Karlsruhe Institute of Technology

Adversarial Attack Detection using Normalizing Flows

Bachelor Thesis

Rafig Gasimov

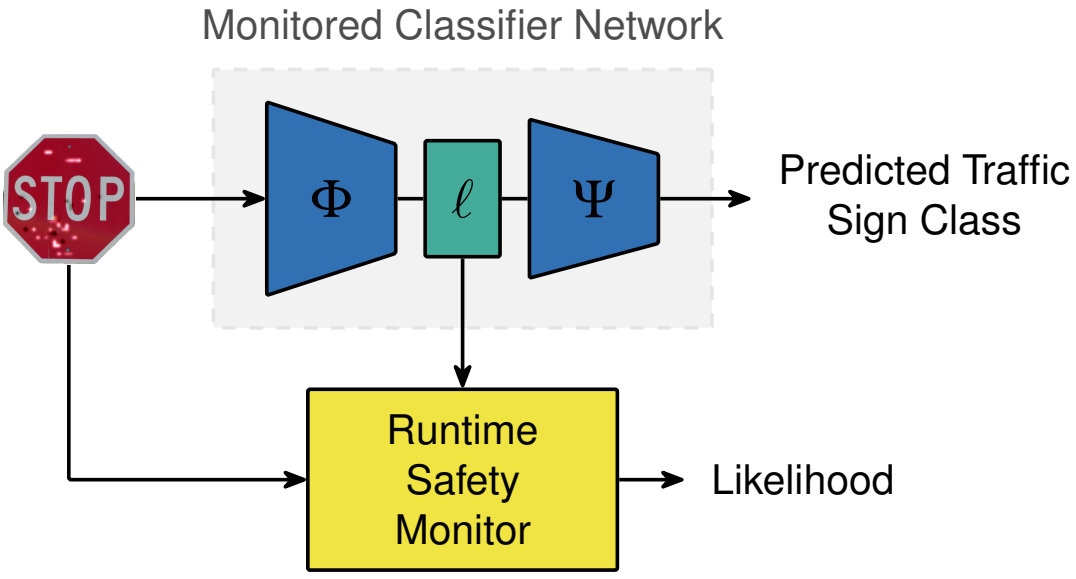
Department of Computer Science
KASTEL Institute of Information Security and Dependability
and
FZI Research Center for Information Technology

Reviewer:	Prof. Dr. Ralf Reussner
Second reviewer:	Prof. Dr.–Ing. J. M. Zöllner
Advisor:	M.Sc. Albert Schotschneider

Research Period: 01. Januar 2026 – 31. March 2026

Adversarial Attack Detection using Normalizing Flows

by
Rafig Gasimov



Bachelor Thesis
March 2026

FZI

Bachelor Thesis, FZI
Department of Computer Science, 2026
Reviewers: Prof. Dr. Ralf Reussner, Prof. Dr.–Ing. J. M. Zöllner

Department Technical Cognitive Systems
FZI Research Center for Information Technology

Affirmation

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Quellen und Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, *Rafig Gasimov*
March 2026

Abstract

The deployment of Deep Neural Networks within the perception pipelines of autonomous driving systems is critically hindered by their vulnerability to visually inconspicuous adversarial perturbations. In safety-critical applications such as Traffic Sign Recognition, these mathematical manipulations can force highly confident misclassifications, bypassing traditional Out-of-Distribution detection mechanisms that rely on compromised softmax probability distributions. To secure these perception systems without degrading their baseline accuracy, this thesis proposes and engineers an unsupervised, parallel runtime safety monitor based on exact density estimation utilizing the multi-scale Glow Normalizing Flow architecture.

The central scientific contribution of this research is a comprehensive, mathematically grounded ablation study investigating the optimal topological input space for generative anomaly detectors. The study directly compares an Input-Only flow architecture, which models the exact log-likelihood of the raw pixel space, against a novel Joint Input-Feature architecture. The joint configuration extracts intermediate semantic representations from the deep hidden layers of a frozen target classifier, spatially aligns them, and models the massively high-dimensional concatenated tensor, theoretically attempting to fuse low-level statistical anomalies with high-level semantic inconsistencies.

