given that the shared p is known, and even if we perform adaptive sampling, this distribution is given by the following theorem:

Theorem 6.1 (Distribution of h_x under H_{0x})**.** *Let $H_{0x} : p = p_x$ be the null hypothesis, where p is given. Assume that the sampling strategy depends only on previous observations and not on the outcome of the current trial. Assume further that at iteration I the number of trials per coin N_{xI} is known. Then, for a coin Y_x, the number of heads h_x in the sample $\mathbf{Y}_{xI}$ after N_{xI} trials follows a Binomial distribution:*

$$h_x \sim \text{Bin}(N_{xI}, p).$$

The proof of Theorem 6.1 is given in Appendix B.1.

Estimating distribution parameters: In practical scenarios, the true parameter p is unknown. We address this by modeling the posterior distribution $\text{Beta}(a_{\neg x}, b_{\neg x})$ of p. First, we build the combined sample set $\mathbf{Y}_{\neg x} = \bigcup_{k \in (\mathcal{X} \setminus \{x\})} \mathbf{Y}_{kI}$. Here, we explicitly exclude the sample $\mathbf{Y}_{xI}$ so that the estimation of parameter p and, thereby, the H_{0x} distribution do not depend on the same data, which we will compare against that distribution later. We can now compute the number of heads $h_{\neg x} = \sum_{y \in \mathbf{Y}_{\neg x}} y$ and the number of tails $t_{\neg x} = N_{\neg x} - h_{\neg x}$ in $\mathbf{Y}_{\neg x}$.

Assigning a Beta prior $\text{Beta}(a, b)$ results in the posterior distributions $\text{Beta}(a_{\neg x}, b_{\neg x})$ of $p \mid \mathbf{Y}_{\neg x}$ with parameters $a_{\neg x} = 1 + h_{\neg x}, b_{\neg x} = 1 + t_{\neg x}$. Combining these Beta posteriors with the Binomial distributions under H_{0x} of Theorem 6.1 yields a Beta-Binomial posterior distribution for h_x:

$$h_x \sim \text{BetaBin}(N_{xI}, a_{\neg x}, b_{\neg x}). \tag{6.6}$$

Performing the test: Finally. we can use $\text{BetaBin}(N_{xI}, a_{\neg x}, b_{\neg x})$ to determine the upper and lower critical values $c_{\uparrow x}, c_{\downarrow x}$ at $\alpha_x/2$-levels for each single coin test:

$$\begin{aligned} c_{\uparrow x} &= \inf\{h'_x : P_{\text{BetaBin}}(h_x < h'_x) \geq 1 - \alpha_x/2\} \\ c_{\downarrow x} &= \sup\{h'_x : P_{\text{BetaBin}}(h_x < h'_x) \leq \alpha_x/2\}\ , \end{aligned} \tag{6.7}$$

and compare them against the observed number of heads h_{xI}. Combining the single coin test results into our omnibus test follows the simple rule if at least one single coin test is significant the omnibus test is also significant.

6.4.4. Properties and Application of AMT

In this section we will give a short hint how to apply AMT in practice and provide details on the runtime, computational complexity, and detectable difference of AMT.

6.4.4.1. Application of AMT

Before using AMT, one should follow the guidelines given in Section 6.1.1.2. To this end one first needs to perform a power analysis. For AMT, we can perform the power analysis

via simulation using the given H_0 and H_1 hypothesis. We then obtain an iteration limit I for which the power matches what is required. Here, I is related to the total sample size N by: $N = \sum_{x \in \mathcal{X}} N_{xI} = N_0 \cdot |\mathcal{X}| + 2I$. One now knows how many data selection iterations are required and can start collecting data.

AMT is designed for binomial data, thus its application requires that the obtained samples are binomial in nature. However, in some instances one can convert metric variables to binomial data, often with an acceptable loss in power. When Using AMT on Metric Variables, we need to distinguish two scenarios: (1) we have multiple samples $\mathbf{Y}_{\text{metric } x}$ of univariate metric observations $y_x \in \mathbb{R}$. (2) we have one sample $\mathbf{Y}_{\text{pair}}$ of paired (bivariate) metric observations $\mathbf{y}_n = (y_{1n}, y_{2n}) \in \mathbb{R}^2$ as often present in independence testing. For scenario (1), we can directly build on [Moo63]. We calculate the median of each sample and classify each data point as either being above or below-and-equal to the median, obtaining our required binary samples $\mathbf{Y}_x$. In scenario (2), we need to create 'pseudo'-samples to apply the transformation from scenario (1). We do this by conditioning the observations y_{2n} of one variable on the observations of the other variable, sorting them into $|\mathcal{X}|$ intervals $\text{int}_x = [\text{start}_x, \text{stop}_x], x \in \mathcal{X}$. That is $\mathbf{Y}_{\text{metric } x} = \{y_{2n} | y_{1n} \in \text{int}_x, n \in N_{\text{metric}}\}$. The same procedure can be extended to multivariate variables by conditioning on multidimensional intervals, similar to [FB19]. We can now apply the transformation from scenario (1).

After the binomial data is collected AMT performs the for adaptive sampling adjusted MT.

6.4.4.2. Detectable Coin Differences

A result from [HCN11] suggests that adaptive sampling is able to detect smaller differences $\delta p = |p_{x_1} - p_{x_2}|$ between coins than classical non-adaptive methods, which are theoretically limited by the combined sample size N. We outline how to adapt this result to AMT in the following:

Using the Beta distribution as a Bayesian estimate of the true underlying coin probabilities, we can follow the proof for normally distributed variables in [HCN11]. We will sketch this out in the following: (1) We use the relation that the Beta distribution $\text{Beta}(\alpha, \beta)$ can be approximated with a Gaussian distribution for an increasing sample size N. Here, the standard rule of thumb is that the parameters α and β should be larger than 10, which is given in our scenarios. This results in a vector of Gaussian distributions, one per coin. (2) We now have precisely the same conditions as in [HCN11] and can follow the proof therein. (3) The results from [HCN11] adapted to our work show that non-adaptive approaches require a coin difference $\delta p = \max_{x \in \mathcal{X}} p_x - \min_{x \in \mathcal{X}} p_x$ in $\Omega(\log(|\mathcal{X}|)$, while adaptive methods only require that δp grows (arbitrarily slowly) as a function of the number of samples $|\mathcal{X}|$.

6.4.4.3. Computational Complexity and Runtime

We noted that permutation testing becomes computationally demanding when working with adaptive strategies. This is due to the need for re-simulation of the iterative sampling process and the fact that one cannot employ re-shuffling of the samples, as in classical permutation testing. In this section, we provide details on re-simulation-based permutation

and some approximate time horizons and complexity comparisons to our test statistic, which does not require permutation testing.

Re-Simulation Based Permutation in a real-life application is performed at iteration I, and provides the null distribution for the subsequent test. It requires an estimate of the shared sample mean under H_0. In our evaluation, we use the ground truth mean; in application, one might use the average sample mean. Given the mean under H_0, one then performs the entire adaptive selection cycle (I iterations) R times. The test statistic is then computed for the resulting $|\mathcal{X}|$ samples, per repetition, yielding an empirical distribution of R distinct values. Thus, in contrast to re-shuffling, which requires R shuffle operations with complexity $O(N)$, re-simulation requires $I \cdot R$ operations, where a single operation is computationally more expensive as it requires the distance calculation and arg max operation, which itself can have a complexity ranging from $O(N)$, in our optimized case, to $O(N^2)$ with a trivial implementation.

Assuming that the computation of a complete adaptive sampling run (excluding data acquisition) of $I = 2000$ iterations takes 1 second. Then repeating such a selection $R = 10000$ times to obtain a good H_0 distribution approximation takes $10000s \approx 167min \approx 2.8h$. However, since parallelization across R is possible, the computational overhead can be reduced considerably. The selection step itself depends linearly on the number of coins $|\mathcal{X}|$, resulting in a total complexity of $O(|\mathcal{X}|glsIR)$ (without parallelization). Although I and R can be considered constants, they are very large constants compared to a typical $|\mathcal{X}|$. The H_0 distribution of our test statistic, on the other hand, is calculated independently of I, does not require Monte Carlo repetitions R, and therefore has only a complexity of $O(|\mathcal{X}|)$. Thus, our test statistic would result in a calculation time of $\frac{1}{2000}s$. The given example times approximately match what we observed during our experiments.

6.5. Experimental Design

The following section describes the baselines and variants, experimental settings and evaluation metrics.

6.5.1. Baselines and Variants

By replacing the *Coin Selection* method (Step (3)), and the used *Multi-sample Test* (Step (5)) in Algorithm 6.1, one can mimic different baselines and create variants of AMT. Here, baselines correspond to existing methods, while variants are different versions of AMT. In the following we will introduce the test statistics used for MT and the coin selection methods we applied.

6.5.1.1. Test Statistics

As test statistics we use: *(1) the χ^2 statistic*, *(2) the KW statistic*, *(3) the difference in means (Mean)* between each coin and the coin average, and *(4) the Beta-dist statistic* which is the 'counterpart' to our Beta sampling based on distributional distance. We compare them with our Beta-Binomial statistic. For all baselines and variants, we use a permutation testing approach based on 10000 re-simulations of the adaptive sampling process to obtain a good approximation under H_0 and to guarantee the required α error. Our new test statistic, with its H_{0x} distributions, is not reliant on such computationally expensive re-simulation.

(1) The χ^2 statistic: After I iterations, we perform a χ^2-test on the observed samples $\mathbf{Y}_1, \mathbf{Y}_1, ..., \mathbf{Y}_{|\mathcal{X}|}$, to test our hypothesis.

In the adaptive setting, we no longer have i.i.d. samples. Thus, the H_0 is not distributed χ^2. We tested this in simulations, see Section 6.6.3. The distribution is 'rounder' at the peak, but 'narrower' in total. Further, the distribution converges to the χ^2 distribution of two coins (df=1). However, increasing the number of coins $|\mathcal{X}|$ adds additional variance, which limits this convergence.

Accordingly, using adaptive sampling, the standard p_{value} of the χ^2-test is too low, which is why we precomputed the significance level under H_0 using permutation testing with 10000 simulations.

(2) The KW statistic: After I iterations, we perform a KW-test (Analysis of Variance (ANOVA) on ranks) on the observed sets $\mathbf{Y}_1, \mathbf{Y}_1, ..., \mathbf{Y}_{|\mathcal{X}|}$, to test our hypothesis. This test statistic is also distributed as χ^2 with i.i.d. sampling, but not in the adaptive setting. We precomputed the significance level accordingly to the χ^2 test.

(3) The difference in means statistic: Calculate the per-sample success probabilities $\bar{p}_x = \frac{h_{xi}}{N_x} \sum_{n \in \{1,...,N_x\}} y_{xn}$ of each sample $\mathbf{Y}_x$. Calculate $|\mathcal{X}|$ reference means $\bar{p}_{\neg x}$ of all combined samples, not including the coin Y_x: $\bar{p}_{\neg x} = \frac{1}{|\mathcal{X}|-1} \sum_{k \in (\mathcal{X} \setminus \{x\})} \bar{p}_k$. Excluding Y_x again ensures independence of the reference mean $\bar{p}_{\neg x}$ to the coin mean $\bar{p}_x$. The statistic is then the sum of weighted square differences SWSD between $\bar{p}_{\neg x}$ and $\bar{p}_x$: SWSD $= \sum_{x \in \mathcal{X}} N_x \cdot (\bar{p}_{\neg x} - \bar{p}_x)^2$.

Here, the weight by sample size N_x ensures that larger samples (having a more accurate estimation of $\bar{p}_x$) have a higher impact on the statistic.

(4) The Beta-dist statistic: To investigate our adaptive Beta sampling strategy more closely, we derived a test statistic based on the same underlying principles. It calculates the difference d_x between the Beta distribution based on a sample $\mathbf{Y}_x$ and the Beta distribution based on a sample $\mathbf{Y}_{\neg x}$ of all observations except the ones from Y_x. The statistic then uses the maximum of the distances $d_x, x \in \mathcal{X}$.

Detailed calculation: We note that the exact calculation closely follows the steps of our Beta sampling and may be skipped, we only provide it here for reference. First, based on the number of heads h_x and tails t_x in sample $\mathbf{Y}_x$, we calculate the distributions B_x of $p_x \mid \mathbf{Y}_x$ under H_1. Here, we assign Beta priors $\text{Beta}(a, b)$ to each success probability p_x, where we set $a = b = 1$ reflecting uniform priors on $(0, 1)$. The resulting posterior distributions of $p_x \mid \mathbf{Y}_x$ are then again Beta distributions $\text{Beta}(a_x, b_x)$ with parameters $a_x = 1 + h_x, b_x = 1 + t_x$. **Note:** This step is equivalent to our Beta sampling.

Next, we compute a reference Beta distribution, $B_{\neg x}$, for the combined sample set $\mathbf{Y}_{\neg x}$ of all coins *excluding* $\mathbf{Y}_{glsx}$, i.e., $\mathbf{Y}_{\neg x} = \bigcup_{k \in (\mathcal{X} \setminus \{x\})} \mathbf{Y}_k$. This yields $B_{\neg x} \sim \text{Beta}(1 + h_{\neg x}, 1 + t_{\neg x})$. **Note:** This step slightly deviates from our Beta sampling by using the mean instead of the median.

Using $B_{\neg x}$, we calculate the two-sided 95% credibility region for p_x:

$$\begin{aligned} c_{\uparrow x} &= \inf\{p' : P_{B_{\neg x}}(p_x < p') \geq 0.975\} \\ c_{\downarrow x} &= \sup\{p' : P_{B_{\neg x}}(p_x < p') \leq 0.025\} \ . \end{aligned} \tag{6.8}$$

Note: This step is equivalent to our Beta sampling, only using the slightly adapted $B_{\neg x}$ distribution.

Using B_x, we now calculate the probabilities $p_{\uparrow x}, p_{\downarrow x}$ of p_x being above or below this bound:

$$p_{\uparrow x} = P_{B_x}(p_x > c_{\uparrow x}) \text{ and } p_{\downarrow x} = P_{B_x}(p_x < c_{\downarrow x}). \tag{6.9}$$

Note: This step is equivalent to our Beta sampling.

Finally, the difference d_x is defined as $d_x = p_{\uparrow x} + p_{\downarrow x}$, i.e., the combined probability that a coin differs from H_0. We then use the maximum of d_x as a statistic:

$$\text{Beta-dist} = \max_{x \in \mathcal{X}}(d_x) \ . \tag{6.10}$$

6.5.1.2. Coin Selection

For coin selection, we use on the one hand classical non-adaptive sampling strategies: *(1) Random Sampling* ***(Rand)***, *(2) Space Filling* ***(Equal)***, and an *(3) Oracle* as upper bound.

On the other hand, we use standard methods for adaptive sampling: *(1) Greedy*, *(2) Epsilon Greedy* ***(Eps)***, *(3) Vanishing Epsilon Greedy* ***(Slow)***, and *(4) Weighted Monte Carlo* ***(MC)***. The first three methods—Greedy, Eps, and Slow—are described in [SB18b]. Weighted MC is detailed in [Liu04].

We combine them with coin differences used to weigh the importance of coins: *(1) Difference in Mean*, *(2) P-Value Difference* ***(PVal)***, and *(3) our proposed posterior distance* ***(Beta)***

from Section 6.4.2. For PVal and Beta, we only consider 'Greedy' because exploration-exploitation is built into the difference.

Furthermore, we apply hybrid methods that combine sampling and weighting: *(1) Thompson Sampling* ***(TS)*** with Beta prior [Rus+20], which is a standard method in the Bandit setting, and *(2)* ***TS5*** which is Thompson Sampling (TS) with five times higher exploitation.

Non-Adaptive Setting: In the non-adaptive setting, we choose coins Y_{x_1} and Y_{x_2} without considering any knowledge of past observations. These strategies fulfill the iid assumption and are equivalent to classical sampling strategies.

(1) Random Sampling: We choose coins Y_{x_1}, Y_{x_2} for each iteration i at random (uniformly from $U[1, |\mathcal{X}|]$).

(2) Space Filling: We choose coins Y_{x_1}, Y_{x_2} for each iteration i so that each coin is observed approximately $i/|\mathcal{X}|$ times. (If $\mod(i/|\mathcal{X}|) = 0$ exactly $i/|\mathcal{X}|$ observations per coin.)

(3) Oracle: At the beginning of sampling, one obtains the information about which coins differ most (from some 'higher entity'). With no difference (corresponding to H_0), the Space Filling strategy is employed. With a difference, only the two most different coins are sampled. This strategy gives us an upper bound on the power.

Adaptive Setting: In the adaptive setting, we select coins Y_{x_1}, Y_{x_2} based on past observations. Therefore, new observations depend on the old observations, thereby breaking the i.i.d. assumption.

(1) Greedy: Always choose the coins with the maximum difference.

Note: Some used differences already contain an exploration term, either explicitly or implicitly, through their non-deterministic nature. In such cases, we do not use the epsilon strategies.

(2) Epsilon Greedy: Choose the coins with the maximum difference in $1 - \epsilon$ of the cases, and use random sampling with a probability of ϵ.

(3) Vanishing Epsilon Greedy: Decrease the probability of random sampling ϵ linearly with increasing iterations i. At $i = I : \epsilon = 0$.

(4) Weighted MC: For all coins, we calculate the difference between samples $d(\mathbf{Y}_{x_1 i}, \mathbf{Y}_{x_2 i})$. Choose a coin randomly weighted by a function of the difference. Here, a coin with a higher difference gains a higher chance of selection. Commonly, linear or exponential weighting is used.

Coin Differences: The adaptive method require that one can quantify the difference between coins. The adaptive method use this difference to weigh the importance of coins resulting in a specific exploration-exploitation tradeoff on coin selection.

(1) Simple Mean: Calculate the mean of each coin. Then select the two coins with the largest difference in means.

Note: This strategy does not account for sample size or uncertainty.

(2) P-Value Difference: Use the p-value of a two-sample test. For example, perform a pair-wise χ^2-test on every possible sample pair $(\mathbf{Y}_{x_1 i}, \mathbf{Y}_{x_2 i}), x_1, x_2 \in \mathcal{X}, x_1 \neq x_2$. Take the

inverse of the p-value to obtain a difference: $d_{\chi^2}(\mathbf{Y}_{x_1 i}, \mathbf{Y}_{x_2 i}) = (1 - p_{value}(\mathbf{Y}_{x_1 i}, \mathbf{Y}_{x_2 i}))$. Select the coins with the highest difference (lowest p-value).

Hybrid Methods: The hybrid methods do not directly calculate a difference between coins, instead they select stochastically according to a given prior distribution, implicitly balancing exploration and exploitation.

(1) Thompson Sampling (TS): Use Thompson Sampling with a Beta prior. Sample p'_x from the prior. Select the two coins with the largest difference in sampled probability p'_x.

(2) Thompson Sampling with higher exploitation (TS5): Increase the exploitation ratio of TS by sampling multiple $p'_{xk}, k \in \{1, \ldots, K\}$ and using the mean $\bar{p}'_x$ for the decision. We choose $K = 5$ in our experiments.

Note: For a high K, this strategy converges to the *Simple Mean* difference.

6.5.2. Experimental Settings and Evaluation Metrics

We vary the number of coins ($|\mathcal{X}| \in [5, 10, 15, 20]$), the number of iterations before MT is performed ($I \in (1 \ldots 2000)$), the number of biased coins ($M \in [0, 1, 2, |\mathcal{X}|/2]$), the difference in success probability ($\delta p \in [0.15, 0.1, 0.05]$), and the significance level ($\alpha \in [0.05, 0.025, 0.01]$).

To calculate the power and type I error, we perform each setup 10000 times under H_0 and H_1. We evaluate the test statistic and coin selection per iteration. With this, we plot the empirical distribution under H_0 and H_1. We investigate changes in distributions under H_0 due to adaptive sampling, and how adaptive sampling changes the difference between the empirical H_0 and H_1 distributions. We further compare the results between different variants and non-adaptive baselines.

Metrics : We calculate the critical values, power, type I error for our evaluation. Further we report the gained power, and the separation of distributions:

Definition 6.7 (Power Gain)**.** *For each variant V, the power gain represents the relative increase in the probability of correctly rejecting the null hypothesis compared to the non-adaptive Equal sampling method (Eq). It is calculated as:*

$$gain_V = \frac{power_V - power_{Eq}}{power_{Eq}}$$

This metric quantifies the efficiency improvement of an adaptive sampling strategy over a baseline uniform allocation.

Definition 6.8 (Separation of Distributions of H_0 and H_1 (Distributional Distance))**.** *The separation between the null and alternative distributions of a test statistic is a measure of how distinct they are. Let T be a test statistic, and let $p_0(t)$ and $p_1(t)$ be its probability density functions (PDFs) under the null hypothesis (H_0) and alternative hypothesis (H_1), respectively. A common way to quantify this separation is by measuring the area of the symmetric difference between the two distributions, also known as the Total Variation Distance*

(TVD). A larger area indicates a greater separation, making it easier for a test to distinguish between the two hypotheses and thus resulting in higher test power.

$$TVD = \frac{1}{2} \int_{-\infty}^{\infty} |p_0(t) - p_1(t)| \, \delta t \tag{6.11}$$

Hardware: All experiments were run using Python under Linux on an 8-core AMD Ryzen 7 4750U and 15 GB of RAM.

6.6. Evaluation

In this section we will evaluate AMT. First, Section 6.6.1 will focus on the test power. We will then verify the controlled type I error of AMT in Section 6.6.2. In Section 6.6.3, we will study how adaptive selection impacts the H_0 distribution of the χ^2 test statistics, and how it increases the distance between the H_0 and H_1 distribution in Section 6.6.4.

6.6.1. AMT Power

This section will first provide a large-scale overview of all results in Table 6.2. We will then proceed to a more thorough investigation of the power behavior over sample size of various sampling strategies in combination with our Beta-Binomial test in Figure 6.2. Finally, we compare the power of our analytic test with computational more expensive permutation based tests in Figure 6.3. We show more results with similar patterns in Appendix B.2.1.

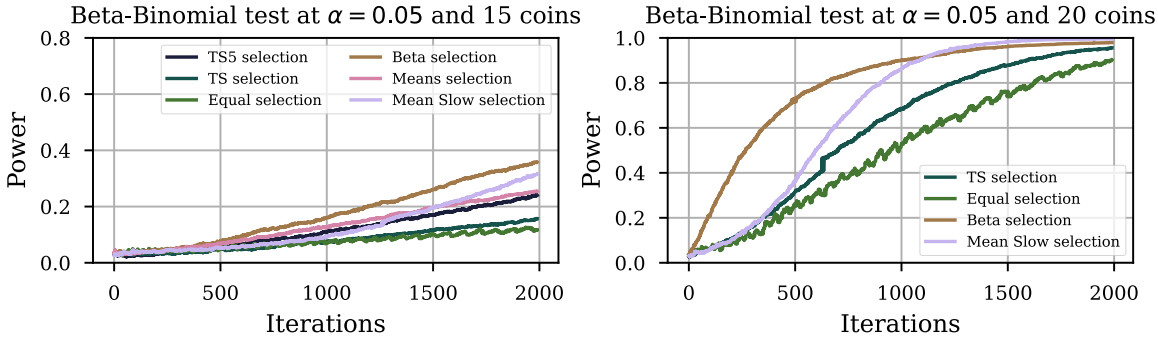
The Large-scale Overview in Table 6.2 reports the power of adaptive sampling in comparison to non-adaptive sampling for different test statistics and sampling methods. The table shows how much power we gain by using adaptive sampling at an α-level of 5% and a difference between coins of $\delta p = 0.05$, with one plated coin ($M = 1$). We vary the number of coins $|\mathcal{X}|$, and the number of iterations I, at which the test is performed. We see that in any scenario, every adaptive strategy is at least as powerful as the corresponding non-adaptive strategy, with a gain of up to 485%. In particular, our Beta sampling (marked in bold) outperforms all other strategies across all test statistics. Finally, we observe that the gain increases with the number of iterations after which the test is performed. This observation is expected as sampling strategies transition from exploration to exploitation.

Table 6.2.: Power and Gain of Multi-sample Testing with Adaptive vs Non-Adaptive Sampling at $\delta p = 0.05$

		Mean		Chi-squared		Kruskal-Wallis		Beta-Dist		Beta-Binomial	
$\lvert\mathcal{X}\rvert$ / I	**Method**	**Power**	**Gain (%)**	**Power**	**Gain (%)**	**Power**	**Gain (%)**	**Power**	**Gain (%)**	**Power**	**Gain (%)**
10 / 1000	Equal	0.068	–	0.068	–	0.068	–	0.071	–	0.073	–
	TS5	0.166	144	0.144	111	0.144	111	0.168	138	0.132	80
	TS	0.100	47	0.092	34	0.092	34	0.101	42	0.088	20
	Beta	**0.201**	**196**	**0.189**	**177**	**0.189**	**177**	**0.198**	**180**	**0.166**	**126**
	Means	0.187	175	0.174	155	0.174	155	0.173	145	0.137	87
	Mean Slow	0.102	51	0.100	46	0.100	46	0.113	60	0.097	33
15 / 1000	Equal	0.043	–	0.043	–	0.043	–	0.048	–	0.045	–
	TS5	0.105	146	0.081	88	0.081	88	0.102	114	0.080	76
	TS	0.050	17	0.047	10	0.047	10	0.056	17	0.045	0
	Beta	**0.154**	**262**	**0.139**	**224**	**0.139**	**224**	**0.145**	**205**	**0.121**	**167**
	Means	0.134	215	0.133	211	0.133	211	0.130	174	0.096	113
	Mean Slow	0.067	57	0.068	58	0.068	58	0.075	57	0.059	30
20 / 1000	Equal	0.041	–	0.041	–	0.041	–	0.029	–	0.027	–
	TS5	0.075	85	0.064	58	0.064	58	0.081	183	0.056	111
	TS	0.042	3	0.042	3	0.042	3	0.045	57	0.034	27
	Beta	**0.131**	**222**	**0.120**	**195**	**0.120**	**195**	**0.126**	**342**	**0.099**	**272**
	Means	0.119	194	0.110	171	0.110	171	0.110	285	0.077	189
	Mean Slow	0.063	56	0.057	40	0.057	40	0.066	131	0.053	100
10 / 1500	Equal	0.101	–	0.101	–	0.101	–	0.100	–	0.110	–
	TS5	0.244	142	0.230	129	0.230	129	0.261	161	0.238	116
	TS	0.146	45	0.136	35	0.136	35	0.148	48	0.144	32
	Beta	**0.326**	**223**	**0.320**	**218**	**0.320**	**218**	**0.337**	**237**	**0.286**	**160**
	Means	0.265	163	0.259	157	0.259	157	0.255	155	0.220	100
	Mean Slow	0.201	100	0.192	91	0.192	91	0.218	118	0.209	90
15 / 1500	Equal	0.051	–	0.052	–	0.052	–	0.063	–	0.062	–
	TS5	0.168	228	0.141	172	0.141	172	0.167	166	0.135	118
	TS	0.071	39	0.069	33	0.069	33	0.090	43	0.074	20
	Beta	**0.256**	**397**	**0.245**	**375**	**0.245**	**375**	**0.241**	**283**	**0.214**	**247**
	Means	0.203	295	0.204	296	0.204	296	0.201	220	0.160	159
	Mean Slow	0.145	181	0.134	159	0.134	159	0.165	162	0.144	133
20 / 1500	Equal	0.042	–	0.042	–	0.042	–	0.042	–	0.039	–
	TS5	0.115	175	0.099	138	0.099	138	0.129	208	0.095	142
	TS	0.058	40	0.056	35	0.056	35	0.059	41	0.050	27
	Beta	**0.206**	**396**	**0.182**	**338**	**0.182**	**338**	**0.191**	**354**	**0.160**	**306**
	Means	0.174	318	0.162	289	0.162	289	0.159	278	0.125	217
	Mean Slow	0.111	166	0.105	153	0.105	153	0.126	200	0.105	166
10 / 2000	Equal	0.139	–	0.139	–	0.139	–	0.162	–	0.151	–
	TS5	0.356	156	0.331	138	0.331	138	0.381	135	0.348	131
	TS	0.202	45	0.194	40	0.194	40	0.226	39	0.208	38
	Beta	**0.454**	**227**	**0.457**	**229**	**0.457**	**229**	**0.468**	**188**	**0.414**	**174**
	Means	0.347	150	0.348	150	0.348	150	0.346	113	0.302	100
	Mean Slow	0.373	168	0.368	165	0.368	165	0.391	141	0.369	144
15 / 2000	Equal	0.064	–	0.064	–	0.064	–	0.081	–	0.077	–
	TS5	0.220	245	0.200	214	0.200	214	0.239	195	0.201	161
	TS	0.106	67	0.100	56	0.100	56	0.131	61	0.108	39
	Beta	**0.339**	**432**	**0.329**	**416**	**0.329**	**416**	**0.337**	**316**	**0.306**	**297**
	Means	0.262	310	0.269	322	0.269	322	0.260	221	0.219	183
	Mean Slow	0.256	301	0.234	267	0.234	267	0.289	256	0.261	239
20 / 2000	Equal	0.049	–	0.049	–	0.049	–	0.057	–	0.048	–
	TS5	0.162	227	0.137	177	0.137	177	0.173	204	0.142	197
	TS	0.072	46	0.069	40	0.069	40	0.081	41	0.067	41
	Beta	**0.289**	**485**	**0.274**	**454**	**0.274**	**454**	**0.280**	**391**	**0.249**	**421**
	Means	0.219	343	0.215	335	0.215	335	0.213	274	0.177	271
	Mean Slow	0.213	331	0.193	290	0.193	290	0.231	306	0.195	309

Comparing the Power Behavior of Sampling Strategies gives insights on how data efficient a sampling strategy is, and how it allocates its resources on exploration and exploitation. We investigate this behavior once for a hard problem (δp is small) where power can only be gained slowly with sample size, and once for a problem with larger δp where one can observe the whole power spectrum (power $\in [0, 1]$).

Figure 6.2a investigates the power of our Beta-Binomial test with various sampling strategies for a range of sample sizes, with $M = 1$, $|\mathcal{X}| = 15$, and $\delta p = 0.05$. We can see that the sampling strategy significantly impacts the resulting power. A direct comparison of TS and TS5 shows that the slightly more exploitative TS5 strategy provides a significant increase in power. Conversely, comparing the purely exploitative Mean strategy to the Mean Slow strategy (a linearly vanishing ϵ-greedy method) reveals that excessive exploitation harms power. This is because the true fake coin is potentially never found. This highlights the well-known exploration-exploitation tradeoff [Tou14] and the sensitivity of these methods to hyperparameter selection. However, our Beta sampling does not require such a hyperparameter and consistently performs best.



(a) $\delta p = 0.05$ (b) $\delta p = 0.15$

Figure 6.2.: The impact of different sampling strategies.

Similarly, in Figure 6.2b for $m = 1$, $n = 20$, and $\delta p = 0.15$, we can observe the full power spectrum. We see again how our Beta sampling gains power much faster, only requiring *half* the sample size of Equal sampling to arrive at a power of 80%. The second best strategy is the Mean Slow strategy, requiring $\frac{2}{3}$ the sample size of Equal sampling. Here, we can observe the typical linearly vanishing exploration-exploitation behavior, where at the low sample sizes Mean Slow behaves exactly like the Equal strategy, and at the end exploits heavily, catching up to our Beta sampling (a bit late) at around 90% power. We further note that Mean Slow is very sensitive to its epsilon hyperparameter, with wrong selection possibly performing worse then Equal selection. Standard TS does not perform significantly better then Equal sampling.

We can conclude, that the sampling strategy has a significant impact on the test power, with any adaptive sampling strategy performing equal or better then non adaptive Equal sampling.

The Comparison of Test Statistics under non-adaptive (Equal) and adaptive (Beta) sampling provides insides if using different test statistics brings any benefit or drawback in terms of power.

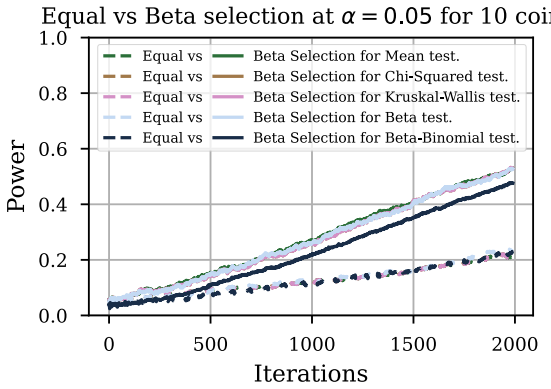


Figure 6.3.: Beta vs. Equal sampling, $\delta p = 0.05$.

From Figure 6.3, we see that for a fixed sampling method all test statistics result in similar power, i.e., the choice of test statistic has a minimal impact on power. However, comparing the power of non-adaptive sampling and adaptive sampling results in a large difference. This suggests that the sampling strategy itself is the primary driver of power, not the specific test statistic. One can observe that under adaptive sampling, our Beta-Binomial statistic is slightly less powerful than the other permutation-based statistics. We attribute this to the Bonferroni correction and will investigate and discuss it in Appendix B.2.3. Nonetheless, our Beta-Binomial statistic still significantly outperforms non-adaptive Equal sampling. Furthermore, our Beta-Binomial statistic offers the distinct advantage of being based on the analytic H_{0x} distributions, thus requiring far less computational power than permutation testing, as we highlighted in Section 6.4.4.3.

6.6.2. AMT Type I Error

In this section, we will verifying the type I error control for different iteration limits I and different significance levels α for our test statistic using the analytic H_0 distribution, and compare it to other test statistics for which we estimate the H_0 distribution with re-simulation based permutation. We show more results with different configurations in Appendix B.2.1.

To obtain a reference, in Figure 6.4 we will compare tests based on our test statistic to tests based on the baseline test statistics using non-adaptive Equal sampling. We can observe that all test statistics conform to the required significance level, independently of sample iteration or number of coins.

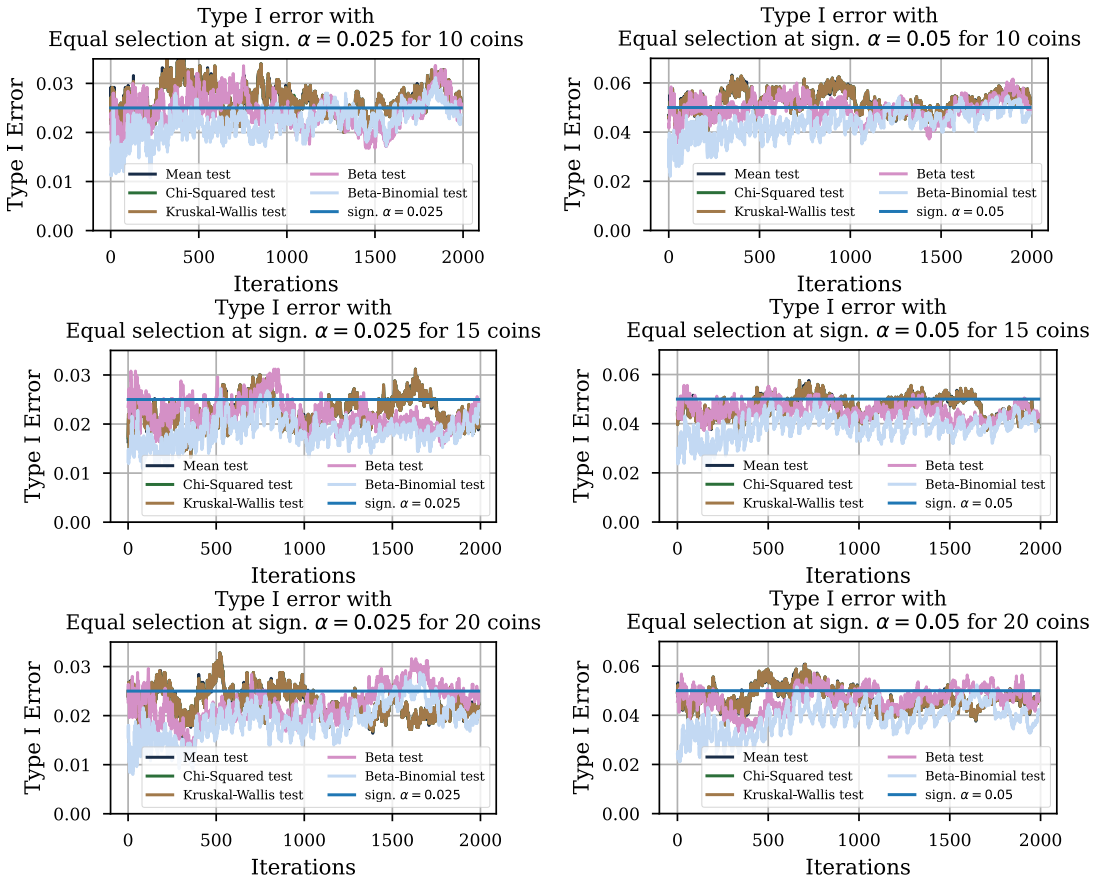


Figure 6.4.: Type I error under non-adaptive equal sampling: we observe that all tests meet the required significance level.

In Figure 6.5 we compare the same tests, but now using our adaptive Beta sampling. We observe that our test statistic always guarantees the required type I error. It is, however, somewhat conservative, a known property when using the Bonferroni correction which we will investigate in Appendix B.2.3. The behavior of our test statistic is consistent across both iteration count and the number of coins, always resulting in a lower type I error that satisfies the required significance level. On the other hand, permutation-based tests are in some cases overly optimistic, indicating that even tests based on re-simulating the adaptive process are sensitive to non-i.i.d. data.

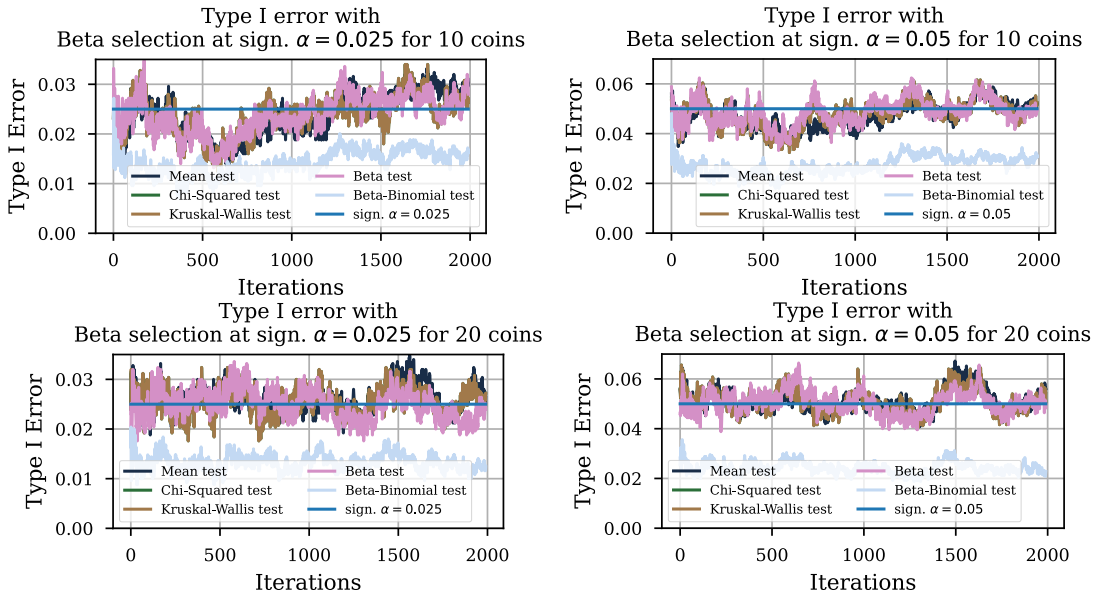


Figure 6.5.: Type I error under adaptive beta sampling: we observe that a test based on our test statistic, although being slightly conservative due to the Bonferroni correction, consistently conforms to the required significance level. The permutation-based tests are not that conservative and can be overly optimistic.

In our power analysis we observed, that the used sampling strategies had an significant impact on the power while the test statistic where negligible. In Figure 6.6 we check if the type I error rate is effected similarly strong when fixing the test to our Beta-Binomial statistic but varying the sampling strategies. We observe that our test conforms to the required type I error rate, regardless of the sampling strategy used, thus verifying that the gained power is not due to a increase in type I error. This holds across iteration count and number of coins, and aligns with our proof, which suggested that the used sampling strategy has negligible impact on the H_0 distribution.

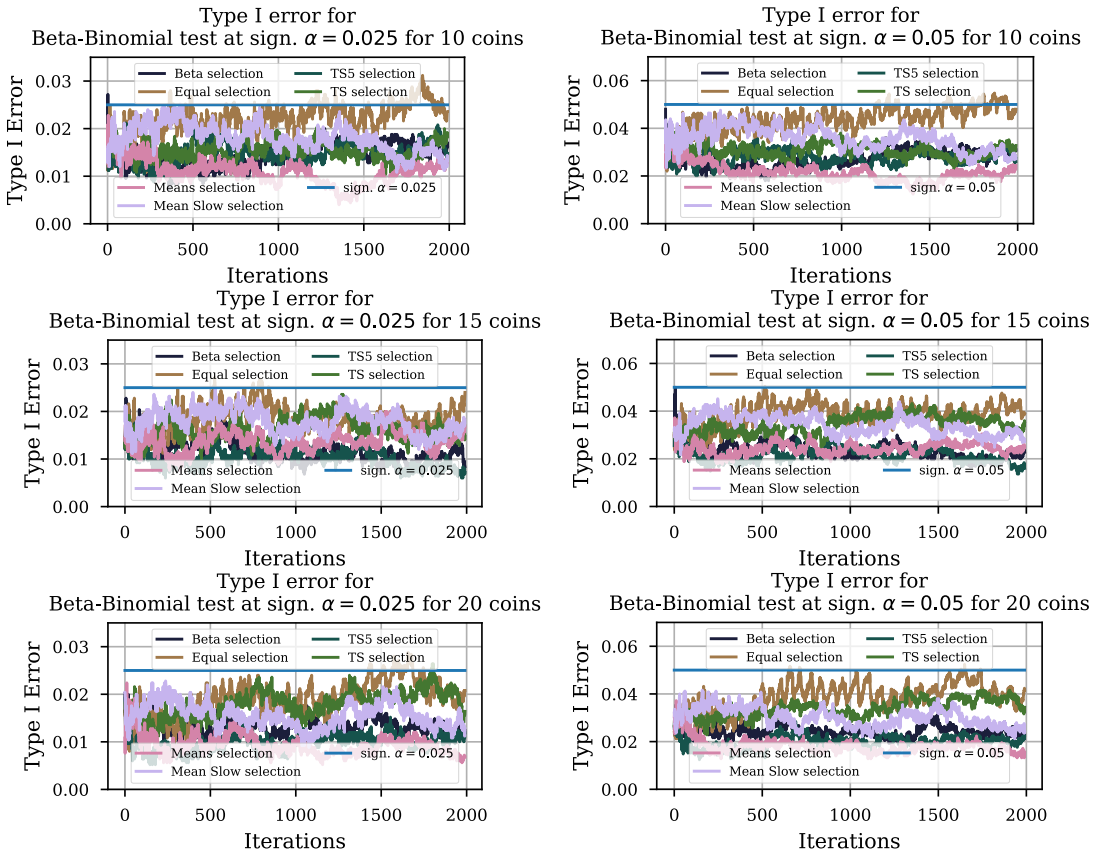


Figure 6.6.: Type I error of the test based on our Beta-Binomial statistic for different sampling strategies – we observe that our test conforms to the required type I error rate, regardless of the sampling strategy used. Further, the behavior over iteration count and number of coins is consistent.

We can conclude that our new test statistic manages to correct for our adaptive sampling, keeping the type I error under the required threshold, while providing a more powerful test.

6.6.3. How AMT Changes the H_0 Distribution of the χ^2 Statistic

We know that adaptive sampling violates the i.i.d. assumptions on which the analytic H_0 distributions of many classical test statistics rely. In this section, we will investigate how this violation affects the distribution of the standard χ^2 statistic, which is used in the KW-test and in the χ^2 test.

In Figure 6.7 we see a large-scale overview of the behavior of the H_0 distribution and the impact on the critical values over all iterations and for different numbers of coins. We observe that the distribution is initially spread out. Being equal to the standard χ^2 distribution at iteration 0, it then converges within the first 200 iterations to a distribution with lower variance, maintaining a consistent shape and width thereafter. This convergence is more pronounced for a smaller number of coins.

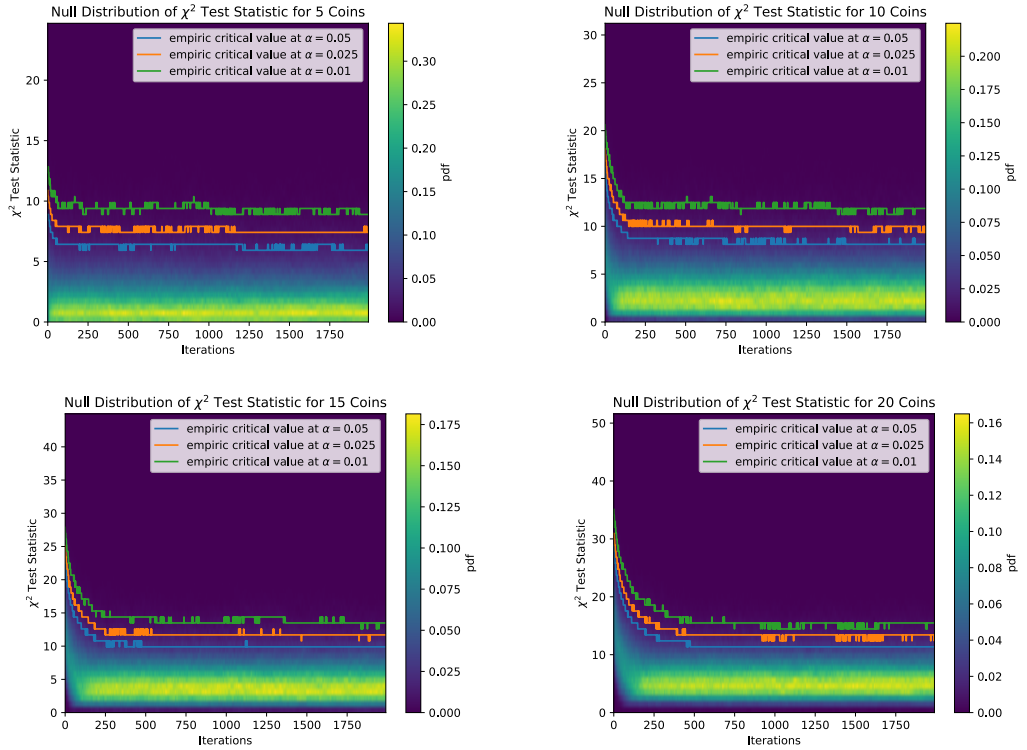


Figure 6.7.: A large-scale overview of the behavior of the H_0 distribution of the χ^2 statistic – it reveals that, compared to the classical χ^2 distribution, the statistic converges to a distribution with significantly lower variance.

In Figure 6.8 we have a closer look at this convergence. We plot the distribution of the classical χ^2 statistic (assuming i.i.d. data), the empirical distribution, and a χ^2 fit to the sampled data. We do this at different iteration counts, and for 5 and 20 coins. We can observe that the variance and mean of the empirical distribution decreases rapidly within the first 250 iterations, for 5 and 20 coins as well. Comparing the χ^2 fit with the empirical distribution, we find that: All distributions start out as the standard χ^2 distribution (iteration 0); In the first 50 iterations the empirical distribution is more spread out then its fit; However, this changes with increasing iterations (from 250 iterations onward), leading to a clear peak compared to the χ^2 fit. We conclude that with adaptive sampling, the empirical distribution of the χ^2 statistic is clearly no longer χ^2 distributed. That is, it is not even χ^2 distributed with a change in degrees of freedom (DOF).

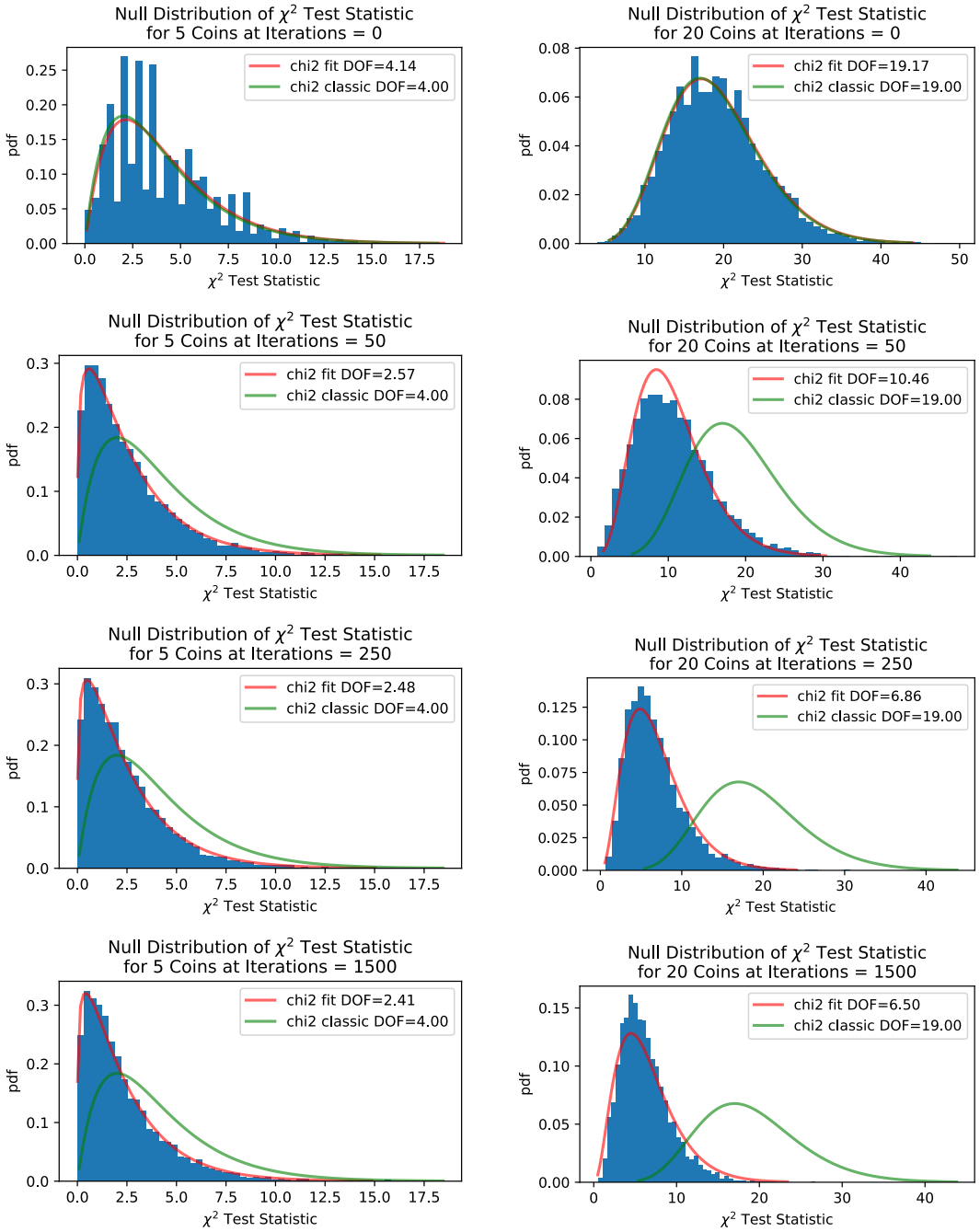


Figure 6.8.: A detailed investigation of the convergence behavior of the H_0 distribution of the χ^2 statistic. It reveals that the H_0 distribution converges to a clearly non-χ^2 distribution with a lower mean and variance.