Extensive empirical evaluations were conducted across multiple deep learning backbones and highly diverse traffic sign datasets against optimal white-box L_∞ attacks. The results conclusively demonstrate that the Input-Only exact density estimator successfully identifies the high-frequency structural disruptions induced by digital adversarial perturbations, achieving robust detection margins. Conversely, the Joint Input-Feature architecture exhibits a severe degradation in detection efficacy. This counter-intuitive failure is formally attributed to the Semantic Alignment Paradox, wherein successfully optimized adversarial features perfectly mimic the natural semantic profile of the target class, thereby tricking the generative model into assigning a high likelihood score.

Furthermore, the computational overhead analysis proves that the massive quadratic inflation of Floating Point Operations and inference latency required to process the concatenated semantic features strictly prohibits real-time deployment. Ultimately, this thesis mathematically and empirically establishes that the lightweight Input-Only topology is the strictly superior, computationally viable paradigm for defending latency-constrained autonomous systems against digital adversarial threats.

Contents

1 Introduction

The deployment of Machine Learning (ML) systems in safety-critical domains has accelerated at an unprecedented rate. This transition moves algorithms from controlled academic benchmarks to real-world, dynamic, and highly unpredictable environments. Within the realm of autonomous driving, environmental perception models, particularly those responsible for Traffic Sign Recognition (TSR), serve as a foundational layer for vehicular navigation, trajectory planning, and overall passenger safety. An autonomous vehicle relies on high-fidelity perception pipelines to parse its surroundings, make split-second decisions, and adhere strictly to traffic regulations. Consequently, the operational reliability, interpretability, and robustness of the underlying Deep Neural Networks (DNNs) employed for these visual recognition tasks are of paramount importance. While modern deep learning architectures consistently achieve human-level accuracy on standard traffic sign benchmarks under ideal conditions, their reliability drops sharply under targeted adversarial interference.

1.1 Motivation: Vulnerabilities in Traffic Sign Recognition

Convolutional Neural Networks (CNNs), which form the structural backbone of most contemporary traffic sign recognition systems, have been shown to be profoundly susceptible to worst-case mathematical perturbations. Adversarial attacks systematically compute subtle, highly optimized noise patterns that cause incorrect predictions when additively combined with a benign input image [10]. The most challenging aspect of these attacks is their visual imperceptibility. By bounding the perturbation within a strict L_∞ mathematical norm, the adversary guarantees that the malicious modifications remain visually inconspicuous to human observers and standard optical sensors, even though they significantly alter the network's high-dimensional decision boundaries.

In the context of traffic signs, such misclassifications undermine the core safety guarantees legally and practically required for intelligent vehicular navigation. A single undetected adversarial perturbation could cause an autonomous driving system to confidently interpret a "Stop" sign as a "Speed Limit" sign. Because the autonomous agent's subsequent planning algorithms implicitly trust the outputs of the perception module, this sensory deception bypasses higher-level logical checks and potentially precipitates catastrophic physical consequences.

Historically, the primary methodology for mitigating these vulnerabilities has been proactive robustification, most notably through adversarial training. In this paradigm, adversarial examples are continuously synthesized and injected into the training dataset, forcing the model to learn smoother and more resilient decision boundaries. However, proactively defending against these threats is inherently flawed for widespread deployment. Adversarial training is exceptionally computationally expensive, frequently degrades the model's standard accuracy on clean data, and fundamentally struggles to generalize. A network robustified against one specific attack algorithm is often exhibits

severe vulnerabilities to a novel, unseen attack strategy.

This fundamental limitation strongly motivates the necessity for reactive, parallel security measures, commonly referred to as runtime safety monitors. A runtime monitor operates continuously and in parallel with the frozen, pre-trained target classifier during the inference phase. Its sole objective is to independently evaluate incoming data streams and flag anomalous, out-of-distribution, or maliciously altered inputs. By acting as a deterministic safety envelope, the monitor can intercept an adversarial input and trigger a system-level fail-safe. This fail-safe could include gracefully degrading vehicular operation or handing control back to a human driver before the classifier's potentially erroneous output can be acted upon.

1.2 Problem Statement: Density Estimation and OOD Detection

Detecting visually inconspicuous adversarial examples at runtime is intrinsically linked to the broader challenge of Out-of-Distribution (OOD) detection. In a theoretical ideal, a deployed classifier should express high confidence only when processing inputs drawn from the exact underlying data distribution it was trained on, and it should express high uncertainty when presented with heavily corrupted or maliciously altered data. Therefore, adversarial examples can be framed as a specific, mathematically optimized subset of OOD data designed specifically to exploit the network's blind spots.