6.6.4. Increase in Distributional Distance due to Adaptive Sampling

In the previous section, we have investigated how adaptive sampling changes the H_0 distribution of classical test statistics. In this section, we will investigate how adaptive sampling yields a greater difference between the H_0 and H_1 distributions compared to non-adaptive sampling. To distinguish between distributions, we use the empirical TVD, as described in Section 6.5.2. We note that the difference between the H_0 and H_1 distributions translates directly into test power because distributions with a higher difference are easier to distinguish. An investigation of this difference for our Beta-Binomial statistic is not straight forward, as it is calculated per coin and thus has N dimensions. As a proxy we compare our Beta statistic, which bases on the assumptions made by our beta selection strategy, with the χ^2 statistic.

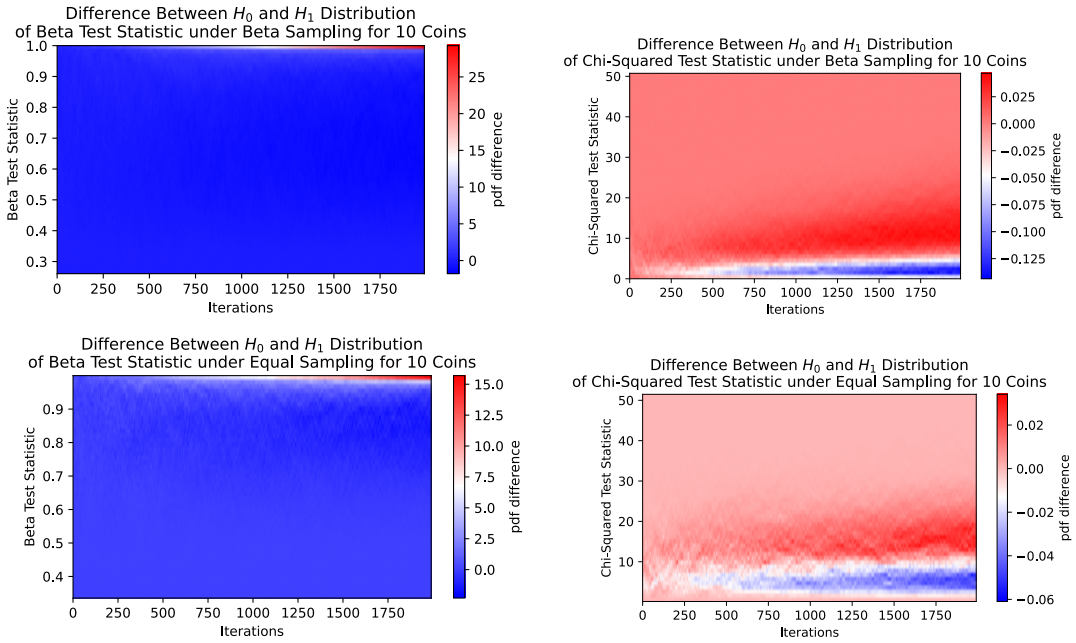


Figure 6.9.: Difference between the H_0 and H_1 distribution across all iterations; Top Beta sampling, bottom Equal sampling.

Figure 6.9 visualizes the difference between the H_0 and H_1 distribution across all iterations for Beta sampling and the χ^2 and Beta statistic. This gives an intuition of the value range in which the two distributions reside.

Beta statistic (left): What immediately stands out is that the value range of the statistic is bound to the interval from 0 to 1, directly following from the beta distribution. This qualifies the statistic as an e-value statistic [RW25]. Further, we note a narrow region (red) near a statistic value of 1 for which the distance between H_1 and H_0 is highest. That is, because most of the probability mass under H_1 falls into this region (which can be derived from the significant color difference with saturated colors). We will investigate this narrow region later, in Figure 6.10.

χ^2 statistic (right): The χ^2 statistic, on the other hand, does not have such an upper bound. Here, the probability mass under H_0 builds up in the lower half of the plot, but the

variance stops to decrease at higher iterations. Further, the distribution under H_1 is spread out over the whole value range (red regions on both sides of the blue region), leading to a smaller differences between the distributions (as we can see from the less saturated colors).
Beta vs Equal sampling (top vs bottom): Comparing the top plots with the bottom plots, we see that the difference between distributions for equal sampling is not as pronounced (less saturated colors) and the distributions are more spread out. This holds for both distributions. However, the Beta statistic still has a relatively significant distributional difference even under equal sampling, while the distance of the χ^2 statistic is less pronounced.

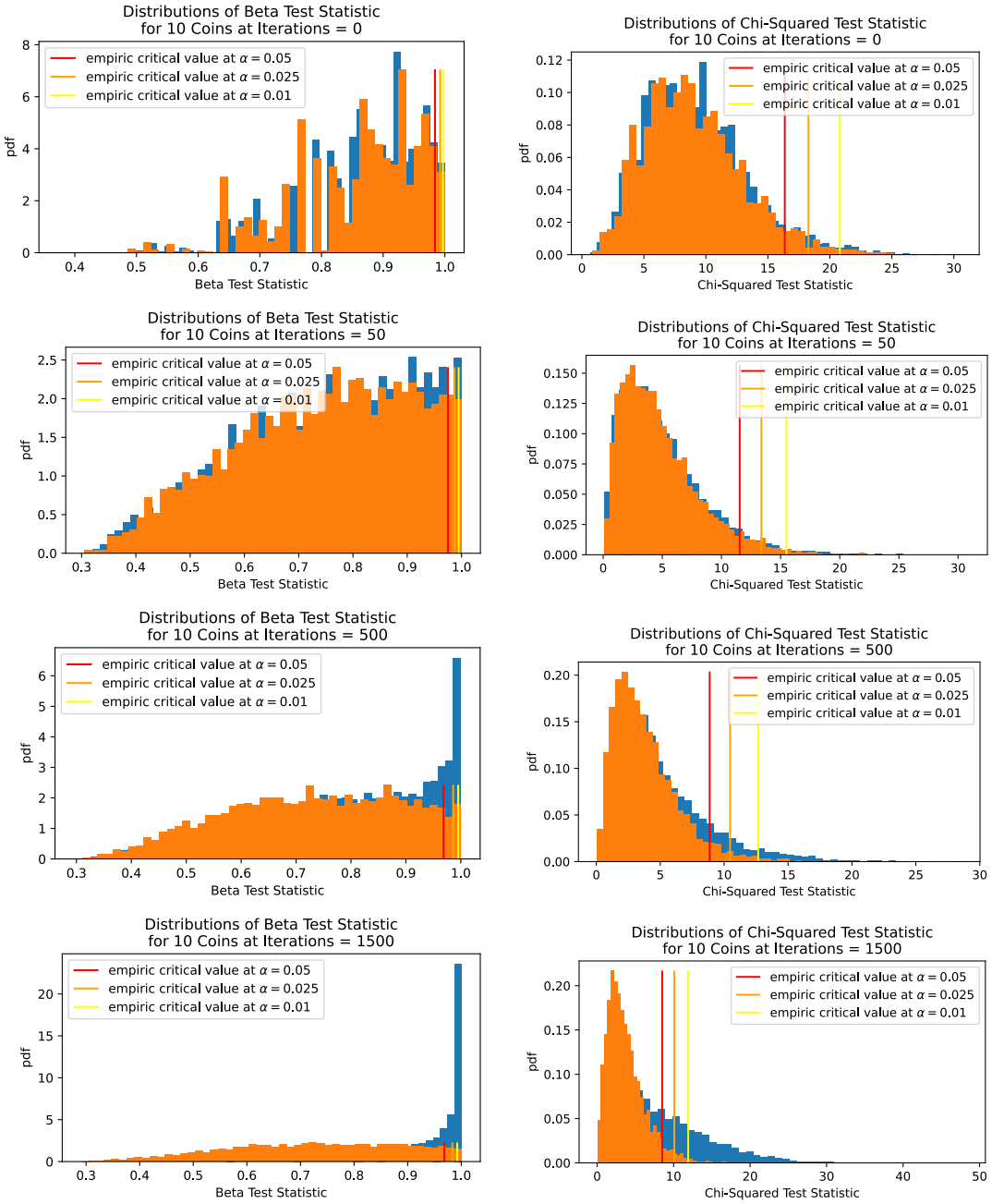


Figure 6.10.: Difference between the H_0 (orange) and H_1 (blue) distributions at a given iteration.

In Figure 6.10 we investigate these different behaviors between the statistics further, by plotting the histograms under Beta sampling at different iterations. We observe that the statistics behave inversely with increasing sample size: The H_0 distribution of the Beta statistic becomes more spread out and the H_1 distribution converges towards 1, explaining the pronounced peak in Figure 6.9, whereas the H_0 distribution of the χ^2 statistic becomes steeper and the H_1 distribution spreads out.

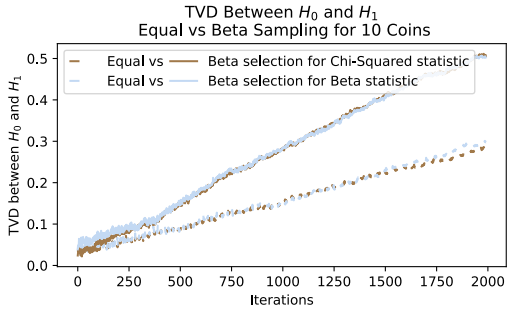


Figure 6.11.: The TVD between the H_0 and H_1 distribution under Equal vs Beta sampling for the χ^2 and the Beta statistic.

Looking, however, at the TVD between the H_0 and H_1 distributions, in Figure 6.11, shows that the difference is similar for both test statistics. In contrast, the TVD is significantly higher when using adaptive sampling compared to Equal sampling. This supports our previous findings from the power analysis, that the main driver of power is the adaptive sampling strategy, not the test statistic.

6.7. Chapter Conclusion

In this chapter we proposed Adaptive Multi-sample Testing (AMT) an adaptive solution to the knowledge discovery task of Multi-sample Testing, which has had only little attention in the AKD context. For AMT we developed a new adaptive sampling scheme using the prior knowledge of Binomial Data to construct a Bayesian prior. We update this prior with newly collected data to estimate data uncertainty, using it to dynamically focus measurement efforts on the most informative regions of the data space. This adaptive process differs fundamentally from classical static sampling, thereby violating the i.i.d. assumption. We address this challenge with a new test statistic, for which we where able to derive the H_0 distribution despite the non-i.i.d. data. As such we can provide theoretical guarantees for type I error control while benefiting from the increase in power due to adaptive sampling. Our approach is particularly valuable in domains where data acquisition is expensive, such as materials science and medicine. Our empirical results demonstrate that AMT gains up to *485%* in test power, requiring *half* the data compared to classical approaches, while keeping the type I error below a required significance level.

Looking ahead, we believe expanding adaptive sampling to hypothesis tests beyond MT is a promising research direction. Specifically, we believe that Independence Testing (IT) can benefit even more from AKD, as one has finer control over the sampling location.

Overall, this chapter closes the circle by combining what we have demonstrated in previous chapters, namely: Incorporating knowledge about the DGP to quantify uncertainty; using the uncertainty for Bayesian data selection; solving a novel knowledge discovery task in the AKD setting, far more data efficiently than previously possible.

Part VII.

Conclusions

7. Outcome

This dissertation addresses fundamental topics in Adaptive Knowledge Discovery (AKD) to account for environments where data is not a cheap commodity but an expensive resource. We explore the AKD paradigm through four distinct lenses: Drifting Data-Generating Process (DGP) (Q1), Uncertainty Quantification (UQ) of Integrated Measurements (Q2), Domain Knowledge Integration (Q3), and, combining our learnings, an adaptive solution to the task of Multi-sample Testing (MT) (Q4).

These topics are fundamentally linked by the necessity of data efficiency. In modern science and industry, the assumption of readily available data is often a fallacy. Whether due to the non-stationarity of the process or the physical constraints of the measurement system, we must move beyond static data collection. The core philosophy of AKD, as established in this work, is to treat the interaction with the DGP as a dynamic, uncertainty-aware loop — shifting the paradigm from models that merely make static predictions to algorithms that, in a Socratic sense, "know what they do not know" and can actively inquire to bridge their own gaps in understanding.

In Part III, we tackled the challenge of concept drift (Q1). We introduced **Data Efficient Active Learning (DEAL)**, a framework for regression under a non-stationary DGP. Unlike traditional stream-based learning that assumes an infinite, low-cost data flow, DEAL explicitly models the stochastic nature of drift. By using a time-variant Gaussian Process to estimate increase in uncertainty, DEAL allows for model recalibration only when necessary. Our results demonstrated that one can maintain a user-defined accuracy while using up to 20 times fewer measurements than standard practices, effectively bridging the gap between active learning and drifting regression tasks.

In Part IV, we shifted focus to the structural constraints of measurements, specifically integrated data (Q2). We derived the **Brownian Integral Kernel (BIK)**, a novel analytical solution for quantifying uncertainty in cases where direct observation of a quantity is impossible due to aggregation — a common hurdle in smart grids and physical sensing. By modeling the inherent uncertainty of integrated Brownian motions, the BIK improves variance estimation by a factor of 2 and data synthesis by a factor of 10 over conventional kernels. This provides a mathematical foundation for AKD in scenarios involving unobservable quantities of interest.

In Part V, we investigated how domain knowledge can act as a catalyst for data efficiency (Q3). In the context of industrial manufacturing simulations, we proposed **Objective Alignment (OA)**. This method ensures that the training of a surrogate model is directly steered by the specific needs of the discovery task through gradient-weighted loss functions. We demonstrated that by anchoring models to physical constraints and task-specific goals, we significantly reduce the reliance on expensive high-fidelity simulations, outperforming state-of-the-art models by a significant margin.

Finally, in Part VI, we synthesized these building blocks to address the task of Multi-sample Testing (Q4). We introduced **Adaptive Multi-sample Testing (AMT)**, which leverages Bayesian uncertainty quantification to focus sampling on the most informative regions for distinguishing distributions. By deriving the null distribution for non-i.i.d. adaptive samples, we maintained theoretical guarantees for Type I error while achieving up to a 485% gain in test power. This closes the circle of the dissertation, proving that AKD methodology can improve even classic statistical tasks by making them adaptive and resource-aware.

Overall, this dissertation establishes AKD as a robust methodology for knowledge discovery in resource-constrained environments. Our contributions have been validated through real-world applications ranging from energy load forecasting, over financial markets, to textile manufacturing, and we have made our algorithms available as open-source tools to support the broader research community.

8. Future Work

The methodologies introduced in this work open several promising avenues for future research:

Hybrid Drift Modeling: Data Efficient Active Learning (DEAL) currently focuses on stochastic drift behavior, yet the DEAL framework is inherently flexible, allowing for the integration of domain-informed kernels that can encode specific physical priors or structural periodicities. A compelling research direction would be the integration of structural change points into this framework. By combining the statistical uncertainty of DEAL with physics-informed drift detection, we could achieve more precise recalibration in industrial settings where certain drifts are semi-predictable. Furthermore, exploring Variational Neural Networks as learned drift priors could allow the system to internalize complex, non-linear patterns of change, moving from reactive recalibration to proactive, uncertainty-aware forecasting of the evolution of a Data-Generating Process (DGP).

Deepening Uncertainty Quantification and Robustness: While the Brownian Integral Kernel (BIK) solved the problem for Brownian motions, many physical processes follow different stochastic dynamics (e.g., Ornstein-Uhlenbeck or Lévy processes). Generalizing integrated kernels by extending our analytical approach for these processes would broaden the applicability of Adaptive Knowledge Discovery (AKD) in fields like finance and epidemiology.

Further, building on the classification in Section 2.3, a vital next step is the precise disentanglement of aleatoric and epistemic uncertainty. Future AKD acquisition functions must distinguish between irreducible noise and reducible missing-data uncertainty to avoid wasting resources in high-noise regions. This challenge is especially present in many industrial scenarios, where measurement noise is heteroscedastic, meaning sensor precision fluctuates across the input space.

Furthermore, we must address input uncertainty. In practice, a selected measurement location x is often subject to mechanical tolerances, leading to observations at a 'noisy location' different from the intended one. Incorporating such uncertainty into Gaussian Process (GP) kernels will naturally lead to *Robust Optimization*, which is particularly critical for surrogate model-based optimization in real-world applications. Unlike deterministic simulations, real processes must account for input noise to identify 'flat' optima that remain stable despite unavoidable perturbations.

Human-in-the-Loop AKD: While most active learning literature assumes an 'expert labeler', the ultimate goal of AKD is a fully closed loop where labeling is performed directly and autonomously by the data generating process. However, a significant challenge remains in bridging the gap between an expert's qualitative intuition (the ability to 'know it when they see it') and the quantitative requirements of a model. Future research could investigate using initial manual heuristics to bootstrap domain knowledge integration, eventually transitioning to a state where these latent expert insights are encoded into the

Objective Alignment (OA) framework. This would allow AKD to move beyond simple labels, leveraging complex domain knowledge to steer autonomous DGP interactions without requiring continuous manual intervention.

Scaling Adaptive Multi-sample Testing (AMT) to High-Dimensional Spaces and Discovery of Structure: Our work on AMT focused primarily on Bernoulli-distributed data. Extending the AMT framework to high-dimensional, multivariate distributions remains an open challenge, requiring new task-aligned acquisition functions that can cope with the curse of dimensionality while maintaining statistical significance guarantees.

Beyond simple testing, this logic naturally extends to the field of *Causality Theory* and *Independence Testing*. Future research could leverage adaptive sampling to efficiently distinguish between mere correlation and true causation, using AKD to focus measurements on potential confounding variables. By applying the AMT methodology to tests for conditional independence in high-dimensional streams, we could move toward the autonomous discovery of dependency networks-dynamic, i.e., graphical representations of the underlying causal structure of a DGP.

Benchmarking and Evaluation of AKD across Discovery Tasks: Throughout this work, we have observed that evaluating AKD algorithms under true application conditions is often impractical. The core assumption of AKD – that observations are expensive – precludes the acquisition of the dense, large-scale datasets typically required for meaningful performance metrics. This challenge is further aggravated in the context of query synthesis, where the learner can request any point within a continuous input space, rendering standard discrete datasets insufficient. While we addressed this by utilizing synthetic generators, expert-informed models, and pre-trained surrogates, a significant gap remains. Specifically, for the aforementioned goal of autonomous discovery of dependency networks, the necessary evaluation data simply does not exist. Future research should focus on the development of standardized, 'cheap' digital twins of complex DGPs that maintain the stochastic and non-stationary characteristics of the real world. Establishing these benchmarks is the only way to move from isolated case studies toward a generalized, comparable assessment of adaptive knowledge discovery.

In summary, this work provides a methodological case for Adaptive Knowledge Discovery, demonstrating that the path toward more intelligent systems lies not in increasing model capacity alone, but in the rigorous quantification of uncertainty. By treating data acquisition as a strategic choice guided by 'what a model knows about what it does not know', we move beyond static prediction toward a closed-loop discovery process. Ultimately, anchoring these adaptive cycles in physical domain knowledge and task-aligned objectives allows us to enable scientific and industrial discovery even when resources are scarce, leaving room for – and quantifying – the *doubt* that is essential to the scientific spirit.

Appendix

Part I.

Additional Materials

A. Brownian Integral Kernel (BIK): A New Kernel for Modeling Integrated Brownian Motions

A.0.1. Proof of Brownian Integral Kernel (BIK) Correctness

Theorem A.1 (Positive-semidefinite BIK)**.** *The Brownian Integral Kernel is positive-semidefinite:*

$$\sum_{j=1}^{N}\sum_{i=1}^{N} a_i a_j k_{\mathcal{F}\mathcal{F}'}((s_i,e_i),(s_j,e_j)) \geq 0 \text{ with } N \text{ a arbitrary number of test points,}$$

for any $s_i, e_i, s_j, e_j \in \mathbb{T} | s_i < e_i, s_j < e_j$ *and for any* $a_i, a_j \in \mathbb{R}$.

Proof. Theorem A.1 directly follows from the derivation of the BIK as an integration of an existing kernel. The well-known Brownian kernel $k_{ff'}(t,t')$ is known to be positive-semidefinite [LRS13]. Any integral of a kernel is again a kernel [Abr97]. Therefore, the two times integral of $k_{ff'}(t,t')$, which results in the Brownian integral kernel, is positive-semidefinite. □

A.0.2. How Kernel Integration yields an Integrated Process

One may ask how integrating a kernel is equivalent to integrating over a random process defined by the kernel. This relation directly follows from the following theorem:

Theorem A.2 (Kernel Integration yields an Integrated Process)**.** *The BIK* $k_{\mathcal{F}\mathcal{F}'}((s,e),(s',e'))$ *yields the covariance between two integral measurements* $\mathcal{B}(s,e)$ *and* $\mathcal{B}(s',e')$ *taken from a Brownian motion* $b(t)$*:*

$$\mathrm{Cov}[\mathcal{B}(s,e),\mathcal{B}(s',e'))] = \mathrm{Cov}[\int_s^e b(t)\,\mathrm{d}t, \int_{s'}^{e'} b(t')\,\mathrm{d}t']$$

.

Theorem A.2 implies that the double integral over the covariance $k_{ff'}(t,t')$ of a Brownian motion $b(t)$ is equal to the covariance between two integrated measurement regions $\mathcal{B}(s,e)$ and $\mathcal{B}(s',e')$, i.e., equal to the covariance of an integrated Brownian motion:

$$\mathrm{Cov}[\int_s^e b(t)\,\mathrm{d}t, \int_{s'}^{e'} b(t')\,\mathrm{d}t'] = \int_{s'}^{e'}\int_s^e \mathrm{Cov}[b(t),b(t')]\,\mathrm{d}t'\,\mathrm{d}t. \tag{A.1}$$

[FTM19] features a respective proof for this relation which holds for integrals of any kernel, by decomposing the covariance as $\mathrm{Cov}[X, Y] = \mathrm{E}[XY] - \mathrm{E}[X]\mathrm{E}[Y]$. An additional proof can be obtained by interchanging the sequence of integration (Fubini) [RW06], [Abr97]. This also guarantees that the resulting stochastic process is again a Gaussian process.

B. AMT: Data-Efficient Adaptive Multi-sample Testing for Binomial Data

B.1. Proof of Theorem 6.1

To prove Theorem 6.1, we introduce a simplified version of the algorithm in B.1.1, which splits the observations of coins into observations used for the selection decision, and independent sampled observations for which the iid assumption holds. We prove its distributional properties in Proposition B.1. We then expand the simplified algorithm to an 'refined' algorithm in Section B.1.2, which is equivalent to our full algorithm from the main paper. Finally, we expand the proof from the simplified algorithm to this full 'refined' algorithm. This results in the proof of Theorem B.1 in this appendix, which is equivalent to Theorem 6.1 in the main paper (with adapted notation).

Notation

For this section, we transition from the global Adaptive Knowledge Discovery (AKD) framework to the standard notation prevalent in statistical test theory and Multi-armed Bandit (MAB) literature. This departure ensures that the formal derivation remains accessible and rigorously comparable to established results in these specialized fields, allowing experts to focus exclusively on the distributional properties and convergence proofs presented herein.

The variables and samples used in the algorithm are as follows:

- **Coins and Probabilities:**
  - **Coin 1:** A coin with an unknown probability of heads p_1. Flips from this coin are Bernoulli random variables, $X_C, X_{R1} \sim \text{Bernoulli}(p_1)$.
  - **Coin 2:** A coin with an unknown probability of heads p_{R2}. Flips from this coin are Bernoulli random variables, $X_{R2} \sim \text{Bernoulli}(p_{R2})$.
  - **Reference Probability p_{ref}:** A predefined constant probability value used in the decision rule.
- **Samples:**
  - **Primary Sample (S_C):** A sample collected from Coin 1. It starts empty ($S_C^{(0)} = \emptyset$) and is populated based on a decision rule.
  - **Reference Sample 1 (S_{R1}):** A sample collected from Coin 1, used to estimate p_1.

  - **Reference Sample 2 (S_{R2}):** A sample collected from Coin 2, used to estimate p_{R2}.

- **Iteration-Specific Variables (at iteration i):**
  - **Samples:** $S_C^{(i)}$, $S_{R1}^{(i)}$, $S_{R2}^{(i)}$ denote the state of the samples at the end of iteration i.
  - **Flip Counts:** $N_C^{(i)}$, $N_{R1}^{(i)}$, $N_{R2}^{(i)}$ are the total number of flips in the respective samples.
  - **Headcounts:** $H_C^{(i)}$, $H_{R1}^{(i)}$, $H_{R2}^{(i)}$ are the total number of heads in the respective samples.
  - **Empirical Proportions:** $\hat{p}_{R1}^{(i)} = \frac{H_{R1}^{(i)}}{N_{R1}^{(i)}}$ and $\hat{p}_{R2}^{(i)} = \frac{H_{R2}^{(i)}}{N_{R2}^{(i)}}$ are the observed head proportions.
  - **Decision Variable (q_i):** A binary variable where $q_i = 1$ means a new flip is added to $S_C^{(i-1)}$ to form $S_C^{(i)}$, and $q_i = 0$ means it is discarded.
  - **Total Iterations (I):** The total number of iterations the algorithm runs.

B.1.1. Simplified Adaptive Sampling Algorithm

The goal of Algorithm B.1 is to adaptively build a primary sample S_C using flips from Coin 1. The decision to add a flip is based on comparing the empirical head proportions of two growing, independent reference samples, S_{R1} (from Coin 1) and S_{R2} (from Coin 2).

Algorithm Steps

1. **Initialization (Iteration $i = 0$):**
   - Initialize the primary sample as empty: $S_C^{(0)} = \emptyset$.
   - Initialize the reference sample from Coin 1: $S_{R1}^{(0)}$ by taking $N_R^{(0)}$ initial independent flips from Coin 1.
   - Initialize the reference sample from Coin 2: $S_{R2}^{(0)}$ by taking $N_R^{(0)}$ initial independent flips from Coin 2.
2. **Iterative Process (For $i = 1, \ldots, I$):** At each iteration i, perform the following steps:
   - **Update Reference Samples:**
     - Add one new independent flip from Coin 1 to $S_{R1}^{(i-1)}$ to form $S_{R1}^{(i)}$.
     - Add one new independent flip from Coin 2 to $S_{R2}^{(i-1)}$ to form $S_{R2}^{(i)}$.
     - The total size of each reference sample is now $N_R^{(i)} = N_R^{(0)} + i$.
   - **Calculate Empirical Proportions:**
     - $\hat{p}_{R1}^{(i)} = \frac{H_{R1}^{(i)}}{N_R^{(i)}}$.

- $\hat{p}_{R2}^{(i)} = \frac{H_{R2}^{(i)}}{N_R^{(i)}}$.

- **Perform New Coin 1 Flip for S_C Consideration:** Flip Coin 1 once, let the result be $X_C^{(i)}$.
- **Decision Rule (q_i):**
  - If $|\hat{p}_{R1}^{(i)} - p_{ref}|^2 > |\hat{p}_{R2}^{(i)} - p_{ref}|^2$: Assign $X_C^{(i)}$ to S_C. (Set $q_i = 1$. Update $S_C^{(i)} := S_C^{(i-1)} \cup \{X_C^{(i)}\}$.)
  - Else: Discard $X_C^{(i)}$. (Set $q_i = 0$. $S_C^{(i)} := S_C^{(i-1)}$.)

3. **Final Output:** The final primary sample $S_C^{(I)}$ (or its headcount $H_C^{(I)}$), combined with $S_{R1}^{(i)}$ resulting in headcount $H_{Total}^{(I)} = H_{R1}^{(I)} + H_C^{(I)}$.

Algorithm B.1 Simplified Adaptive Sampling

procedure SimplifiedSampling($N_R^{(0)}$, I, p_{ref}, X_{R1}, X_{R2}, X_C) ▷ *Initialization at i=0*
  $S_C^{(0)} := \emptyset$
  $S_{R1}^{(0)} \underset{N_R^{(0)}}{\leftarrow} X_{R1}$
  $S_{R2}^{(0)} \underset{N_R^{(0)}}{\leftarrow} X_{R2}$
  for $i \in \{1, \ldots, I\}$ **do** ▷ *Update reference samples*
    $X_{R1}^{(i)} \leftarrow X_{R1}$
    $S_{R1}^{(i)} := S_{R1}^{(i-1)} \cup \{X_{R1}^{(i)}\}$
    $X_{R2}^{(i)} \leftarrow X_{R2}$
    $S_{R2}^{(i)} := S_{R2}^{(i-1)} \cup \{X_{R2}^{(i)}\}$
    ▷ *Calculate empirical proportions for the decision*
    $\hat{p}_{R1}^{(i)} := |\{h \in S_{R1}^{(i)} : h = 1\}| / |S_{R1}^{(i)}|$
    $\hat{p}_{R2}^{(i)} := |\{h \in S_{R2}^{(i)} : h = 1\}| / |S_{R2}^{(i)}|$
    ▷ *Sample a candidate flip and apply decision rule*
    $X_C^{(i)} \leftarrow X_C$
    if $|\hat{p}_{R1}^{(i)} - p_{ref}|^2 > |\hat{p}_{R2}^{(i)} - p_{ref}|^2$ **then**
      $S_C^{(i)} := S_C^{(i-1)} \cup \{X_C^{(i)}\}$
    else
      $S_C^{(i)} := S_C^{(i-1)}$
    ▷ *Calculate final headcounts from Coin 1*
  $H_{R1}^{(I)} := |\{h \in S_{R1}^{(I)} : h = 1\}|$
  $H_C^{(I)} := |\{h \in S_C^{(I)} : h = 1\}|$
  return $H_{R1}^{(I)} + H_C^{(I)}$

Distributional Properties of the Simplified Algorithm

The sampling process of the simplified algorithm results in the following distributions under H_0 for the accumulated headcounts $H_{Total}^{(I)}$. The number of heads in the primary sample, $H_C^{(I)}$, conditioned on the total number of accepted flips $N_C^{(I)} = \sum_{j=1}^{I} q_j$, follows a Binomial distribution, as each accepted flip is an independent Bernoulli trial from Coin 1.

$$H_C^{(I)} | N_C^{(I)} \sim \text{Bin}(N_C^{(I)}, p_1) \tag{B.1}$$

Furthermore, since both the reference sample S_{R1} and the primary sample S_C draw from the same source (Coin 1), the total number of heads from Coin 1 across both samples also follows a Binomial distribution.

$$H_{Total}^{(I)} = H_{R1}^{(I)} + H_C^{(I)} \sim \text{Bin}(N_{R1}^{(I)} + N_C^{(I)}, p_1) \tag{B.2}$$

The probability mass function of $H_{Total}^{(I)}$ thereby is:

$$P(H_{Total}^{(I)} = k) = \binom{N_{R1}^{(I)} + N_C^{(I)}}{k} p_1^k (1-p_1)^{N_{R1}^{(I)} + N_C^{(I)} - k} \quad \text{for } k \in \{0, 1, \ldots, N_{R1}^{(I)} + N_C^{(I)}\} \tag{B.3}$$

The following proposition suggests that the total number of heads $H_{Total}^{(i)} = H_{R1}^{(i)} + H_C^{(i)}$ collected from Coin 1, encompassing both the reference sample $\text{S}_{R1}^{(i)}$ and the primary sample $\text{S}_C^{(i)}$, consistently follows a Binomial distribution with the true probability p_1.

Proposition B.1 (Head count as the sum of individual counts)**.** *Assume the reference sample $\text{S}_{R1}^{(i)}$ and the primary sample $\text{S}_C^{(i)}$ are collected using the Simplified Adaptive Algorithm B.1. Then, at any iteration i, $H_{Total}^{(i)} = H_{R1}^{(i)} + H_C^{(i)} \sim Bin(N_{R1}^{(i)} + N_C^{(i)}, p_1)$.*

A formal proof for this combined distribution is provided in the subsequent section.

Proof of Proposition B.1: Distribution of Heads in Combined Samples S_{R1} and S_C

Proof. We conduct a proof by induction.

Base Case ($i = 0$): $\text{S}_C^{(0)}$ is empty: $N_C^{(0)} = 0, H_C^{(0)} = 0$. $\text{S}_{R1}^{(0)}$ has $N_R^{(0)}$ flips from Coin 1: $H_{R1}^{(0)} \sim \text{Bin}(N_R^{(0)}, p_1)$. Combined total: $H_{R1}^{(0)} + H_C^{(0)} = H_{R1}^{(0)} \sim \text{Bin}(N_R^{(0)} + 0, p_1)$. The base case holds.

Inductive Step: Assume the hypothesis holds for iteration $i-1$: $H_{R1}^{(i-1)} + H_C^{(i-1)} \sim \text{Bin}(N_{R1}^{(i-1)} + N_C^{(i-1)}, p_1)$.

Consider iteration i:

1. **New Flips for Reference Samples:** A new flip $X_{R1}^{(i)}$ from Coin 1 is added to $\text{S}_{R1}^{(i-1)}$. This flip is a Bernoulli trial $X_{R1}^{(i)} \sim \text{Bernoulli}(p_1)$, where $X_{R1}^{(i)} = 1$ for a head. So, $H_{R1}^{(i)} = H_{R1}^{(i-1)} + X_{R1}^{(i)}$.

2. **Assignment Decision for S_C:** A new flip $X_C(i)$ from Coin 1 is performed. This is also a Bernoulli trial $X_C^{(i)} \sim \text{Bernoulli}(p_1)$. Based on the decision rule, it is either assigned to S_C ($q_i = 1$) or it is discarded ($q_i = 0$).
   - If $q_i = 1$, $H_C^{(i)} := H_C^{(i-1)} + X_C^{(i)}$.
   - If $q_i = 0$, $H_C^{(i)} := H_C^{(i-1)}$.

3. **Total Heads from Coin 1:** The total number of heads from Coin 1 is $H_{Total}^{(i)} = H_{R1}^{(i)} + H_C^{(i)}$. This sum consists of:
   - The $(N_{R1}^{(i-1)} + N_C^{(i-1)})$ flips already accumulated, which, by hypothesis, form a $\text{Bin}(N_{R1}^{(i-1)} + N_C^{(i-1)}, p_1)$ distribution.
   - The new flip $X_{R1}^{(i)} \sim \text{Bin}(1, p_1)$.
   - The new flip $X_C^{(i)} \sim \text{Bin}(1, p_1)$, which contributes to $H_C^{(i)}$ only if $q_i = 1$. If $q_i = 0$, its contribution is effectively 0.

4. **Application of Sum of Binomials:** All these individual flips (whether part of S_{R1}, S_C, or the new additions) originate independently of Coin 1 with success probability p_1. Since the sum of independent Binomial (or Bernoulli) random variables with the same p is also Binomial, the total sum of successes will be:

$$H_{Total}^{(i)} \sim \text{Bin}((N_{R1}^{(i-1)} + N_C^{(i-1)}) + 1 + q_i, p_1) \tag{B.4}$$

   Recognizing that $N_{R1}^{(i)} = N_{R1}^{(i-1)} + 1$ and $N_C^{(i)} = N_C^{(i-1)} + q_i$, we get:

$$H_{Total}^{(i)} \sim \text{Bin}(N_{R1}^{(i)} + N_C^{(i)}, p_1) \tag{B.5}$$

Thus we have shown that $H_{Total}^{(i)} \sim \text{Bin}(N_{R1}^{(i)} + N_C^{(i)}, p_1)$ as proposed. □

Conclusion

The inductive step holds, which confirms that the total count of heads originating from Coin 1 (combined from S_{R1} and S_C) maintains its Binomial distribution with parameter p_1 throughout the iterative process, despite the dynamic assignment rule. This property is crucial for validating the overall methodology.

B.1.2. Refined Adaptive Sampling Algorithm

This section details a refined version of the simplified adaptive sampling algorithm. The refined algorithm is equivalent to our proposed method presented in the main paper. That is, if we only investigate the distribution of one single coin out of the N given coins, and combine the samples from the other coins to form one larger reference sample, we arrive at this algorithm. By extending the proof of the simplified algorithm, this refined algorithm can bridge the gap to the algorithm in the main paper, thereby providing us with an inductive proof for the distribution of the refined Coin 1 reference sample S_{R1} given in Theorem B.1. Doing so directly proves the equivalent Theorem 6.1 from our main paper.

Algorithm Steps

The core objective remains to adaptively collect a primary sample S_C from Coin 1, while using reference samples S_{R1} and S_{R2} to guide the collection process. This refined algorithm in Algorithm B.2 introduces a dynamic update rule for S_{R1}, making it dependent on S_C.

1. **Initialization (Iteration $i = 0$):**
   - Initialize the primary sample $\mathrm{S}_C^{(0)} = \emptyset$.
   - Initialize the initial component of the Coin 1 reference sample: $\mathrm{S}_{R1}^{(0)}$ by taking $N_{R1}^{(0)} = N_R^{(0)}$ initial independent flips from Coin 1.
   - Initialize the reference sample from Coin 2: $\mathrm{S}_{R2}^{(0)}$ by taking $N_{R2}^{(0)} = N_R^{(0)}$ initial independent flips from Coin 2.
2. **Iterative Process (For $i = 1, \ldots, I$):** At each iteration i, perform the following steps:
   - **Update Reference Sample S_{R2}:** Add one new independent flip from Coin 2 to $\mathrm{S}_{R2}^{(i-1)}$ to form $\mathrm{S}_{R2}^{(i)}$.
   - **Define Current Reference Sample $\mathrm{S}_{R1}^{(i)}$ for Decision:** The Coin 1 reference sample $\mathrm{S}_{R1}^{(i)}$ used for the decision in iteration i is defined as the combination of the initial sample $\mathrm{S}_{R1}^{(0)}$ and all Coin 1 flips currently in $\mathrm{S}_C^{(i-1)}$. That is, $\mathrm{S}_{R1}^{(i)} := \mathrm{S}_{R1}^{(0)} \cup \mathrm{S}_C^{(i-1)}$. *Note: As S_C grows with accepted flips, S_{R1} (the reference for decision-making) also grows to incorporate all previously accepted Coin 1 data.*
   - **Calculate Empirical Proportions:**
     - $\hat{p}_{R1}^{(i)} = \frac{H_{R1}^{(i)}}{N_{R1}^{(i)}}$ (where $H_{R1}^{(i)}$ and $N_{R1}^{(i)}$ are the headcount and total flips in $\mathrm{S}_{R1}^{(i)}$).
     - $\hat{p}_{R2}^{(i)} = \frac{H_{R2}^{(i)}}{N_{R2}^{(i)}}$.
   - **Perform New Coin 1 Flip for S_C Consideration:** Flip Coin 1 once, let the result be $X_C^{(i)}$.
   - **Decision Rule (q_i):**

- If $|\hat{p}_{R1}^{(i)} - p_{ref}|^2 \leq |\hat{p}_{R2}^{(i)} - p_{ref}|^2$: Assign $X_C^{(i)}$ to S_C. Set $q_i = 1$. Update $S_C^{(i)} := S_C^{(i-1)} \cup \{X_C^{(i)}\}$.
- Else ($|\hat{p}_{R1}^{(i)} - p_{ref}|^2 > |\hat{p}_{R2}^{(i)} - p_{ref}|^2$): Discard $X_C^{(i)}$. Set $q_i = 0$. $S_C^{(i)} := S_C^{(i-1)}$.

3. **Final Output:** The final primary sample $S_C^{(I)}$ (or its headcount $H_C^{(I)}$), combined with $S_{R1}^{(i)}$ resulting in headcount $H_{Total}^{(I)} = H_{R1}^{(I)} + H_C^{(I)}$.

This refined algorithm dynamically adjusts the Coin 1 reference sample (S_{R1}) to include all available Coin 1 data, aiming for faster convergence of the reference estimate.

Algorithm B.2 Refined Adaptive Sampling

procedure RefinedSampling($N_R^{(0)}$, I, p_{ref}, X_{R1}, X_{R2}, X_C) ▷ *Initialization at i=0*
  $S_C^{(0)} := \emptyset$
  $S_{R1}^{(0)} \underset{N_R^{(0)}}{\leftarrow} X_{R1}$ ▷ $S_{R1}^{(0)}$ *is fixed*
  $S_{R2}^{(0)} \underset{N_R^{(0)}}{\leftarrow} X_{R2}$
  for $i \in \{1, \ldots, I\}$ **do** ▷ *Update* S_R2 *sample*
    $X_{R2}^{(i)} \leftarrow X_{R2}$
    $S_{R2}^{(i)} := S_{R2}^{(i-1)} \cup \{X_{R2}^{(i)}\}$
    ▷ *Define the Coin 1 reference sample for this iteration's decision*
    $S_{R1}^{(i)} := S_{R1}^{(0)} \cup S_C^{(i-1)}$
    ▷ *Calculate empirical proportions for the decision*
    $\hat{p}_{R1}^{(i)} := |\{h \in S_{R1}^{(i)} : h = 1\}| / |S_{R1}^{(i)}|$
    $\hat{p}_{R2}^{(i)} := |\{h \in S_{R2}^{(i)} : h = 1\}| / |S_{R2}^{(i)}|$
    ▷ *Sample a candidate flip and apply decision rule*
    $X_C^{(i)} \leftarrow X_C$
    if $|\hat{p}_{R1}^{(i)} - p_{ref}|^2 \leq |\hat{p}_{R2}^{(i)} - p_{ref}|^2$ **then**
      $S_C^{(i)} := S_C^{(i-1)} \cup \{X_C^{(i)}\}$
    else
      $S_C^{(i)} := S_C^{(i-1)}$
  ▷ *Calculate final headcounts from Coin 1*
  $H_{R1}^{(I)} := |\{h \in S_{R1}^{(0)} : h = 1\}|$ ▷ *Heads from the initial ref. sample*
  $H_C^{(I)} := |\{h \in S_C^{(I)} : h = 1\}|$ ▷ *Heads from the final primary sample*
  return $H_{R1}^{(I)} + H_C^{(I)}$

The following theorem states that the **Coin 1 reference sample** $S_{R1}^{(i)}$, used for decision-making in each iteration, always represents a valid Binomial distribution with the true probability p_1.

Theorem B.1 (Binomial distributed headcount)**.** *Let $S_{R1}^{(i)}$ be the Coin 1 reference sample used in iteration i, defined as the combination of the initial sample $S_{R1}^{(0)}$ and all flips collected in $S_C^{(i-1)}$. Let $N_{R1}^{(i)}$ be the total number of flips in $S_{R1}^{(i)}$ (i.e., $N_{R1}^{(i)} := N_{R1}^{(0)} + N_C^{(i-1)}$). Then, at*

any iteration $i \geq 1$, the headcount $H_{R1}^{(i)}$ from sample $\mathrm{S}_{R1}^{(i)}$ follows a Binomial distribution: $H_{R1}^{(i)} \sim Bin(N_{R1}^{(i)}, p_1)$.

Proof of Theorem B.1: Distribution of Reference Sample S_{R1}

Proof. We conduct a proof by induction. We wish to show that at any iteration $i \geq 1$, $H_{R1}^{(i)} \sim \mathrm{Bin}(N_{R1}^{(i)}, p_1)$,
where $N_{R1}^{(i)} := N_{R1}^{(0)} + N_C^{(i-1)}$.

Base Case ($i = 1$): According to the algorithm, for $i = 1$, $\mathrm{S}_{R1}^{(1)} := \mathrm{S}_{R1}^{(0)} \cup \mathrm{S}_C^{(0)}$. Since $\mathrm{S}_C^{(0)}$ is empty ($N_C^{(0)} = 0, H_C^{(0)} = 0$), $\mathrm{S}_{R1}^{(1)} := \mathrm{S}_{R1}^{(0)}$.
$N_{R1}^{(1)} := N_{R1}^{(0)} + N_C^{(0)} = N_{R1}^{(0)} + 0 = N_{R1}^{(0)}$. $H_{R1}^{(1)} = H_{R1}^{(0)}$.
By initialization, $H_{R1}^{(0)} \sim \mathrm{Bin}(N_{R1}^{(0)}, p_1)$. Thus, $H_{R1}^{(1)} \sim \mathrm{Bin}(N_{R1}^{(1)}, p_1)$.
The base case holds.

Inductive Step: Assume the hypothesis holds for iteration $i - 1$ (where $i - 1 \geq 1$): $H_{R1}^{(i-1)} \sim \mathrm{Bin}(N_{R1}^{(i-1)}, p_1)$. This means $H_{R1}^{(i-1)}$ is the headcount from $\mathrm{S}_{R1}^{(0)} \cup \mathrm{S}_C^{(i-2)}$, and $N_{R1}^{(i-1)} := N_{R1}^{(0)} + N_C^{(i-2)}$.
Now consider iteration i:

1. **Definition of $\mathrm{S}_{R1}^{(i)}$:** According to the algorithm, $\mathrm{S}_{R1}^{(i)} := \mathrm{S}_{R1}^{(0)} \cup \mathrm{S}_C^{(i-1)}$.

2. **Relationship to Previous State:** We know $\mathrm{S}_C^{(i-1)}$ is formed from $\mathrm{S}_C^{(i-2)}$ by potentially including the flip $X_C^{(i-1)}$ based on the decision q_{i-1}.
   - If $q_{i-1} = 1$, then $\mathrm{S}_C^{(i-1)} := \mathrm{S}_C^{(i-2)} \cup \{X_C^{(i-1)}\}$.
   - If $q_{i-1} = 0$, then $\mathrm{S}_C^{(i-1)} := \mathrm{S}_C^{(i-2)}$.

3. **Headcount Update:** The headcount for $\mathrm{S}_{R1}^{(i)}$ is $H_{R1}^{(i)} := H_{R1}^{(0)} + H_C^{(i-1)}$. We can express $H_C^{(i-1)}$ in terms of $H_C^{(i-2)}$ and $X_C^{(i-1)}$:

$$H_C^{(i-1)} := H_C^{(i-2)} + q_{i-1} X_C^{(i-1)} \tag{B.6}$$

   (where $q_{i-1} X_C^{(i-1)}$ is $X_C^{(i-1)}$ if assigned, 0 if discarded). Substituting this into the expression for $H_{R1}^{(i)}$:

$$H_{R1}^{(i)} := H_{R1}^{(0)} + H_C^{(i-2)} + q_{i-1} X_C^{(i-1)} \tag{B.7}$$

   Recognizing that $H_{R1}^{(i-1)} := H_{R1}^{(0)} + H_C^{(i-2)}$ (by definition of $\mathrm{S}_{R1}^{(i-1)}$), we get:

$$H_{R1}^{(i)} := H_{R1}^{(i-1)} + q_{i-1} X_C^{(i-1)} \tag{B.8}$$

4. **Application of Sum of Binomials Proof:** By the inductive hypothesis, $H_{R1}^{(i-1)} \sim \mathrm{Bin}(N_{R1}^{(i-1)}, p_1)$. The flip $X_C^{(i-1)}$ is an independent Bernoulli trial from Coin 1 $X_C^{(i-1)} \sim$

Bernoulli(p_1). The term $q_{i-1}X_C^{(i-1)}$ represents either one Bernoulli trial (if $q_{i-1} = 1$) or zero trials (if $q_{i-1} = 0$). In either case, it adds q_{i-1} independent Bernoulli trials from Coin 1 to the sum. Since $H_{R1}^{(i)}$ is the sum of $H_{R1}^{(i-1)}$ and q_{i-1} new independent Bernoulli trials, and all these trials originate from Coin 1 with the same probability p_1:

$$H_{R1}^{(i)} \sim \text{Bin}(N_{R1}^{(i-1)} + q_{i-1}, p_1) \tag{B.9}$$

5. **Verifying the Parameter $N_{R1}^{(i)}$:** From the algorithm's definition of $N_{R1}^{(i)}$, we have $N_{R1}^{(i)} := N_{R1}^{(0)} + N_C^{(i-1)}$. Also, $N_C^{(i-1)} := N_C^{(i-2)} + q_{i-1}$. So, $N_{R1}^{(i)} := N_{R1}^{(0)} + N_C^{(i-2)} + q_{i-1}$. Since $N_{R1}^{(i-1)} := N_{R1}^{(0)} + N_C^{(i-2)}$, we can substitute this:

$$N_{R1}^{(i)} := N_{R1}^{(i-1)} + q_{i-1}, \tag{B.10}$$

which now matches the parameter derived from the sum of Binomials. Therefore, $H_{R1}^{(i)} \sim \text{Bin}(N_{R1}^{(i)}, p_1)$.

□

Concluding Remark: Generalizability of the Method

The proofs presented demonstrate the fundamental properties of the collected Coin 1 data. It is crucial to emphasize that the validity of these distributional properties – specifically, that the total number of heads from Coin 1 (encompassing $S_{R1}^{(0)}$ and $S_C^{(i)}$) always follows a Binomial distribution $\text{Bin}(N_{R1}^{(0)} + N_C^{(i)}, p_1)$ – holds for **any adaptive selection strategy**.

The only prerequisites for this robustness are:

1. The selection strategy (determining q_i) must **solely be based on observations from previous iterations** (i.e., data collected in $S_{R1}^{(i)}$ and $S_{R2}^{(i)}$ before the current flip $X_C^{(i)}$ is made). It must not depend on the outcome of the very flip $X_C^{(i)}$ that is being decided upon.

2. The outcome of the selection decision (q_i) must be **observable**, meaning we must know whether a given Coin 1 flip was assigned to S_C or discarded. Precomputing the distribution for arbitrary q_i is thus not possible.

As long as these conditions are met, the adaptive nature of the algorithm, while influencing the *number of samples* in S_C, does not bias the statistical properties of the collected Coin 1 data in relation to its true underlying probability p_1. This generalizability underscores the strength and broad applicability of this adaptive sampling approach.

B.2. Additional Experimental Results

B.2.1. Power in Different Scenarios

This section provides additional power plots under different scenarios. These plots all exhibit behavior that is very similar to that shown in the main paper, and lead to the same conclusions: (1) The main gain in power comes from using adaptive sampling instead of using classic non-adaptive sampling. (2) The used sampling strategy has the largest impact on the gained power. (3) The used test statistics have a negligible impact on the power. As we can draw the same conclusions from each plot, we will only provide the plots and refer to the discussion in the main paper for further details.

Figure B.1 shows the power of different tests using our beta sampling vs the non-adaptive equal sampling.

Figure B.2 shows the power of a test based on our Beta-Binomial statistic for varying numbers of coins.

Figure B.3 shows the power of different tests using our beta sampling vs the non-adaptive equal sampling, now for $\delta p = 0.15$. We can see significant gains in power early on (with fewer data points), reaching a power of over 80% with only $\frac{1}{4}$ of the data. Furthermore, we observe that under non-adaptive equal sampling, there are significant differences in the power of the test statistics, where our test statistic outperforms the χ^2 and Kruskal–Wallis Test (KW), even with equal sampling.

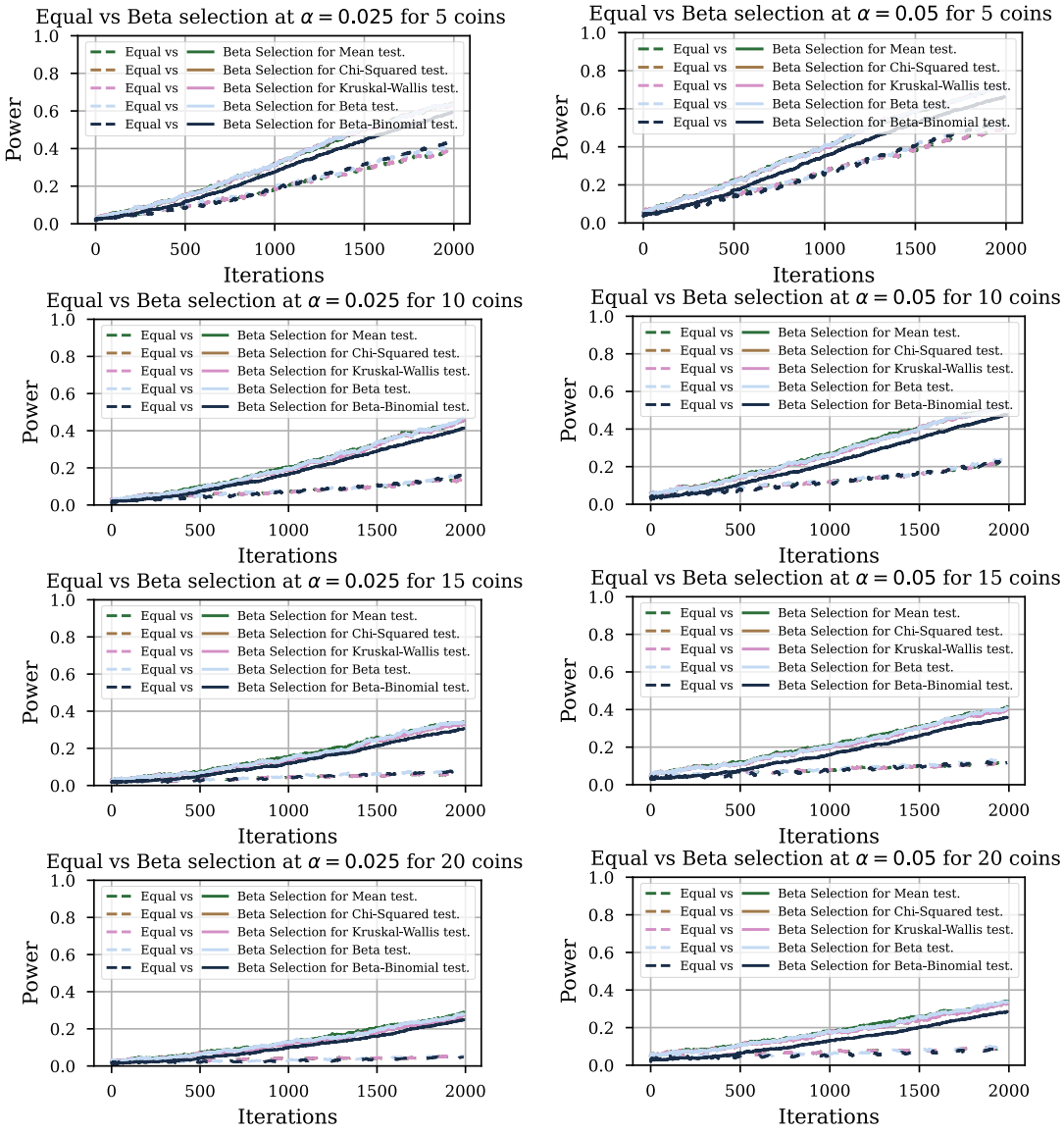


Figure B.1.: Power of tests using different test statistics with non-adaptive equal vs adaptive beta sampling, $\delta p = 0.05$ – adaptive sampling always results in higher power, for any number of coins, and any number of sampling iterations.

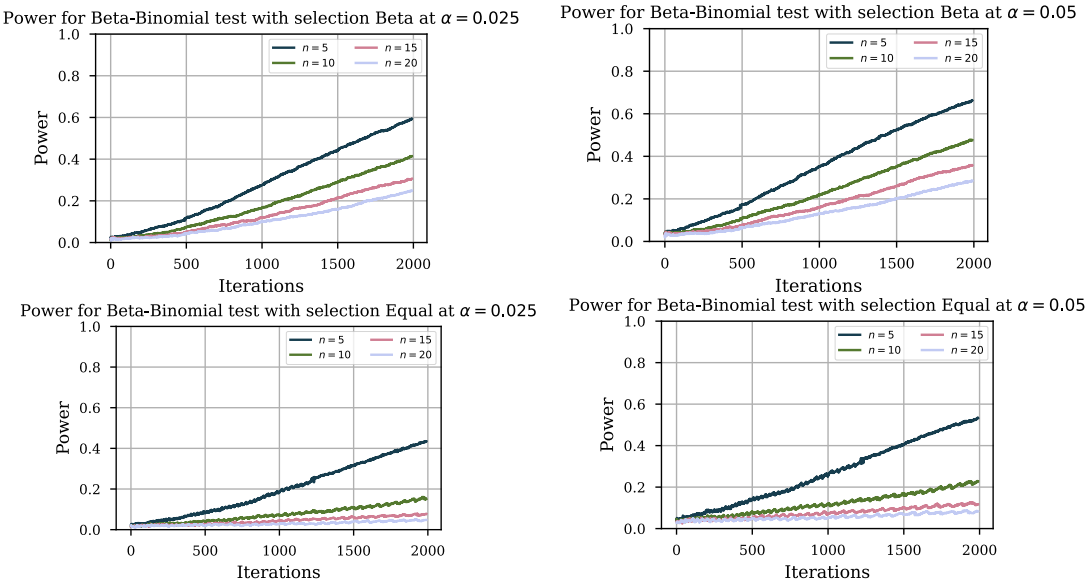


Figure B.2.: Power of the test using our Beta-Binomial statistic for different numbers of coins. Top: adaptive beta sampling; Bottom: non-adaptive equal sampling, $\delta p = 0.05$. While the power naturally decreases for larger numbers of coins, the gain in power due to adaptive sampling is greater when the number of coins is higher. This behavior is caused by adaptive sampling having more freedom to concentrate on important samples, thereby achieving a greater difference compared to non-adaptive sampling.

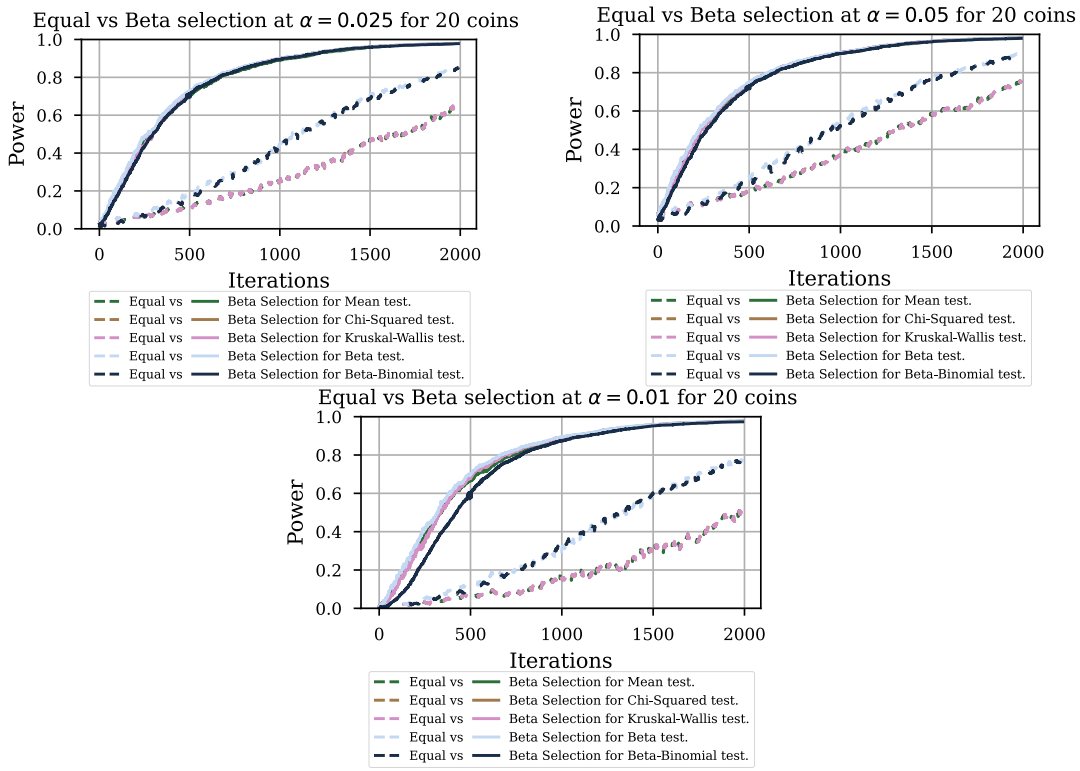


Figure B.3.: Power of tests using different test statistics with non-adaptive equal vs adaptive beta sampling, $\delta p = 0.15$ – adaptive sampling results in significantly higher power, especially early on with fewer data points and at any required significance level.

B.2.2. Type I Error in Different Scenarios

In this section, we will present the results of our simulation study, verifying the type I error control for different iteration limits I and different significance levels α for our new test statistic and other statistics using re-simulation permutation testing.

To obtain a reference, in Figure B.4 we will start by comparing tests based on our test statistic to tests based on the baseline test statistics using the non-adaptive equal sampling. We can observe in Figure B.4 that all test statistics conform to the significance level, independently of sample iteration or number of coins.

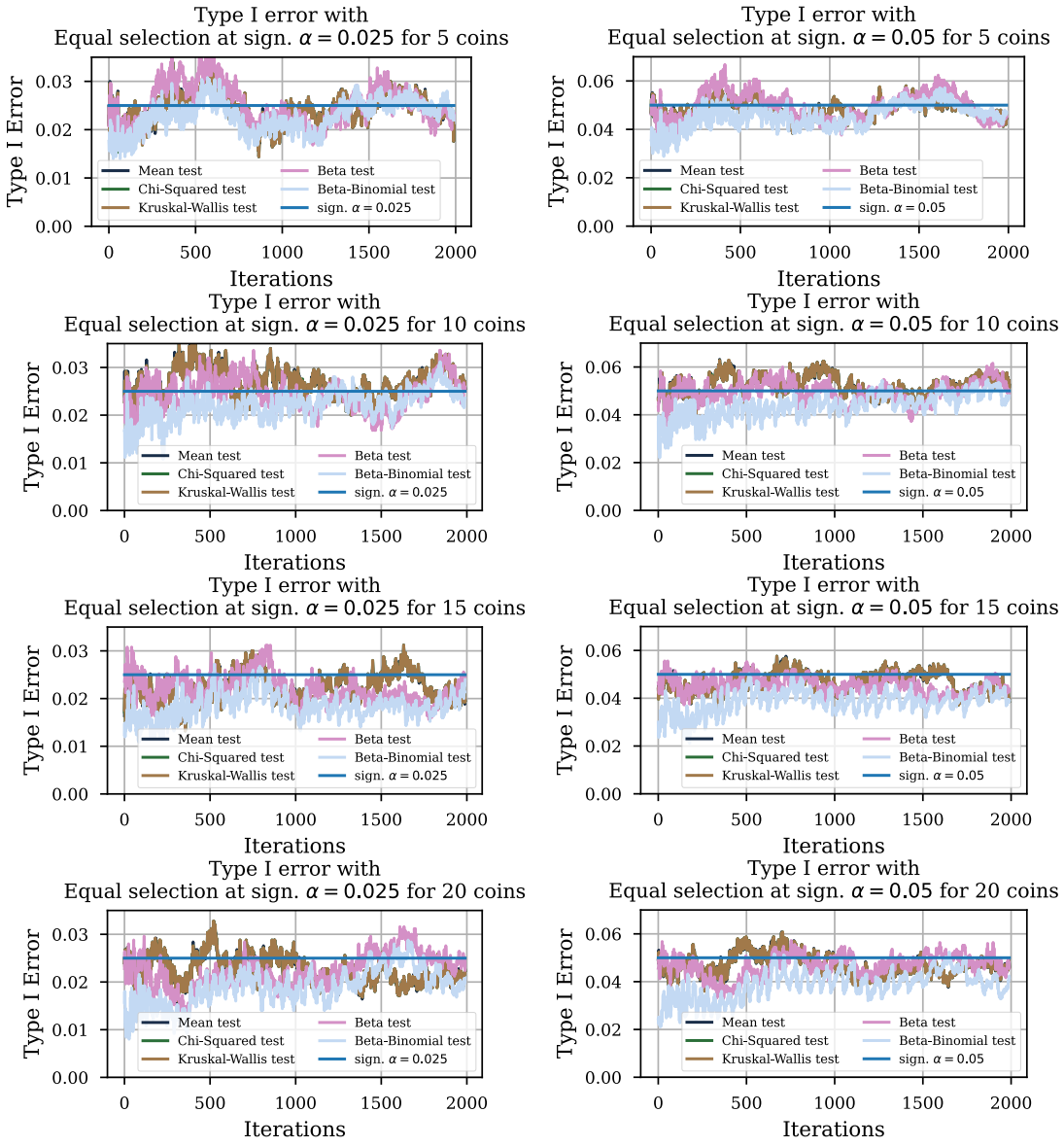


Figure B.4.: Type I error under non-adaptive equal sampling: we observe that all tests meet the required significance level.

In Figure B.5 we compare the same tests, but now using our adaptive Beta sampling. We observe that our test statistic always guarantees the required type I error. It is, however,

somewhat conservative, a known property when using the Bonferroni correction. This behavior of our test statistic is consistent across both iteration count and the number of coins. On the other hand, permutation-based tests are in some cases overly optimistic, indicating that even tests based on re-simulating the adaptive process are sensitive to non-iid data.

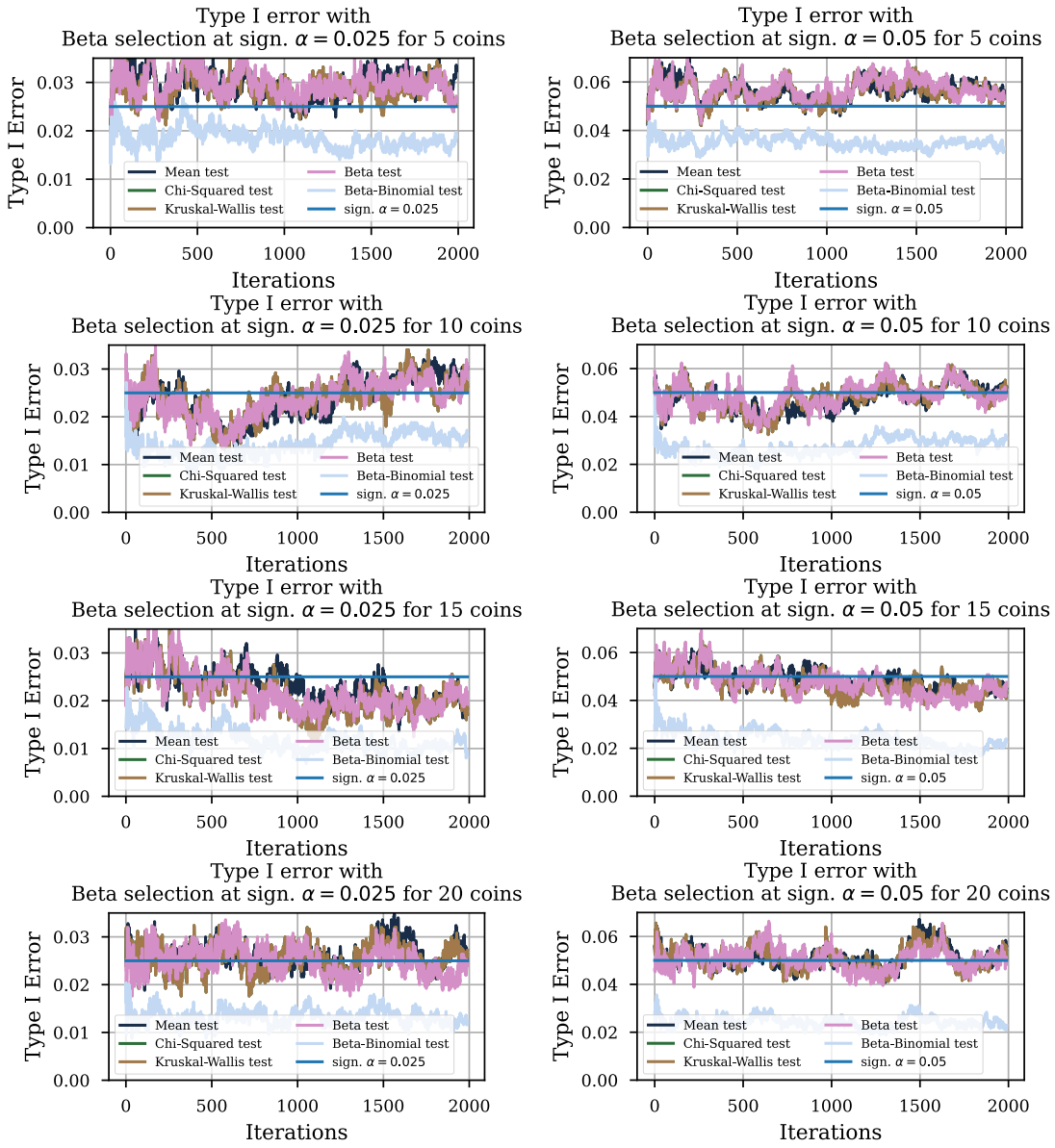


Figure B.5.: Type I error under adaptive beta sampling: we observe that a test based on our test statistic, although being slightly conservative due to the Bonferroni correction, consistently conforms to the required significance level. The permutation-based tests are not that conservative and can be overly optimistic in instances with few coins.

Figure B.6 compares the type I error of the test based on our Beta-Binomial statistic but for different sampling strategies. We observe that our test conforms to the required type

I error rate, regardless of the sampling strategy used. This behavior is consistent across iteration count and number of coins, which aligns with our proof, where we found that the used sampling strategy does not influence the distribution under $H0$.

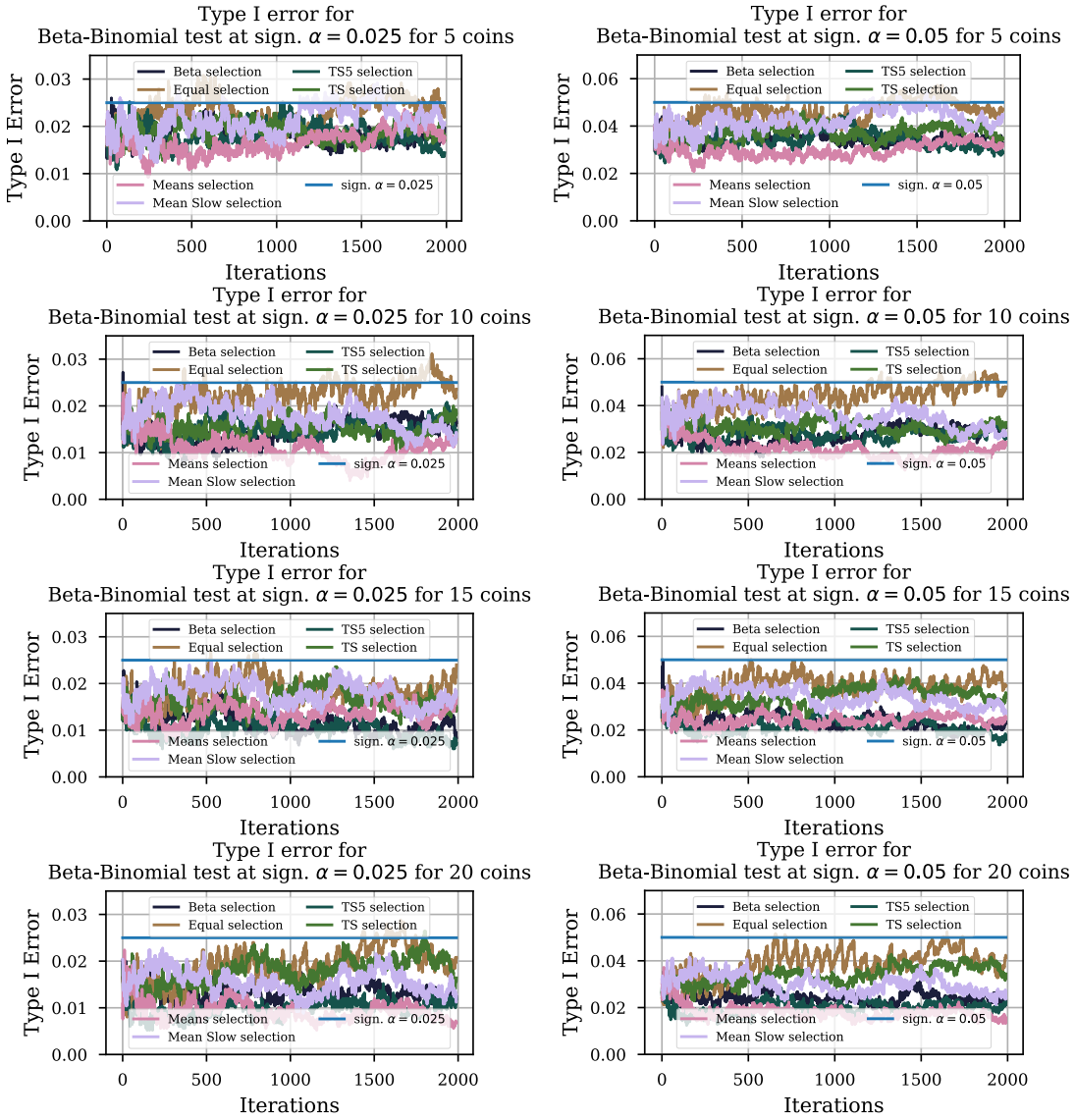


Figure B.6.: Type I error of the test based on our Beta-Binomial statistic for different sampling strategies – we observe that our test conforms to the required type I error rate, regardless of the sampling strategy used. Further, the behavior over iteration count and number of coins is very consistent.

Figure B.7 shows the type I error of the test based on our Beta-Binomial statistic for different numbers of coins, once for equal sampling, and once for our beta sampling strategy. We observe that our test statistic behaves similarly for any number of coins, regardless of whether Equal sampling or our Beta sampling is used. Thus, we can confirm that our sampling strategy as well as our test statistic is robust against different numbers of coins.

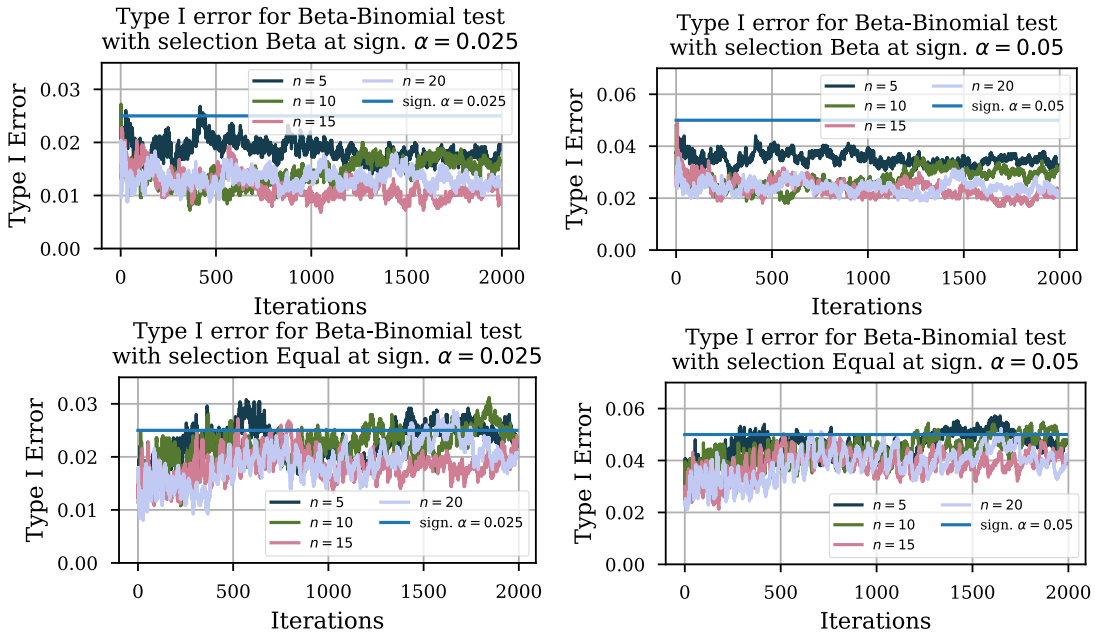


Figure B.7.: Type I error of our Beta-Binomial statistic for different numbers of coins, with our beta sampling (top) vs equal sampling (bottom) – We see that our test statistic behaves similarly for any number of coins, and maintains the required type I error under equal sampling as well as under our beta sampling.

B.2.3. Alpha Error with Bonferroni Correction

In this Section we will investigate the impact of the used Bonferroni correction on the subtest type I error.

In Section 6.6.1 we found, that our new test statistic is slightly less powerful than permutation based methods. Similarly, we observed in Section 6.6.2, that we obtain a slightly lower type I error under H_0 than the user requires. Both is somewhat expected because it builds on the Bonferroni correction, which is known to be conservative [Gar23], leading to the reduction in power and type I error.

Recap that our test is made up of multiple subtests, one for each coin, and that the alpha level of each subtest α_x is calculated based on the required overall significance level α using the Bonferroni correction. Investigating the subtest results, we find that they are positively correlated. That is, if one subtest is significant, it is more likely that another subtest will also be significant. As a result the combined type I error will be lower, even if the subtest type I error matches what is required. Meaning that the combined test *under-rejects* H_0, leading to a loss of power under H_1.

We will now verify, that the subtest type I error, which is not effected by the correction, matches indeed with the required subtest alpha level α_x provided by the Bonferroni correction. Note however, that the variance of the type I error is now significantly higher, even so we still perform 10000 repetitions, because the Bonferroni-corrected subtest alpha level α_x is much lower than the significance level α of the combined test. That is, $\alpha_x \in [0.005, 0.0025, 0.0017]$ for $|\mathcal{X}| \in \{5, 10, 15\}$ and $\alpha \in \{0.05, 0.025\}$, reaching the limit where we can accurately estimate the type I error. Thus we will not show the type I error for $|\mathcal{X}| = 20$.

Figure B.8 shows the type I errors for each subtest individually, for different significance levels and different numbers of coins. We see in every plot shown in Figure B.8 that the subtests meet the required type I error.

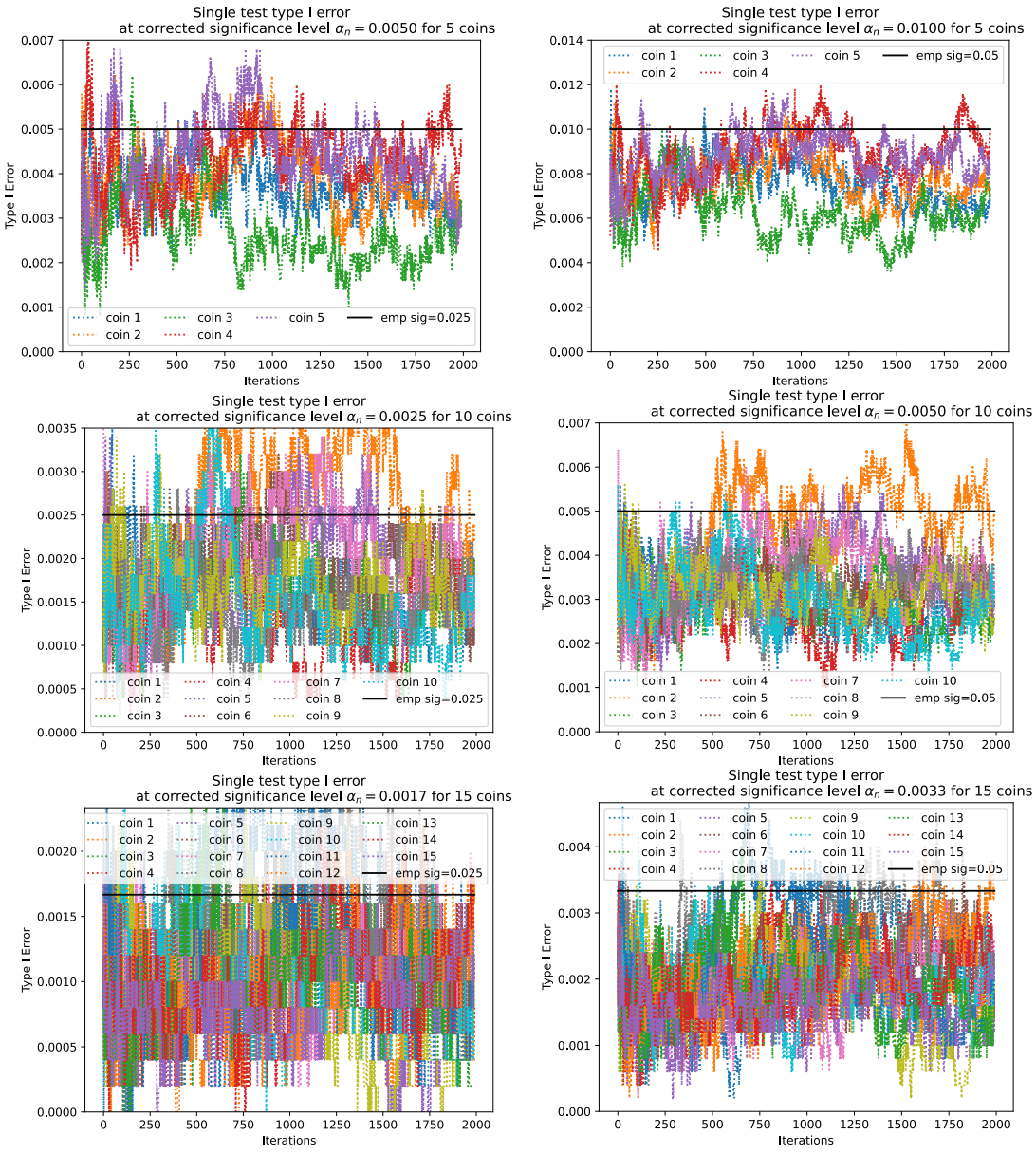


Figure B.8.: Type I error of the subtests with Bonferroni-corrected significance level: we see that the subtests meet the required type I error rate.

Part II.

Glossaries

Acronyms

AAIL Active and Adaptive Incremental Learning. 37, 42

AKD Adaptive Knowledge Discovery. iii–ix, 3–7, 11–31, 35, 48, 96, 97, 99, 101, 103, 132, 135–138, 145, 177, 179

AL Active Learning. 5, 6, 17, 29, 30, 35–38, 41, 107

AMT Adaptive Multi-sample Testing. iii, vi, ix–xi, 6, 12, 101–104, 106–116, 118, 120, 122–124, 126, 128, 130, 132, 136, 138, 145, 146, 148, 150, 152, 154, 156, 158, 160, 162, 178, 181

ANOVA Analysis of Variance. 106, 115

AOCC Active One-class Classification. 30

AU Aleatoric Uncertainty. 20–28

AUC Area Under the Curve. 86, 95

BAI Best Arm Identification. 18, 30, 31, 103–105

BIK Brownian Integral Kernel. iii, v, viii, ix, xi, 5, 51–54, 56–64, 66–70, 72, 135, 137, 143, 144, 177, 183

BK Brownian Kernel. ix, 52–54, 56, 63, 66–70, 72, 177

BMT Bernoulli Multi-sample Testing. 107, 108, 183

BNK Brownian Kernel with added white noise. 63, 66–68

BUCB Bayesian upper confidence bound. 102, 110, 111

CAL Classic Active Learning. 41, 44, 45

CD Concept Drift. 4, 35, 37, 39

CE Change Error. 42, 44, 45

CFP Channel-wise Feature Pyramid. 87

CI Change Ideal. 42, 45

CM Consecutive Measurement. 41, 43, 45, 48

CNN Convolutional Neural Network. 86–90, 92, 95

CoFRP Continuous-Fiber Reinforced Plastics. 79

CPU Central Processing Unit. 4

CRISP-DM Cross-Industry Standard Process for Data Mining. 3

DEAL Data Efficient Active Learning. iii, v, viii, 5, 12, 35–38, 40–48, 135, 137, 177

DGP Data-Generating Process. vii, 3–6, 11–14, 19–22, 28–31, 35, 39, 48, 51, 53, 72, 80, 96, 103, 132, 135, 137, 138, 177, 179

EA Evolutionary Algorithm. 94, 95

EER Expected Error Reduction. 18

EI Expected Improvement. 18

EU Epistemic Uncertainty. 20–22, 24–27

FEM Finite Element Method. 5, 7, 14, 77, 79, 80, 96

GP Gaussian Process. 5, 36, 38–43, 52–59, 63, 69, 72, 137, 181, 183

GPU Graphic Processing Unit. 14

GT Ground Truth. 42, 64, 69, 71, 72

HIPE High-resolution Industrial Production Energy. 7, 63, 66

i.i.d. Independent and Identically Distributed. 31, 102, 115, 117, 124, 126, 127, 132, 136

IT Independence Testing. 105, 132, 138

KD Knowledge Discovery. iii, v, vi, 3

KDD Knowledge Discovery in Databases. 3

KW Kruskal–Wallis Test. 106, 115, 126, 154

MAB Multi-armed Bandit. 145

MAE Mean Absolute Error. 64, 66, 67, 85, 86, 88, 89, 91–93, 177

MCMC Markov Chain Monte Carlo. 54

MD Multi-sample Detection. 103, 105–107

ML Machine Learning. 78

MLP Multi-Layer Perceptron. 87–89, 94, 95, 177

MSE Mean Square Error. 64, 66, 67, 85, 86, 92, 93

MT Multi-sample Testing. vii, ix, 6, 99, 101–107, 109–111, 113, 115, 118, 121, 132, 135, 136, 179, 183

OA Objective Alignment. iii, v, vii, 5–7, 81, 85, 90, 92, 93, 96, 135, 138

QBC Query-by-Committee. 18, 26, 27

RBF Radial Basis Function. 38, 40, 41, 43, 53, 54

RBFIK Radial Basis Function Integral Kernel. 52, 53, 63, 66, 67

RL Reinforcement Learning. 31

RMSE Root Mean Squared Error. 42–46, 86, 88–91, 93, 177

SAL Stream-based Active Learning. viii, 35, 38–40, 181

SuMO Surrogate Model-based Optimization. ix, 6, 12, 30, 77, 78, 85, 86, 90, 94–96, 107

SVM Support Vector Machine. 25, 54

TS Thompson Sampling. 117, 118, 122

TU Total Uncertainty. 20, 27

TVD Total Variation Distance. 118, 119, 129, 132, 178

UCB Upper Confidence Bound. 18, 19, 25

UQ Uncertainty Quantification. vii, viii, 3, 5, 6, 11, 19–21, 23–25, 135

WMAE Weighted Mean Absolute Error. 64, 66, 67, 183

Notation

Θ Parameter space of parameters θ. 22–24, 26, 103, 171, 174

$\Theta_{\mathbf{D}}$ Parameter space of of all parameters θ that satisfy the data $\Theta_{\mathbf{D}} = \{\theta \in \Theta \mid h_\theta \in \mathcal{V}_\Theta\} \subset \Theta$. 24, 26

$\delta t_{\boldsymbol{meas}}$ Measurement time interval after which a measurement is performed. 41, 45

$\bar{y}_{\theta x}$ Mean of an estimated random outcome $Y_{\theta x}$. 25

$\mathbb{T}$ Time horizon in which a stream X can be observed, typically $\mathbb{T} = \mathbb{R}_+$. 38, 42, 55, 61, 143, 171, 174

$\mathbf{D}$ Data set $\mathbf{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$ made up of $(x_i, y_{i)}$ pairs. 11, 15, 22–26, 41, 53, 171, 173, 174

$\mathbf{X}$ Input set $\mathbf{X} = \{x_1, x_2, \ldots, x_N\} \subset \mathcal{X}$ containing all inputs from a dataset $\mathbf{D}$. 15, 22, 25, 53, 171

$\mathbf{Y}$ Observation set $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\} \subset \mathcal{Y}$ containing all outcomes from a dataset $\mathbf{D}$. 15, 22, 23, 102, 103, 107–110, 112, 113, 115–118, 171, 172

$\mathbf{Y}_{\theta x}$ Prediction sample taken from a Estimated outcome variable $Y_{\theta x}$. 22

$\mathbf{Y}_\theta$ Prediction sets are a set of all predictions made for the input set $\mathbf{X}$. 22, 23, 53, 172

$\mathcal{H}$ Hypothesis space of hypothesis H. 22, 23, 171

$\mathcal{H}_\Theta$ Parametrized hypothesis space of a parametrized hypothesis H_θ. 22–24, 26

$\mathcal{H}_\theta$ Parametrized space hypothesis provide a predictive space $\mathcal{Y}_{\theta x}$. 22, 171

$\mathcal{V}$ Version space contains all the hypothesis that satisfy the given data $\mathcal{V}(\mathcal{H}, \mathbf{D}) \subset \mathcal{H}$. 23, 24, 26, 171

$\mathcal{X}$ Input space of intervention variable X. 13, 22–25, 27, 102, 103, 105, 108–118, 120–122, 162, 171, 172

$\mathcal{Y}$ Outcome space of outcome variable Y. 13, 22–24, 26, 171

$\mathcal{Y}_{\theta x}$ Predictive space given by a parametrized space hypothesis $\mathcal{H}_\theta$. 22, 171

Θ Random parameters. 26, 173

$\Theta_{\mathbf{D}}$ Random parameters given the data $\mathbf{D}$. 24, 27

$\mathscr{L}$ Overall loss is calculated between the set of all predictions $\mathbf{Y}_\theta$ and the set of ground truth observations $\mathbf{Y}$. 23

ℓ Element loss is calculated between a single prediction $y_{\theta x}$ and its ground truth observation y_x. 23, 24, 26

$\sigma_{\theta x}$ Variance of an estimated random outcome $Y_{\theta x}$. 25

θ Parameters of a parametrized hypothesis H_θ. 11, 22–24, 26, 27, 39–41, 53, 86, 92, 95, 103, 171, 172, 174

θ^* Optimal parameters. 23, 24

$*_{\mathcal{F}\mathcal{F}'}$ Full (two-time) integrated function in respect to both variables. 56–58, 60, 143

$*_{\mathcal{F}f'}$ One time integrated function in respect to the first variable. 56–58

$*_{f\mathcal{F}'}$ One time integrated function in respect to the second variable. 58

$*_{ff'}$ Original two variable function without integration. 56–58, 143

B Brownian motions are stochastic processes $B : t \mapsto B(t)$ characterized by random increments $\delta B(\delta t) = B(t + \delta t) - B(t) \forall t \in \mathbb{T}$, where these increments follow the normal distribution $\delta B(\delta t) \sim \mathcal{N}(0, \delta t)$. 38, 40, 55, 172

C Concept $Y_X = C(X)$ underling a DGP. 13, 28, 35, 39, 42, 43, 55, 80, 103, 172

c Sample function of a concept $C(\mathcal{X})$. 13, 42, 44

$\mathbf{do}(X = x)$ Intervention, i.e., actively setting a intervention variable X to a intervention value x. 13, 22, 36, 37, 103, 174

F^* Probabilistic Optimal point-wise Bayes predictor. 23

f^* Optimal point-wise Bayes predictor. 23, 24

H Stochastic hypothesis underlying a model; hypothesis and model are thereby interchangeable. 11, 23, 40, 171, 174

h Hypothesis underlying a model. 22–24, 171

H_θ Parametrized hypothesis with parameters θ. 22, 24, 26, 171, 172, 174, 175

h_{xi} Number of heads in sample $\mathbf{Y}_{xi}$ at iteration i. 108, 110, 115

I Maximum number of iterations after which an algorithm terminates. 102, 108, 109, 112–115, 117, 118, 120, 121, 123

i Specific iteration in which an action, intervention, or observation was performed. 11, 13, 28, 86, 95, 102, 108, 109, 117, 171, 174

$k(x, x')$ Kernel Functions, in the context of *GP* also called covariance functions, measure the similarity between two inputs x, x' as a scalar product in a not explicitly defined higher dimensional space. 38, 56

N Training set size refers to the number of data points (x, y_x) within a dataset $\mathbf{D}$. 41, 58, 102, 103, 105, 107–110, 112–116, 143

n Training set index refers to a specific index n of data point x_n, y_n within a dataset $\mathbf{D}$. 80, 84–86, 92, 107, 108, 110, 113, 115, 173

N_c Number of changes within a time series of length t_{end}. 40, 43

$P(\Theta_\mathbf{D})$ Posterior distribution over random parameters $\Theta_\mathbf{D}$ given the data $\mathbf{D}$. 24, 26, 27

$P(P(Y_{\theta x}) \mid \Theta_\mathbf{D})$ Distribution over possible predictive distributions $P(Y_{\theta x})$ given the data $\mathbf{D}$. 24, 26

$P(X)$ Marginal probability measure of a intervention variable X. 22

$p(x)$ Marginal probability mass or density function for a observation x. 22, 23

$P(X, Y)$ Joint probability measure of the data, i.e., a intervention variable X and outcome variable Y. 22, 23

$p(x, y)$ Joint probability mass or density function for a observation x and y. 22, 23

$P(Y \mid \mathbf{do}(X = x))$ Conditional probability measure of a outcome variable Y conditioned on a intervention value y. 23, 24

$P(Y)$ Marginal probability measure of a outcome variable Y. 22

$p(y)$ Marginal probability mass or density function for a observation y. 22

$P(Y \mid X)$ Conditional probability measure of a outcome variable Y conditioned on a intervention variable X. 22

$p(y \mid x)$ Conditional probability mass or density function for a observation y conditioned on intervention value x. 22, 23

$P(Y_X)$ Probability measure of a conditional outcome variable Y_X. 22

$p(y_x)$ Probability mass or density function for a conditional observation y_x. 22

$P(Y_{\theta x})$ Predictive distribution of a estimated outcome variable $Y_{\theta x}$. 22–24, 173

$p(y_{\theta x})$ Predictive mass or density function $p(y_{\theta x}) = p(y_{\theta x} \mid x)$ of a point prediction $y_{\theta x}$. 22, 24–27, 173, 174

$P(Y_{\theta x} \mid \mathbf{do}(X = x))$ Predictive distribution of a Estimated outcome variable $Y_{\theta x}$, $P(Y_{\theta x} \mid \text{do}(X = x) = P(Y_{\theta x})$. 22, 23, 25, 27

$p_{\mathbf{D}}(y_{\theta x})$ Bayesian predictive mass or density function $p_{\mathbf{D}}(y_{\theta x}) = \int_\Theta p(y_{\theta x}) p_{\mathbf{D}}(\theta)\, \mathrm{d}\theta$ given the data $\mathbf{D}$. 24, 27, 174

p_x Success probability of a Bernoulli-distributed random variables $Y_x \sim \text{Bernoulli}(p_x)$. 107, 110–112, 174

$p_{\mathbf{D}}(\theta)$ Posterior density over parameters θ given the data $\mathbf{D}$. 24, 27, 174

t Specific time of an action, intervention, or observation. 13, 22, 35, 36, 38–40, 42–44, 46, 55–58, 60–62, 64, 143, 172–174

X Data streams are random functions which can be observed at a random times within the time horizon $\mathbb{T}$. 38–40, 42, 171, 174

X Random intervention variable of a DGP or random input variable to a model. 13, 22, 23, 27, 35, 37, 39, 103, 105, 171–175

x Observed data streams or time series are sample functions $x(\mathbb{T}) \leftarrow X(\mathbb{T})$ which enable the observation of a value $x(t) = x_t$ at a given time t. 38, 39, 42–44, 174

x Observation $x \leftarrow X$ or the intervention value $do(X = x)$ of a intervention variable X, often used as a specific input to a model H. 11, 13, 15, 22–24, 26–28, 31, 36–43, 55, 56, 80, 102–105, 107–113, 115–118, 123, 137, 162, 171–175, 183

Y Global random Outcome Variable not conditioned on intervention variable X. 13, 22, 23, 37–39, 102, 103, 109, 117, 171, 173, 174

$y_{i/t}$ Observation made without specific reference to x and instead with reference to a iteration i or time t. 11, 13, 15, 22–24, 26, 28, 38–40, 42, 44, 57, 64, 80, 102, 103, 107–110, 112, 113, 115, 171, 173

Y_X Random outcome variable for a given intervention variable X, equivalent to $Y \mid X$. 13, 22, 35, 39, 103, 172, 173

Y_x Random outcome variable for a given intervention value x, equivalent to $Y \mid X = x$. 102, 103, 107–109, 111, 112, 115, 116, 174

y_x Observation made when $do(X = x)$. 13, 22, 23, 26, 27, 36, 37, 103, 108, 113, 172, 173

$Y_{\theta^* x}$ Estimated random outcome variable of a stochastic parametrized hypothesis $H_\theta(x)$ parametrized with optimal parameters θ^*. 23

$y_{\theta i/t}$ Point prediction made by a parametrized hypothesis H_θ at iteration i or time t. 22, 39, 43, 57, 64

$Y_{\theta x}$ Estimated random outcome variable of a stochastic parametrized hypothesis $H_\theta(x) = Y_{\theta x}$. 22–24, 171–175

$y_{\theta x}$ Point prediction made by a parametrized hypothesis H_θ. 22–24, 26, 36, 172, 173

Y_θ Estimated random outcome variable of a stochastic parametrized hypothesis $H_\theta(X) = Y_{\theta x}$ not conditioned on any x. 27

List of Figures

List of Tables

List of Algorithms

List of Theorems

Bibliography

[AB10] Jean-Yves Audibert and Sébastien Bubeck. 'Best Arm Identification in Multi-Armed Bandits'. In: *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*. 2010 (cited on page 19).

[AB21] Vadim Arzamasov and Klemens Böhm. 'REDS: Rule Extraction for Discovering Scenarios'. In: *Proceedings of the 2021 International Conference on Management of Data*. SIGMOD '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 115–128. DOI: 10.1145/3448016.3457301 (cited on page 30).

[ABM10] Jean-Yves Audibert, Sébastien Bubeck and Rémi Munos. 'Best Arm Identification in Multi-Armed Bandits'. In: *COLT - 23th Conference on Learning Theory - 2010*. Omnipress, 2010, pp. 41–53 (cited on page 107).

[Abr97] Petter Abrahamsen. 'A Review of Gaussian Random Fields and Correlation Functions'. In: *Rapport 917* (1997) (cited on pages 143, 144).

[ABT21] Johannes Allotey, Keith T. Butler and Jeyan Thiyagalingam. 'Entropy-Based Active Learning of Graph Neural Network Surrogate Models for Materials Properties'. In: *arXiv:2108.02077 [cond-mat]* (2021) (cited on page 30).

[ACF02] Peter Auer, Nicolò Cesa-Bianchi and Paul Fischer. 'Finite-Time Analysis of the Multiarmed Bandit Problem'. In: *Machine Learning* 47.2 (2002), pp. 235–256. DOI: 10.1023/A:1013689704352 (cited on page 18).

[AD99] S. Argamon-Engelson and I. Dagan. 'Committee-Based Sample Selection for Probabilistic Classifiers'. In: *Journal of Artificial Intelligence Research* 11 (1999), pp. 335–360. DOI: 10.1613/jair.612 (cited on page 26).

[Akb23] M. Eren Akbiyik. *Data Augmentation in Training CNNs: Injecting Noise to Images*. 2023. DOI: 10.48550/arXiv.2307.06855 (cited on page 15).

[App+17] R. R. Appino, T. Mühlpfordt, T. Faulwasser and V. Hagenmeyer. 'On Solving Probabilistic Load Flow for Radial Grids Using Polynomial Chaos'. In: *2017 IEEE Manchester PowerTech*. 2017, pp. 1–6. DOI: 10/gjmqhr (cited on page 25).

[ASE18] Antreas Antoniou, Amos Storkey and Harrison Edwards. *Data Augmentation Generative Adversarial Networks*. 2018. DOI: 10.48550/arXiv.1711.04340 (cited on pages 16, 17).

[Awa+20] Noor Awad, Gresa Shala, Difan Deng, Neeratyoy Mallik, Matthias Feurer, Katharina Eggensperger, Andre' Biedenkapp, Diederick Vermetten, Hao Wang, Carola Doerr et al. 'Squirrel: A switching hyperparameter optimizer'. In: *arXiv preprint arXiv:2012.08180* (2020) (cited on page 94).

[BFB24] Béla H. Böhnke, Edouard Fouché and Klemens Böhm. 'DEAL: Data-Efficient Active Learning for Regression Under Drift'. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by De-Nian Yang, Xing Xie, Vincent S. Tseng, Jian Pei, Jen-Wei Huang and Jerry Chun-Wei Lin. Singapore: Springer Nature, 2024, pp. 188–200. DOI: 10.1007/978-981-97-2266-2_15 (cited on pages 5, 35).

[BFB25] Béla H. Böhnke, Edouard Fouché and Klemens Böhm. 'The Brownian Integral Kernel: A New Kernel for Modeling Integrated Brownian Motions'. In: *Data Science: Foundations and Applications*. Ed. by Xintao Wu, Myra Spiliopoulou, Can Wang, Vipin Kumar, Longbing Cao, Xiangmin Zhou, Guansong Pang and Joao Gama. Singapore: Springer Nature, 2025, pp. 122–134. DOI: 10.1007/978-981-96-8295-9_8 (cited on pages 5, 51).

[Bif+10] Albert Bifet, Geoff Holmes, Richard Kirkby and Bernhard Pfahringer. 'MOA: Massive Online Analysis'. In: *J. Mach. Learn. Res.* (2010) (cited on page 42).

[Bis+18] Simon Bischof, Holger Trittenbach, Michael Vollmer, Dominik Werle, Thomas Blank and Klemens Böhm. 'HIPE: An Energy-Status-Data Set from Industrial Production'. In: *e-Energy*. ACM, 2018, pp. 599–603 (cited on pages 7, 63).

[Bis95] Christopher Bishop. 'Training with Noise Is Equivalent to Tikhonov Regularization'. In: *Neural Computation* 7.1 (1995), pp. 108–116. DOI: 10.1162/neco.1995.7.1.108 (cited on page 15).

[BKC13] Albert P. Bartók, Risi Kondor and Gábor Csányi. 'On Representing Chemical Environments'. In: *Physical Review B* 87.18 (2013), p. 184115. DOI: 10.1103/PhysRevB.87.184115 (cited on page 16).

[BKew] Béla H. Böhnke and Nadja Klein. 'AMT: Adaptive Multi-Sample Testing for Data Efficiency on Binomial Data'. In: *Data Science: Foundations and Applications*. Singapore: Springer Nature, under review (cited on pages 6, 101).

[BML25] Ha Manh Bui, Iliana Maifeld-Carucci and Anqi Liu. *Calibrated Uncertainty Sampling for Active Learning*. 2025. DOI: 10.48550/arXiv.2510.03162 (cited on page 26).

[Böh+24a] Bela H. Böhnke, Aleksandr Eismont, Clemens Zimmerling, Luise Kärger and Klemens Böhm. 'How Domain Knowledge Can Improve Machine Learning Surrogates for Manufacturing Process Optimization – a Comparative Study'. In: *Procedia CIRP*. 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024) 130 (2024), pp. 145–153. DOI: 10.1016/j.procir.2024.10.069 (cited on page 6).

[Böh+24b] Bela H. Böhnke, Aleksandr Eismont, Clemens Zimmerling, Luise Kärger and Klemens Böhm. 'How Domain Knowledge Can Improve Machine Learning Surrogates for Manufacturing Process Optimization – a Comparative Study'. In: *Procedia CIRP*. 57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024) 130 (2024), pp. 145–153. DOI: 10.1016/j.procir.2024.10.069 (cited on pages 16, 77).

[Boi+18] P. Boisse, J. Colmars, N. Hamila, N. Naouar and Q. Steer. 'Bending and wrinkling of composite fiber preforms and prepregs. A review and new developments in the draping simulations'. In: *Comp. P. B* 141 (2018), pp. 234–249 (cited on page 80).

[Bou+19] Mohamed Amine Bouhlel, John T. Hwang, Nathalie Bartoli, Rémi Lafage, Joseph Morlier and Joaquim R.R.A. Martins. 'A Python surrogate modeling framework with derivatives'. In: *Adv. Eng. Softw.* 135 (2019), p. 102662 (cited on page 78).

[Boy07] Phillip Boyle. 'Gaussian Processes for Regression and Optimisation'. In: *Thesis at WGTN* (2007) (cited on page 63).

[BST17] Philip Bachman, Alessandro Sordoni and Adam Trischler. 'Learning Algorithms for Active Learning'. In: *ICML*. 2017 (cited on page 37).

[BW22] Mickaël Binois and Nathan Wycoff. 'A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization'. In: *ACM Trans. Evol. Learn. Optim.* 2.2 (2022), 8:1–8:26. DOI: 10.1145/3545611 (cited on pages 27, 31).

[C05] Klebaner Fima C. *Introduction to stochastic calculus with applications*. WSPC, 2005 (cited on pages 13, 38, 55).

[Car+18] Giovanni De Carne, Giampaolo Buticchi, Marco Liserre and Constantine Vournas. 'Load Control Using Sensitivity Identification by Means of Smart Transformer'. In: *IEEE Trans. Smart Grid* (2018) (cited on pages 37, 38).

[Car+19] Giovanni De Carne, Giampaolo Buticchi, Marco Liserre and Constantine Vournas. 'Real-Time Primary Frequency Regulation Using Load Power Control by Smart Transformers'. In: *IEEE Trans. Smart Grid* (2019) (cited on page 37).

[CDL20] Shuxiao Chen, Edgar Dobriban and Jane H. Lee. *A Group-Theoretic Framework for Data Augmentation*. 2020. DOI: 10.48550/arXiv.1907.10905 (cited on page 15).

[CGJ94] David Cohn, Zoubin Ghahramani and Michael Jordan. 'Active Learning with Statistical Models'. In: *Advances in Neural Information Processing Systems*. Vol. 7. MIT Press, 1994 (cited on page 18).

[CGJ96] D. A. Cohn, Z. Ghahramani and M. I. Jordan. 'Active Learning with Statistical Models'. In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 129–145. DOI: 10.1613/jair.295 (cited on page 27).

[CGK18] Roberto Cipolla, Yarin Gal and Alex Kendall. 'Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics'. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491. DOI: 10.1109/CVPR.2018.00781 (cited on page 16).

[Che+15] S. Chen, L.T. Harper, A. Endruweit and N.A. Warrior. 'Formability optimisation of fabric preforms by controlling material draw-in through in-plane constraints'. In: *Compos Part A* (2015) (cited on page 80).

[Che+19] Ching Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith Mv, Stefan Stojanov and James M. Rehg. 'Unsupervised 3D Pose Estimation with Geometric Self-Supervision'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (2019), pp. 5707–5717. DOI: 10.1109/CVPR.2019.00586 (cited on page 17).

[Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi and Geoffrey Hinton. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. DOI: 10.48550/arXiv.2002.05709 (cited on pages 16, 17).

[Che+24] Pengguang Chen, Shu Liu, Hengshuang Zhao, Xingquan Wang and Jiaya Jia. *GridMask Data Augmentation*. 2024. DOI: 10.48550/arXiv.2001.04086 (cited on page 15).

[CN08] Rui M. Castro and Robert D. Nowak. 'Minimax Bounds for Active Learning'. In: *IEEE Transactions on Information Theory* 54.5 (2008), pp. 2339–2353. DOI: 10.1109/TIT.2008.920189 (cited on page 31).

[Coh88] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988 (cited on page 105).

[Coh93] David Cohn. 'Neural Network Exploration Using Optimal Experiment Design'. In: *Advances in Neural Information Processing Systems*. Vol. 6. Morgan-Kaufmann, 1993 (cited on page 27).

[Cui+24] Lingxi Cui, Huan Li, Ke Chen, Lidan Shou and Gang Chen. *Tabular Data Augmentation for Machine Learning: Progress and Prospects of Embracing Generative AI*. 2024. DOI: 10.48550/arXiv.2407.21523 (cited on page 16).

[CV98] Thierry Cutsem and Costas Vournas. *Voltage Stability of Electric Power Systems*. Springer US, 1998 (cited on page 52).

[CW16] Taco Cohen and Max Welling. 'Group Equivariant Convolutional Networks'. In: *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2016, pp. 2990–2999 (cited on page 17).

[Das+22] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja and Ashwin Srinivasan. 'A review of some techniques for inclusion of domain-knowledge into deep neural networks'. In: *Sci Rep* 12 (2022), p. 1040 (cited on page 92).

[Don+09] Han Dong, Ma Jin, He Ren-mu and Dong Z.Y. 'A Real Application of Measurement-Based Load Modeling in Large-Scale Power Grids and its Validation'. In: *IEEE Trans. Power Systems* (2009) (cited on page 37).

[EB20] Adrian Englhardt and Klemens Böhm. 'Exploring the Unknown ? Query Synthesis in One-Class Active Learning'. In: *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, 2020, pp. 145–153. DOI: 10.1137/1.9781611976236.17 (cited on page 30).

[EE04] A. Endruweit and P. Ermanni. 'The in-plane permeability of sheared textiles. Experimental observations and a predictive conversion model'. In: *Comp. P. A* 35.4 (2004), pp. 439–451 (cited on page 80).

[EGS26] Michael Eid, Mario Gollwitzer and Manfred Schmitt. *Statistik und Forschungsmethoden*. 6th ed. Mit Online-Materialien. Weinheim: Julius Beltz, 2026. ISBN: 978-3-621-28970-2 (cited on page 105).

[EMA18] Schulz Eric, Speekenbrink Maarten and Krause Andreas. 'A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions'. In: *J. Math. Psychol.* (2018) (cited on pages 51, 53).

[Eng+20] Adrian Englhardt, Holger Trittenbach, Dennis Vetter and Klemens Böhm. 'Finding the Sweet Spot: Batch Selection for One-Class Active Learning'. In: *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, 2020, pp. 118–126. DOI: 10.1137/1.9781611976236.14 (cited on page 30).

[Ern04] Michael D. Ernst. 'Permutation Methods: A Basis for Exact Inference'. In: *Statistical Science* 19.4 (2004), pp. 676–685 (cited on page 106).

[FAL17] Chelsea Finn, Pieter Abbeel and Sergey Levine. 'Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks'. In: *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135 (cited on page 17).

[FB19] Edouard Fouché and Klemens Böhm. 'Monte Carlo Dependency Estimation'. In: *Proceedings of the 31st International Conference on Scientific and Statistical Database Management*. SSDBM '19. Association for Computing Machinery, 23rd July 2019, pp. 13–24. DOI: 10.1145/3335783.3335795 (cited on page 113).

[FGL15] P. K. Friz, P. Gassiat and T. Lyons. 'Physical brownian motion in a magnetic field as a rough path'. In: *AMS* 367 (2015), pp. 7939–7955 (cited on page 51).

[FHF06] Brian Ferris, Dirk Hähnel and Dieter Fox. 'Gaussian Processes for Signal Strength-Based Location Estimation'. In: *Robotics: Science and Systems*. 2006 (cited on page 53).

[Fis92] R. A. Fisher. 'Statistical Methods for Research Workers'. In: *Breakthroughs in Statistics: Methodology and Distribution*. Ed. by Samuel Kotz and Norman L. Johnson. Springer, 1992, pp. 66–70. DOI: 10.1007/978-1-4612-4380-9_6 (cited on page 106).

[FM82] Kunihiko Fukushima and Sei Miyake. 'Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position'. In: *Pattern Recognition* 15.6 (1982), pp. 455–469. DOI: 10.1016/0031-3203(82)90024-3 (cited on page 17).

[FPD09] Peter Frazier, Warren Powell and Savas Dayanik. 'The Knowledge-Gradient Policy for Correlated Normal Beliefs'. In: *INFORMS Journal on Computing* 21.4 (2009), pp. 599–613. DOI: 10.1287/ijoc.1080.0314 (cited on pages 18, 28).

[FS08] Dean P. Foster and Robert A. Stine. 'α-Investing: A Procedure for Sequential Control of Expected False Discoveries'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.2 (2008), pp. 429–444. DOI: 10.1111/j.1467-9868.2007.00643.x (cited on pages 31, 106).

[FTM19] Yousefi Fariba, Smith Michael T and Álvarez Mauricio. 'Multi-task Learning for Aggregated Data using Gaussian Processes'. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019 (cited on pages 52, 53, 56, 63, 66, 144).

[Fuj+98] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga and Hozumi Tanaka. 'Selective Sampling for Example-based Word Sense Disambiguation'. In: *Computational Linguistics* 24.4 (1998). Ed. by Julia Hirschberg, pp. 573–597 (cited on page 25).

[Gam+14] João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy and Abdelhamid Bouchachia. 'A survey on concept drift adaptation'. In: *ACM Comput. Surv.* (2014) (cited on pages 35, 37, 39).

[Gao+20] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö. Arik, Larry S. Davis and Tomas Pfister. 'Consistency-Based Semi-Supervised Active Learning: Towards Minimizing Labeling Cost'. In: *European Conference on Computer Vision (ECCV)*. 2020. DOI: 10.1007/978-3-030-58607-2_30 (cited on page 18).

[Gar23] Miguel A. García-Pérez. 'Use and Misuse of Corrections for Multiple Testing'. In: *Methods in Psychology* 8 (Nov. 2023), p. 100120. DOI: 10.1016/j.metip.2023.100120 (cited on pages 111, 162).

[GDY19] Sam Greydanus, Misko Dzamba and Jason Yosinski. 'Hamiltonian Neural Networks'. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 1378. Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 15379–15389 (cited on page 17).

[GG16a] Yarin Gal and Zoubin Ghahramani. 'Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference'. In: *arXiv:1506.02158 [cs, stat]* (2016) (cited on page 26).

[GG16b] Yarin Gal and Zoubin Ghahramani. 'Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning'. In: *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2016, pp. 1050–1059 (cited on page 26).

[GHK17] Yarin Gal, Jiri Hron and Alex Kendall. 'Concrete Dropout'. In: *arXiv:1705.07832 [stat]* (2017) (cited on page 26).

[Gig04] Gerd Gigerenzer. 'Mindless Statistics'. In: *The Journal of Socio-Economics*. Statistical Significance 33.5 (2004), pp. 587–606. DOI: 10.1016/j.socec.2004.09.033 (cited on page 105).

[Gig93] Gerd Gigerenzer. 'The Superego, the Ego, and the Id in Statistical Reasoning'. In: *A Handbook for Data Analysis in the Behavioral Sciences: Methodological Issues*. Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc, 1993, pp. 311–339 (cited on page 105).

[Gil+20] Mostafa Gilanifar, Hui Wang, Lalitha Madhavi Konila Sriram, Eren Erman Ozguven and Reza Arghandeh. 'Multitask Bayesian Spatiotemporal Gaussian Processes for Short-Term Load Forecasting'. In: *IEEE Trans. Ind. Elect.* (2020) (cited on page 52).

[Gol+18] Florian Golemo, Adrien Ali Taiga, Aaron Courville and Pierre-Yves Oudeyer. 'Sim-to-Real Transfer with Neural-Augmented Robot Simulation'. In: *Proceedings of The 2nd Conference on Robot Learning*. PMLR, 2018, pp. 817–828 (cited on page 17).

[Goo+21] B. M. d. Gooijer, J. Havinga, H. Geijselaers and A. v. d. Boogaard. 'Evaluation of pod based surrogate models of fields resulting from nonlinear fem simulations'. In: *Adv. Mod. Sim. Eng. Sci.* 8 (2021), p. 25 (cited on pages 78, 84).

[Gor+10a] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene and Karel Crombecq. 'A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design'. In: *JMLR* 11 (2010), pp. 2051–2055 (cited on page 78).

[Gor+10b] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene and Karel Crombecq. 'A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design'. In: *Journal of Machine Learning Research* 11.68 (2010), pp. 2051–2055 (cited on page 107).

[GPy12] GPy. *A Gaussian process framework in python*. `http://github.com/SheffieldML/GPy`. since 2012 (cited on pages 41, 52, 58, 65).

[Gra11] Alex Graves. 'Practical Variational Inference for Neural Networks'. In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011 (cited on page 26).

[Gre+12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf and Alexander Smola. 'A Kernel Two-Sample Test'. In: *J. Mach. Learn. Res.* 13 (1st Mar. 2012), pp. 723–773 (cited on page 106).

[HCN11] Jarvis Haupt, Rui M. Castro and Robert Nowak. 'Distilled Sensing: Adaptive Sampling for Sparse Detection and Estimation'. In: *IEEE Transactions on Information Theory* 57.9 (2011), pp. 6222–6235. DOI: `10.1109/TIT.2011.2162269` (cited on pages 31, 106, 107, 113).

[Hen+18] J. N. Hendriks, C. Jidling, A. Wills and T. B. Schön. *Evaluating the squared-exponential covariance function in Gaussian processes with integral observations*. 2018. eprint: `1812.07319` (cited on pages 54, 59).

[Hil+13] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel and Oren Somekh. 'Distributed Exploration in Multi-Armed Bandits'. In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013 (cited on page 104).

[HKP11] Jiawei Han, Micheline Kamber and Jian Pei. *Data Mining: Concepts and Techniques, 3rd edition*. Morgan Kaufmann, 2011. ISBN: 978-0123814791 (cited on page 3).

[HKZ20] Lukas Hewing, Juraj Kabzan and Melanie N. Zeilinger. 'Cautious Model Predictive Control Using Gaussian Process Regression'. In: *IEEE Trans. Control Syst. Technol.* 28 (2020), pp. 2736–2743 (cited on page 78).

[HMT25] Shadi Haj-Yahia, Omar Mansour and Tomer Toledo. 'Incorporating Domain Knowledge in Deep Neural Networks for Discrete Choice Models'. In: *Transportation Research Part C: Emerging Technologies* 171 (2025), p. 105014. DOI: 10.1016/j.trc.2025.105014 (cited on page 16).

[Hoe+22] N. Hoernle, R. Karampatsis, V. Belle and K. Gal. 'Multiplexnet: towards fully satisfied logical constraints in neural networks'. In: *AAAI* 36 (2022), pp. 5700–5709 (cited on page 92).

[Hol+23] David Holzmüller, Viktor Zaverkin, Johannes Kästner and Ingo Steinwart. *A Framework and Benchmark for Deep Batch Active Learning for Regression*. 2023. DOI: 10.48550/arXiv.2203.09410 (cited on page 29).

[How+21] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe and Jasjeet Sekhon. 'Time-Uniform, Nonparametric, Nonasymptotic Confidence Sequences'. In: *The Annals of Statistics* 49.2 (2021), pp. 1055–1080. DOI: 10.1214/20-AOS1991 (cited on page 31).

[HPB08] Alex Holub, Pietro Perona and Michael C. Burl. 'Entropy-Based Active Learning for Object Recognition'. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2008, pp. 1–8. DOI: 10.1109/CVPRW.2008.4563068 (cited on page 18).

[HS12] Philipp Hennig and Christian J. Schuler. 'Entropy Search for Information-Efficient Global Optimization'. In: *Journal of Machine Learning Research* 13.57 (2012), pp. 1809–1837 (cited on pages 14, 19, 28).

[HT26] Tongchen Han and Solomon Tesfamariam. 'Multi-Fidelity Modelling for Uncertainty Quantification of Timber Beam-Column Connections Exposed to Standard Fire'. In: *Reliability Engineering & System Safety* 265 (2026), p. 111467. DOI: 10.1016/j.ress.2025.111467 (cited on page 26).

[Hu+25] Shouri Hu, Haowei Wang, Zhongxiang Dai, Bryan Kian Hsiang Low and Szu Hui Ng. 'Adjusted Expected Improvement for Cumulative Regret Minimization in Noisy Bayesian Optimization'. In: *Journal of Machine Learning Research* 26.46 (2025), pp. 1–33 (cited on page 28).

[Hua+24] Zijie Huang, Wanjia Zhao, Jingdong Gao, Ziniu Hu, Xiao Luo, Yadi Cao, Yuanzhou Chen, Yizhou Sun and Wei Wang. 'Physics-Informed Regularization for Domain-Agnostic Dynamical System Modeling'. In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. Vol. 37. NIPS '24. Red Hook, NY, USA: Curran Associates Inc., 2024, pp. 739–774 (cited on page 16).

[HVD15] Geoffrey Hinton, Oriol Vinyals and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. DOI: 10.48550/arXiv.1503.02531 (cited on page 17).

[HW21] Eyke Hüllermeier and Willem Waegeman. 'Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods'. In: *Machine Learning* 110.3 (2021), pp. 457–506. DOI: 10.1007/s10994-021-05946-3 (cited on pages 20–22, 26).

[IL15] Bertrand Iooss and Paul Lemaître. 'A Review on Global Sensitivity Analysis Methods'. In: *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*. Ed. by Gabriella Dellino and Carlo Meloni. Boston, MA: Springer US, 2015, pp. 101–122. DOI: 10.1007/978-1-4899-7547-8_5 (cited on page 31).

[IP19] Adriana S. Iwashita and João Paulo Papa. 'An Overview on Concept Drift Learning'. In: *IEEE Access* (2019) (cited on pages 35, 37).

[Iwa22] Tomoharu Iwata. 'Active Learning for Regression with Aggregated Outputs'. In: *CoRR* (2022) (cited on page 37).

[Jam+14] Kevin Jamieson, Matthew Malloy, Robert Nowak and Sébastien Bubeck. 'Lil' UCB : An Optimal Exploration Algorithm for Multi-Armed Bandits'. In: *Proceedings of The 27th Conference on Learning Theory*. PMLR, 2014, pp. 423–439 (cited on page 30).

[Jia+15] Honghua Jiang, Pandurang M. Kulkarni, Craig H. Mallinckrodt, Linda Shurzinske, Geert Molenberghs and Ilya Lipkovich. 'Adjusting for Baseline on the Analysis of Repeated Binary Responses With Missing Data'. In: *Statistics in Biopharmaceutical Research* 7.3 (3rd July 2015), pp. 238–250. DOI: 10.1080/19466315.2015.1067251 (cited on page 101).

[Jia+19] Hansi Jiang, Haoyu Wang, Wenhao Hu, Deovrat Kakde and Arin Chaudhuri. 'Fast Incremental SVDD Learning Algorithm with the Gaussian Kernel'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (2019), pp. 3991–3998. DOI: 10.1609/aaai.v33i01.33013991 (cited on page 30).

[Jid+17] Carl Jidling, Niklas Wahlström, Adrian Wills and Thomas B Schön. 'Linearly constrained Gaussian processes'. In: *NIPS*. Vol. 30. Curran Associates, Inc., 2017 (cited on page 54).

[JM15] Adel Javanmard and Andrea Montanari. *On Online Control of False Discovery Rate*. 2015. DOI: 10.48550/arXiv.1502.06197 (cited on pages 31, 106).

[Joã+04] Gama João, Medas Pedro, Castillo Gladys and Rodrigues Pedro. 'Learning with Drift Detection'. In: *artif. intell. adv. – SBIA*. 2004 (cited on page 42).

[Joh20] Max Kuhn and Kjell Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Taylor & Francis, 2020 (cited on page 16).

[Kal+25] Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala and Edwin Zhang. *Why Language Models Hallucinate*. 2025. arXiv: 2509.04664 [cs.CL] (cited on page i).

[Kan+18] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic and Bharath K. Sriperumbudur. 'Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences'. In: *arXiv:1807.02582 [cs, stat]* (2018) (cited on page 25).

[Kär+15] Luise Kärger, Alexander Bernath, Florian Fritz, Siegfried Galkin, Dino Magagnato, André Oeckerath, Alexander Schön and Frank Henning. 'Development and validation of a CAE chain for unidirectional fibre reinforced composite components'. In: *Comp. Struct.* 132 (2015), pp. 350–358 (cited on page 79).

[Kar+17a] Anuj Karpatne, Gowtham Atluri, James Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova and Vipin Kumar. 'Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data'. In: *IEEE Transactions on Knowledge and Data Engineering* 29.10 (2017), pp. 2318–2331. DOI: 10.1109/TKDE.2017.2720168 (cited on page 78).

[Kar+17b] Anuj Karpatne, William Watkins, Jordan S. Read and Vipin Kumar. 'Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling'. In: *CoRR* cs.LG (2017), p. 1710.11431 (cited on pages 16, 92).

[Kär+18] Luise Kärger, Siegfried Galkin, Clemens Zimmerling, Dominik Dörr, Johannes Linden, André Oeckerath and Klaus Wolf. 'Forming optimisation embedded in a CAE chain to assess and enhance the structural performance of composite components'. In: *Compos. Struct.* 192 (2018), pp. 143–152 (cited on pages 79, 80).

[Kar+20] Toni Karvonen, George Wynne, Filip Tronarp, Chris Oates and Simo Särkkä. 'Maximum Likelihood Estimation and Uncertainty Quantification for Gaussian Process Approximation of Deterministic Functions'. In: *SIAM/ASA JUQ* (2020) (cited on page 55).

[KB15] Diederik P. Kingma and Jimmy Ba. 'Adam: A Method for Stochastic Optimization'. In: *ICLR*. 2015, p. 1412.6980 (cited on page 86).

[KC19] Bartosz Krawczyk and Alberto Cano. 'Adaptive Ensemble Active Learning for Drifting Data Stream Mining'. In: *IJCAI*. 2019 (cited on page 37).

[KCG12] Emilie Kaufmann, Olivier Cappe and Aurelien Garivier. 'On Bayesian Upper Confidence Bounds for Bandit Problems'. In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. PMLR, 21st Mar. 2012, pp. 592–600 (cited on page 110).

[KCG16] Emilie Kaufmann, Olivier Cappé and Aurélien Garivier. 'On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models'. In: *J. Mach. Learn. Res.* 17.1 (1st Jan. 2016), pp. 1–42 (cited on page 107).

[KD09] Armen Der Kiureghian and Ove Ditlevsen. 'Aleatory or Epistemic? Does It Matter?' In: *Structural Safety*. Risk Acceptance and Risk Communication 31.2 (2009), pp. 105–112. DOI: 10.1016/j.strusafe.2008.06.020 (cited on pages 20, 21).

[KG20] Punit Kumar and Atul Gupta. 'Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey'. In: *Journal of Computer Science and Technology* 35.4 (2020), pp. 913–945. DOI: 10.1007/s11390-020-9487-4 (cited on page 17).

[KGC17] Jan Kukacka, Vladimir Golkov and Daniel Cremers. 'Regularization for Deep Learning: A Taxonomy'. In: *CoRR* cs.LG (2017), p. 1710.10686 (cited on page 92).

[KL13] Slawomir Koziel and Leifur Leifsson. *Surrogate-based modeling and optimization*. Springer, 2013 (cited on pages 77, 78, 84).

[KL19] Vince Kurtz and Hai Lin. *Kalman Filtering with Gaussian Processes Measurement Noise*. 2019. arXiv: 1909.10582 [stat.ML] (cited on page 54).

[KPW18] Bartosz Krawczyk, Bernhard Pfahringer and Michal Wozniak. 'Combining active learning with concept drift detection for data stream mining'. In: *IEEE BigData*. 2018 (cited on pages 37, 40).

[KSF17] Ksenia Konyushkova, Raphael Sznitman and Pascal Fua. 'Learning Active Learning from Data'. In: *NIPS*. 2017 (cited on page 37).

[Kum24] Saket Kumar. *2019-2024 US Stock Market Data*. 2024. DOI: 10.34740/KAGGLE/DSV/7553516 (cited on page 63).

[KW06] Christine Körner and Stefan Wrobel. 'Multi-Class Ensemble-Based Active Learning'. In: *Machine Learning: ECML 2006*. Ed. by Johannes Fürnkranz, Tobias Scheffer and Myra Spiliopoulou. Berlin, Heidelberg: Springer, 2006, pp. 687–694. DOI: 10.1007/11871842_68 (cited on page 26).

[KW11] Bartosz Kurlej and Michal Wozniak. 'Learning Curve in Concept Drift While Using Active Learning Paradigm'. In: *ICAIS*. 2011 (cited on page 37).

[KW12] Bartosz Kurlej and Michal Wozniak. 'Active learning approach to concept drift problem'. In: *Log. J. IGPL* (2012) (cited on page 37).

[KW52] William H. Kruskal and W. Allen Wallis. 'Use of Ranks in One-Criterion Variance Analysis'. In: *Journal of the American Statistical Association* 47.260 (1952), pp. 583–621. DOI: 10.2307/2280779 (cited on page 106).

[LBD18] Y. Lanoiselée, G. Briand and O. Dauchot. 'Statistical analysis of random trajectories of vibrated disks: towards a macroscopic realization of brownian motion'. In: *Physical Review* 98 (2018) (cited on page 51).

[LDA05] Ilya Lipkovich, Yuyan Duan and Saeeduddin Ahmed. 'Multiple Imputation Compared with Restricted Pseudo-Likelihood and Generalized Estimating Equations for Analysis of Binary Repeated Measures in Clinical Studies'. In: *Pharmaceutical Statistics* 4.4 (2005), pp. 267–285. DOI: 10.1002/pst.188 (cited on page 101).

[LG94] David D. Lewis and William A. Gale. 'A Sequential Algorithm for Training Text Classifiers'. In: *SIGIR '94*. Ed. by Bruce W. Croft and C. J. van Rijsbergen. London: Springer, 1994, pp. 3–12. DOI: 10.1007/978-1-4471-2099-5_1 (cited on page 25).

[LGL23] Ange Lou, Shuyue Guan and Murray Loew. 'CFPNet-M: A light-weight encoder-decoder based network for multimodal biomedical image real-time segmentation'. In: *Comput. Biol. Med.* 154 (2023), p. 106579 (cited on pages 87, 94).

[Li+20] Zhenxing Li, Fan Guo, Lei Chen, Kuangrong Hao and Biao Huang. 'Hybrid kernel approach to Gaussian process modeling with colored noises'. In: *Comput. Chem. Eng.* 143 (2020), p. 107067 (cited on page 63).

[Lia+18] Liang Liang, Minliang Liu, Caitlin Martin and Wei Sun. 'A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis'. In: *J. R. Soc. Interface.* 15 (2018), p. 20170844 (cited on page 78).

[Lin+22] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass and Frank Hutter. 'SMAC3: A versatile Bayesian optimization package for hyperparameter optimization'. In: *The Journal of Machine Learning Research* 23.1 (2022), pp. 2475–2483 (cited on page 94).

[Liu+20] F. Liu, Z. Gao, C. Yang and R. Ma. 'Extended kalman filters for continuous-time nonlinear fractional-order systems involving correlated and uncorrelated process and measurement noises'. In: *International Journal of Control, Automation and Systems* 18 (9 2020), pp. 2229–2241 (cited on page 54).

[Liu+21] Weike Liu, Hang Zhang, Zhaoyun Ding, Qingbao Liu and Cheng Zhu. 'A comprehensive active learning method for multiclass imbalanced data streams with concept drift'. In: *Knowl. Based Syst.* (2021) (cited on page 37).

[Liu+23] Sanmin Liu, Shan Xue, Jia Wu, Chuan Zhou, Jian Yang, Zhao Li and Jie Cao. 'Online Active Learning for Drifting Data Streams'. In: *IEEE Trans. Neural Networks Learn. Syst.* (2023) (cited on pages 37, 40).

[Liu+25] Siqi Liu, Ruina Li, Jiayi Zhou, Chaoyuan Dai, Jingui Yu and Qiaoxin Zhang. 'Physics-Based Data Augmentation Enables Accurate Machine Learning Prediction of Melt Pool Geometry'. In: *Applied Sciences* 15.15 (2025), p. 8587. DOI: 10.3390/app15158587 (cited on pages 16, 17).

[Liu04] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, 2004 (cited on page 116).

[LKN25] Zihan Liu, Prashant N. Kambali and C. Nataraj. 'Hybrid Adaptive Modeling Using Neural Networks Trained with Nonlinear Dynamics Based Features'. In: *Knowledge-Based Systems* 323 (2025), p. 113674. DOI: 10.1016/j.knosys.2025.113674 (cited on page 17).

[Lon+20] Krista Longi, Chang Rajani, Tom Sillanpää, Joni Mäkinen, Timo Rauhala, Ari Salmi, Edward Haeggström and Arto Klami. 'Sensor Placement for Spatial Gaussian Processes with Integral Observations'. In: *UAI*. Vol. 124. PMLR, 2020, pp. 1009–1018 (cited on pages 53, 54).

[Loo+19] Turab Lookman, Prasanna V. Balachandran, Dezhen Xue and Ruihao Yuan. 'Active Learning in Materials Science with Emphasis on Adaptive Sampling Using Uncertainties for Targeted Design'. In: *npj Computational Materials* 5.1 (2019), pp. 1–17. DOI: 10.1038/s41524-019-0153-8 (cited on pages 18, 30).

[LRS13] Georg Lindgren, Holger Rootzen and Maria Sandsten. *Stationary Stochastic Processes for Scientists and Engineers*. T&F, 2013 (cited on pages 13, 38, 40, 55, 56, 143).

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. 1st ed. Cambridge University Press, 2020. DOI: 10.1017/9781108571401 (cited on pages 30, 107).

[Lu+19] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama and Guangquan Zhang. 'Learning under Concept Drift: A Review'. In: *IEEE Trans. Knowl. Data Eng.* (2019) (cited on pages 35, 37).

[Mal19] Andrey Malinin. 'Uncertainty Estimation in Deep Learning with Application to Spoken Language Assessment'. In: (2019). DOI: 10.17863/CAM.45912 (cited on page 20).

[MCO07] David R. Musicant, Janara M. Christensen and Jamie F. Olson. 'Supervised Learning by Training on Aggregate Outputs'. In: *ICDM*. IEEE, 2007, pp. 252–261 (cited on page 51).

[Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. DOI: 10.48550/arXiv.1301.3781 (cited on page 16).

[MKH18] Vikram Mullachery, Aniruddh Khera and Amir Husain. *Bayesian Neural Networks*. 2018. DOI: 10.48550/arXiv.1801.07710 (cited on page 26).

[MN14] Matthew L. Malloy and Robert D. Nowak. *Near-Optimal Adaptive Compressed Sensing*. 2014. DOI: 10.48550/arXiv.1306.6239 (cited on page 31).

[MN98] A. McCallum and K. Nigam. 'Employing EM and Pool-Based Active Learning for Text Classification'. In: *International Conference on Machine Learning*. 1998 (cited on page 26).

[MO05] H. Mori and M. Ohmi. 'Probabilistic short-term load forecasting with Gaussian processes'. In: *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. IEEE, 2005 (cited on page 52).

[Mob+21] Aryan Mobiny, Pengyu Yuan, Supratik K. Moulik, Naveen Garg, Carol C. Wu and Hien Van Nguyen. 'DropConnect Is Effective in Modeling Uncertainty of Bayesian Deep Networks'. In: *Scientific Reports* 11.1 (2021), p. 5458. DOI: 10/gjkwdr (cited on page 26).

[Moč75] J. Močkus. 'On Bayesian Methods for Seeking the Extremum'. In: *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*. Ed. by G. I. Marchuk. Berlin, Heidelberg: Springer, 1975, pp. 400–404. DOI: 10.1007/978-3-662-38527-2_55 (cited on page 18).

[Mon+21] Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem et al. 'River: machine learning for streaming data in Python'. In: *JMLR* (2021) (cited on page 42).

[Moo63] Alexander McFarlane Mood. *Introduction to the Theory of Statistics*. 2d ed. [by] Alexander M. Mood [and] Franklin A. Graybill. McGraw-Hill, 1963. 443 pp. (cited on pages 103, 106, 113).

[Mou20] Dimitris Mourtzis. 'Simulation in the design and operation of manufacturing systems: state of the art and new trends'. In: *Int. J. Prod. Res.* 58 (2020), pp. 1927–1949 (cited on page 77).

[MSB16] Saad Mohamad, Moamar Sayed Mouchaweh and Abdelhamid Bouchachia. 'Active Learning for Data Streams Under Concept Drift and Concept Evolution'. In: *ECML-PKDD*. 2016 (cited on pages 37, 40).

[Müh+17] T. Mühlpfordt, T. Faulwasser, L. Roald and V. Hagenmeyer. 'Solving Optimal Power Flow with Non-Gaussian Uncertainties via Polynomial Chaos Expansion'. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 4490–4496. DOI: 10/gjmqhq (cited on page 25).

[MXZ06] Charles A. Micchelli, Yuesheng Xu and Haizhang Zhang. 'Universal Kernels'. In: *J. Mach. Learn. Res.* (2006) (cited on pages 40, 53).

[MZG16] X. Meng, J. Zhang and H. Guo. 'Quantum brownian motion model for the stock market'. In: *Physica A Stat. Mech. Appl.* 452 (2016), pp. 281–288 (cited on page 51).

[NG94] Andrew F. Neuwald and Philip Green. 'Detecting Patterns in Protein Sequences'. In: *Journal of Molecular Biology* 239.5 (23rd June 1994), pp. 698–712. DOI: 10.1006/jmbi.1994.1407 (cited on page 101).

[NP33a] J. Neyman and E. S. Pearson. 'On the Problem of the Most Efficient Tests of Statistical Hypotheses'. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231 (1933), pp. 289–337 (cited on page 105).

[NP33b] J. Neyman and E. S. Pearson. 'The testing of statistical hypotheses in relation to probabilities a priori'. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 29.4 (1933), pp. 492–510 (cited on page 105).

[Öch23] Andreas Öchsner. *Composite Mechanics*. Springer Cham, 2023 (cited on page 82).

[OR11] Simon O'Callaghan and Fabio Ramos. 'Continuous Occupancy Mapping with Integral Kernels'. In: *AAAI* 25 (2011), pp. 1494–1500 (cited on pages 54, 59).

[Pea92] Karl Pearson. 'On the Criterion That a Given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such That It Can Be Reasonably Supposed to Have Arisen from Random Sampling'. In: *Breakthroughs in Statistics: Methodology and Distribution*. Ed. by Samuel Kotz and Norman L. Johnson. Springer, 1992, pp. 11–28 (cited on page 106).

[Pfl+22] Noah Pflugradt, Peter Stenzel, Leander Kotzur and Detlef Stolten. 'LoadProfileGenerator: An Agent-Based Behavior Simulation for Generating Residential Load Profiles'. In: *Journal of Open Source Software* 7 (2022), p. 3574 (cited on pages 7, 63).

[Pfr+18] Julius Pfrommer, Clemens Zimmerling, Jinzhao Liu, Luise Kärger, Frank Henning and Jürgen Beyerer. 'Optimisation of manufacturing process parameters using deep neural networks as surrogate models'. In: *CIRP* 72 (2018), pp. 426–431 (cited on pages 78, 84, 94).

[PK16] Cheong Hee Park and Youngsoon Kang. 'An active learning method for data streams with concept drift'. In: *IEEE BigData*. 2016 (cited on pages 37, 42).

[PLB20] Tim Pearce, Felix Leibfried and Alexandra Brintrup. 'Uncertainty in Neural Networks: Approximately Bayesian Ensembling'. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 234–244 (cited on page 26).

[Pur+19] Zenith Purisha, Carl Jidling, Niklas Wahlström, Thomas B Schön and Simo Särkkä. 'Probabilistic approach to limited-data computed tomography reconstruction'. In: *Inverse Problems* 35 (2019), p. 105004 (cited on page 54).

[RFB15] Olaf Ronneberger, Philipp Fischer and Thomas Brox. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In: *MICCAI*. Springer, 2015, pp. 234–241 (cited on page 87).

[RJZ17] Carlos Riquelme, Ramesh Johari and Baosen Zhang. 'Online Active Linear Regression via Thresholding'. In: *AAAI*. 2017 (cited on pages 29, 35, 37).

[RL21] Rui Ren and Shaoyuan Li. 'Enhanced Gaussian Process Regression for Active Learning Model-based Predictive Control'. In: *CCC*. 2021, pp. 2731–2736 (cited on page 78).

[RM01] Nicholas Roy and Andrew McCallum. 'Toward Optimal Active Learning through Sampling Estimation of Error Reduction'. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 441–448 (cited on page 18).

[Rob52] Herbert Robbins. 'Some Aspects of the Sequential Design of Experiments'. In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535 (cited on pages 31, 106).

[RR10] Steven Reece and Stephen Roberts. 'An introduction to Gaussian processes for the Kalman filter expert'. In: *2010 13th International Conference on Information Fusion*. 2010, pp. 1–9 (cited on pages 54, 58).

[Run25] Thomas A. Runkler. *Data Analytics: Models and Algorithms for Intelligent Data Analysis - A Comprehensive Introduction*. Wiesbaden: Springer Fachmedien, 2025. DOI: 10.1007/978-3-658-45951-2 (cited on page 3).

[Rus+20] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband and Zheng Wen. *A Tutorial on Thompson Sampling*. 2020 (cited on page 117).

[RW06] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. The MIT Press, 2006 (cited on pages 58, 144).

[RW25] Aaditya Ramdas and Ruodu Wang. 'Hypothesis Testing with E-values'. In: *Foundations and Trends® in Statistics* 1 (2025), pp. 1–390 (cited on pages 106, 129).

[Sal08] Andrea Saltelli, ed. *Global Sensitivity Analysis: The Primer*. Chichester, England Hoboken, NJ: John Wiley, 2008. DOI: 10.1002/9780470725184 (cited on page 31).

[SAL19] Michael Thomas Smith, Mauricio A Alvarez and Neil D Lawrence. *Gaussian Process Regression for Binned Data*. 2019. arXiv: 1809.02010 [stat.ML] (cited on pages 52, 53).

[SB18a] Marelli Stefano and Sudret Bruno. 'An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis'. In: *Struct. Saf.* (2018) (cited on page 37).

[SB18b] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second edition. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018 (cited on pages 31, 116).

[SC08] Burr Settles and Mark Craven. 'An Analysis of Active Learning Strategies for Sequence Labeling Tasks'. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Ed. by Mirella Lapata and Hwee Tou Ng. Honolulu, Hawaii: Association for Computational Linguistics, 2008, pp. 1070–1079 (cited on page 25).

[SCR07] Burr Settles, Mark Craven and Soumya Ray. 'Multiple-Instance Active Learning'. In: *Advances in Neural Information Processing Systems*. Vol. 20. Curran Associates, Inc., 2007 (cited on page 26).

[SD91] Jocelyn Sietsma and Robert J. F. Dow. 'Creating Artificial Neural Networks That Generalize'. In: *Neural Networks* 4.1 (1991), pp. 67–79. DOI: 10.1016/0893-6080(91)90033-2 (cited on page 15).

[SDW01] Tobias Scheffer, Christian Decomain and Stefan Wrobel. 'Active Hidden Markov Models for Information Extraction'. In: *Advances in Intelligent Data Analysis*. Ed. by Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher and Gabriela Guimaraes. Berlin, Heidelberg: Springer, 2001, pp. 309–318. DOI: 10.1007/3-540-44816-0_31 (cited on page 25).

[Sen+14] Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff and Eyke Hüllermeier. 'Reliable Classification: Learning Classifiers That Distinguish Aleatoric and Epistemic Uncertainty'. In: *Information Sciences* 255 (2014), pp. 16–29. DOI: 10.1016/j.ins.2013.07.030 (cited on page 20).

[Set09] Burr Settles. *Active Learning Literature Survey*. Technical Report. University of Wisconsin-Madison Department of Computer Sciences, 2009 (cited on pages 17, 25, 27).

[Set12] Burr Settles. 'Active Learning'. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (2012), pp. 1–114. DOI: 10.2200/S00429ED1V01Y201207AIM018 (cited on pages 17, 18, 26, 29, 38, 39, 57, 107).

[SH12] Simo Sarkka and Jouni Hartikainen. 'Infinite-Dimensional Kalman Filtering Approach to Spatio-Temporal Gaussian Process Regression'. In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Neil D. Lawrence and Mark Girolami. Vol. 22. Proceedings of Machine Learning Research. PMLR, 2012, pp. 993–1001 (cited on pages 54, 58).

[Sha+19a] Jicheng Shan, Hang Zhang, Weike Liu and Qingbao Liu. 'Online Active Learning Ensemble Framework for Drifted Data Streams'. In: *IEEE Trans. Neural Networks Learn. Syst.* (2019) (cited on page 37).

[Sha+19b] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich and T. Tsunoda. 'Deepinsight: a methodology to transform a non-image data to an image for convolution neural network architecture'. In: *Sci Rep* 9 (2019), p. 11399 (cited on page 90).

[She+18] Mahmoud Shepero, Dennis van der Meer, Joakim Munkhammar and Joakim Widén. 'Residential probabilistic load forecasting: A method using Gaussian process designed for electric load data'. In: *Applied Energy* (2018) (cited on page 52).

[She00] Colin Shearer. 'The CRISP-DM model: the new blueprint for data mining'. In: *Journal of data warehousing* 5.4 (2000), pp. 13–22 (cited on page 3).

[Sim+01] T. W. Simpson, J. D. Poplinski, P. Koch and J. K. Allen. 'Metamodels for computer-based engineering design: survey and recommendations'. In: *EWC* 17 (2001), pp. 129–150 (cited on page 78).

[SK22] A. Sharma and D. Kumar. 'Classification with 2-d convolutional neural networks for breast cancer diagnosis'. In: *Sci Rep* 12 (2022), p. 21857 (cited on page 90).

[Sma+18] Francesco Smarra, Achin Jain, Tullio de Rubeis, Dario Ambrosini, Alessandro D'Innocenzo and Rahul Mangharam. 'Data-driven model predictive control using random forests for building energy optimization and climate control'. In: *Appl. Energy* 226 (2018), pp. 1252–1272 (cited on page 78).

[SOS92] H. S. Seung, M. Opper and H. Sompolinsky. 'Query by Committee'. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. New York, NY, USA: Association for Computing Machinery, 1992, pp. 287–294. DOI: 10.1145/130385.130417 (cited on pages 18, 26).

[Spr+16] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner and Frank Hutter. 'Bayesian Optimization with Robust Bayesian Neural Networks'. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Curran Associates Inc., 5th Dec. 2016, pp. 4141–4149 (cited on page 107).

[SR13] Gábor J. Székely and Maria L. Rizzo. 'Energy Statistics: A Class of Statistics Based on Distances'. In: *Journal of Statistical Planning and Inference* 143.8 (1st Aug. 2013), pp. 1249–1272 (cited on page 106).

[SS02] Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT, 2002 (cited on page 56).

[SS18] Ozan Sener and Silvio Savarese. *Active Learning for Convolutional Neural Networks: A Core-Set Approach*. 2018. DOI: 10.48550/arXiv.1708.00489 (cited on page 18).

[SS19] A. Solin and S. Särkkä. 'Hilbert space methods for reduced-rank gaussian process regression'. In: *Stat. Comput.* 30 (2019), pp. 419–446 (cited on page 54).

[Sun+19] Ligang Sun, Hamza Alkhatib, Boris Kargoll, Vladik Kreinovich and Ingo Neumann. 'Ellipsoidal and Gaussian Kalman Filter Model for Discrete-Time Nonlinear Systems'. In: *Mathematics* 7.12 (2019) (cited on page 54).

[Tag+17] Amirhossein Taghvaei, Jana de Wiljes, Prashant G. Mehta and Sebastian Reich. 'Kalman Filter and Its Modern Extensions for the Continuous-Time Nonlinear Filtering Problem'. In: *Journal of Dynamic Systems, Measurement, and Control* 140.3 (Nov. 2017), p. 030904 (cited on page 54).

[Tal19] E. F. Talantsev. 'Classifying Superconductivity in Compressed H3S'. In: *Modern Physics Letters B* 33.17 (20th June 2019), p. 1950195. DOI: 10.1142/S0217984919501951 (cited on page 101).

[Tan+19] Yusuke Tanaka, Toshiyuki Tanaka, Tomoharu Iwata, Takeshi Kurashima, Maya Okawa, Yasunori Akagi and Hiroyuki Toda. 'Spatially Aggregated Gaussian Processes with Multivariate Areal Outputs'. In: *NEURIPS*. Vol. 32. Curran Associates, Inc., 2019 (cited on pages 54, 59).

[TK01] Simon Tong and Daphne Koller. 'Support Vector Machine Active Learning with Applications to Text Classification'. In: *Journal of Machine Learning Research* 2.Nov (2001), pp. 45–66 (cited on page 25).

[TLK20] Ville Tanskanen, Krista Longi and Arto Klami. 'Non-Linearities in Gaussian Processes with Integral Observations'. In: *MLSP*. 2020 (cited on pages 52, 54).

[TLR02] Min Tang, Xiaoqiang Luo and Salim Roukos. 'Active Learning for Statistical Natural Language Parsing'. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. USA: Association for Computational Linguistics, 2002, pp. 120–127. DOI: 10.3115/1073083.1073105 (cited on page 25).

[TMU20] Ryo Takahashi, Takashi Matsubara and Kuniaki Uehara. 'Data Augmentation Using Random Image Cropping and Patching for Deep CNNs'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.9 (2020), pp. 2917–2931. DOI: 10.1109/TCSVT.2019.2935128 (cited on page 16).

[Tod+20] Marco Todescato, Andrea Carron, Ruggero Carli, Gianluigi Pillonetto and Luca Schenato. 'Efficient spatio-temporal Gaussian regression via Kalman filtering'. In: *Automatica* 118 (Aug. 2020), p. 109032 (cited on pages 54, 58).

[Tou14] Marc Toussaint. 'The Bayesian Search Game'. In: *Theory and Principled Methods for the Design of Metaheuristics*. Ed. by Yossi Borenstein and Alberto Moraglio. Springer Berlin Heidelberg, 2014, pp. 129–144. DOI: 10.1007/978-3-642-33206-7_7 (cited on page 122).

[Tur+19] Lookman Turab, Balachandran Prasanna V., Xue Dezhen and Yuan Ruihao. 'Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design'. In: *Npj Comput. Mater.* (2019) (cited on page 37).

[Vas+23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. *Attention Is All You Need*. 2023. DOI: 10.48550/arXiv.1706.03762 (cited on page 16).

[VBW16] Losing Viktor, Hammer Barbara and Heiko Wersing. 'KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift'. In: *ICDM*. 2016 (cited on page 42).

[Vii+23] J.V. Viisainen, F. Yu, A. Codolini, S. Chen, L.T. Harper and M.P.F. Sutcliffe. 'Rapidly predicting the effect of tool geometry on the wrinkling of biaxial NCFs during composites manufacturing using a deep learning surrogate model'. In: *Comp. Part B* 253 (2023), p. 110536 (cited on page 81).

[Wan+04] Zhou Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli. 'Image quality assessment: from error visibility to structural similarity'. In: *IEEE Trans. Image Process.* 13 (2004), pp. 600–612 (cited on page 92).

[Wan+16] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson and Nando De Feitas. 'Bayesian Optimization in a Billion Dimensions via Random Embeddings'. In: *Journal of Artificial Intelligence Research* 55 (19th Feb. 2016), pp. 361–387. DOI: 10/gf7bcv (cited on page 107).

[Wan+25] Zhiyuan Wang, Jinwoo Go, Byung-Jun Yoon, Nathan Urban and Xiaoning Qian. 'A Plug-and-Play Query Synthesis Active Learning Framework for Neural PDE Solvers'. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025 (cited on page 29).

[Wil+20] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach and Vipin Kumar. 'Integrating Physics-Based Modeling with Machine Learning: A Survey'. In: *arXiv:2003.04919 [physics, stat]* (2020) (cited on pages 16, 17, 78).

[Wil99] Leland Wilkinson. 'Statistical Methods in Psychology Journals'. In: *American Psychologist* (1999). DOI: 10.1037/0003-066X.54.8.594 (cited on page 105).

[Wim+23] Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl and Eyke Hüllermeier. 'Quantifying Aleatoric and Epistemic Uncertainty in Machine Learning: Are Conditional Entropy and Mutual Information Appropriate Measures?' In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. PMLR, 2023, pp. 2282–2292 (cited on pages 20, 22, 27).

[WLH19] Dongrui Wu, Chin-Teng Lin and Jian Huang. 'Active learning for regression using greedy sampling'. In: *Inf. Sci.* (2019) (cited on pages 29, 35, 37).

[WS14] Dan Wang and Yi Shang. 'A New Active Labeling Method for Deep Learning'. In: *2014 International Joint Conference on Neural Networks (IJCNN)*. 2014, pp. 112–119. DOI: 10.1109/IJCNN.2014.6889457 (cited on page 25).

[WS25] Thorben Werner and Lars Schmidt-Thieme. *Bayesian Active Learning By Distribution Disagreement*. 2025. DOI: 10.48550/arXiv.2501.01248 (cited on page 29).

[WTP21] Sifan Wang, Yujun Teng and Paris Perdikaris. 'Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks'. In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3055–A3081. DOI: 10.1137/20M1318043 (cited on page 16).

[Wu13] Yongcheng Wu. 'Active Learning Based on Diversity Maximization'. In: *2nd International Symposium on Computer, Communication, Control and Automation (ISCCCA 2013)*. Atlantis Press, 2013, pp. 822–825. DOI: 10.2991/isccca.2013.207 (cited on page 18).

[XD19] Junyao Xie and Stevan Dubljevic. 'Discrete-Time Kalman Filter Design for Linear Infinite-Dimensional Systems'. In: *Processes* 7.7 (2019) (cited on page 54).

[Xu+09] Jiangbin Xu, Chao Yang, Jian Zhao and Lingda Wu. 'Fast Modeling of Realistic Clouds'. In: *CNMT*. 2009 (cited on page 64).

[Xu+24] Qingsong Xu, Yilei Shi, Jonathan Bamber, Ye Tuo, Ralf Ludwig and Xiao Xiang Zhu. *Physics-Aware Machine Learning Revolutionizes Scientific Paradigm for Machine Learning and Process-based Hydrology*. 2024. DOI: 10.48550/arXiv.2310.05227 (cited on page 17).

[YK19] Donggeun Yoo and In So Kweon. 'Learning Loss for Active Learning'. In: *CVPR*. 2019 (cited on page 37).

[Yos+14] Jason Yosinski, Jeff Clune, Yoshua Bengio and Hod Lipson. 'How Transferable Are Features in Deep Neural Networks?' In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014. DOI: 10.48550/arXiv.1411.1792 (cited on pages 16, 17).

[Zha+18] Hang Zhang, Weike Liu, Jicheng Shan and Qingbao Liu. 'Online Active Learning Paired Ensemble for Concept Drift and Class Imbalance'. In: *IEEE Access* (2018) (cited on page 37).

[Zha+22] Lili Zhang, Yuda Wu, Ping Jiang, Seung-Kyum Choi and Qi Zhou. 'A Multi-Fidelity Surrogate Modeling Approach for Incorporating Multiple Non-Hierarchical Low-Fidelity Data'. In: *Advanced Engineering Informatics* 51 (2022), p. 101430. DOI: 10.1016/j.aei.2021.101430 (cited on page 26).

[Zhu+10] Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou and Matthew Ma. 'Active Learning With Sampling by Uncertainty and Density for Data Annotations'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6 (2010), pp. 1323–1331. DOI: 10.1109/TASL.2009.2033421 (cited on pages 25, 27).

[Zim+19] Clemens Zimmerling, Daniel Trippe, Benedikt Fengler and Luise Kärger. 'An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks'. In: *ESAFORM* 2113 (2019), p. 020007 (cited on pages 81, 87).

[Zim+21] Clemens Zimmerling, Patrick Schindler, Julian Seuffert and Luise Kärger. 'Deep neural networks as surrogate models for time-efficient manufacturing process optimisation'. In: *ESAFORM* MS11 (2021), p. 3882 (cited on pages 18, 77–79, 84, 87, 88, 94).

[Zim+22] Clemens Zimmerling, Christian Poppe, Oliver Stein and Luise Kärger. 'Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning'. In: *Mater. Des.* 214 (2022), p. 110423 (cited on page 81).

[Zim23] Clemens Zimmerling. *Machine learning algorithms for efficient process optimisation of variable geometries at the example of fabric forming*. PhD-thesis at KIT, 2023 (cited on pages 80, 87).

[Zli+11a] Indre Zliobaite, Albert Bifet, Geoff Holmes and Bernhard Pfahringer. 'MOA Concept Drift Active Learning Strategies for Streaming Data'. In: *WAPA*. JMLR Proceedings. 2011 (cited on page 37).

[Zli+11b] Indre Zliobaite, Albert Bifet, Bernhard Pfahringer and Geoff Holmes. 'Active Learning with Evolving Streaming Data'. In: *ECML/PKDD (3)*. 2011 (cited on page 37).

[Zli+14] Indre Zliobaite, Albert Bifet, Bernhard Pfahringer and Geoffrey Holmes. 'Active Learning With Drifting Streaming Data'. In: *IEEE Trans. Neural Networks Learn. Syst.* (2014) (cited on pages 37, 40).

[Zli10] Indre Zliobaite. 'Learning under Concept Drift: an Overview'. In: *CoRR* (2010) (cited on pages 35, 37).