However, many traditional OOD detection methodologies rely heavily on evaluating the target classifier's internal confidence scores, softmax probability distributions, or terminal logits. This dependency presents a critical security flaw, as these are precisely the computational components that an adversarial attack is engineered to manipulate and compromise. Modern deep neural networks are notoriously overconfident, and adversarial perturbations frequently induce incorrect predictions with near-perfect confidence scores. This phenomenon renders logit-based OOD detectors severely compromises the efficacy of logit-based OOD detectors against sophisticated white-box adversaries.

To develop a truly robust runtime monitor without altering the underlying classifier or relying on its compromised confidence metrics, one must accurately capture the true probability density of the benign data. Normalizing flows are a powerful class of probabilistic generative models particularly well-suited for this precise task [19]. By transforming a complex, unknown data distribution into a tractable base distribution via bijective mathematical mappings, normalizing flows provide exact log-likelihood estimation. By training a flow solely on clean, unperturbed data, the model learns the exact target distribution of normal traffic signs. During inference, any sample that deviates from this learned distribution receives a mathematically rigorous, low likelihood score and can subsequently be flagged as an anomaly.

Despite the mathematical elegance of normalizing flows, a fundamental architectural challenge arises regarding what specific topological information the generative model should evaluate to achieve optimal anomaly detection. A monitor could rely exclusively on modeling the raw input image distribution, assessing whether the pixel-level statistics align with clean data. Conversely, because adversarial attacks often aim to change the semantic meaning of an image, the monitor could leverage the hierarchical processing of the target classifier by extracting internal feature rep-

resentations and evaluating those for semantic inconsistencies. Identifying the optimal topological input space for generative anomaly detection forms the central problem addressed by this research.

1.3 Contributions

To systematically address the aforementioned problem, this thesis presents the design, architectural implementation, and rigorous empirical evaluation of a flow-based anomaly detection framework tailored for autonomous perception pipelines. The primary contributions of this work are detailed comprehensively in the following subsections.

1.3.1 Architectural Configurations for Runtime Monitoring

We propose and engineer a modular runtime safety monitor based on a multi-scale, Glow-style normalizing flow. Crucially, to investigate the optimal input space for density estimation, this thesis develops and formally defines two distinct architectural configurations for the flow-based monitor. The first configuration is an Input-Only architecture. This generative model evaluates the exact log-likelihood of the raw pixel data directly from the optical sensor, operating entirely independently of the target classifier's internal feature representations. The second configuration is a novel Joint Input-Feature architecture. In this paradigm, intermediate feature maps are extracted from a specific, defined layer of the frozen target classifier. These feature maps are spatially aligned via interpolation and concatenated with the raw input image. The normalizing flow is subsequently tasked with modeling the significantly more complex, high-dimensional joint distribution of the image and its corresponding semantic activations.

1.3.2 Comprehensive Ablation Study

A core analytical contribution of this work is a systematic, mathematically grounded ablation study comparing the two proposed architectures. We rigorously evaluate whether the joint modeling approach, which theoretically bridges low-level statistical pixel anomalies with high-level semantic anomalies by monitoring the network's internal state, actually improves the detection of visually inconspicuous adversarial attacks when compared to the computationally lighter input-only architecture. This ablation isolates the precise impact of incorporating internal classifier features into exact density estimation.

1.3.3 Empirical Evaluation Across Datasets and Threat Models

To ensure the findings are robust and not merely artifacts of a specific network topology, the proposed monitoring configurations are rigorously evaluated against a wide spectrum of deep learning backbones. This evaluation includes highly parameterized deep networks such as ResNet18 [12] and DenseNet121, as well as highly efficient, edge-optimized models like MobileNetV2 [32], ShuffleNet V2 [25], and EfficientNet B0 [38]. Furthermore, spatial invariance is evaluated utilizing Spatial Transformer Networks (CNN-STN) [16].

To guarantee generalizability across different environmental conditions and geographic signage standards, the evaluation is conducted across four distinct traffic sign datasets. These include the United States-based LISA Traffic Sign Dataset [28], the widely utilized German Traffic Sign Recognition Benchmark (GTSRB) [34], the modern high-resolution DFG Traffic Sign Dataset [37], and the purely synthetic Synset dataset. The monitor's robustness is tested explicitly and exclusively against a rigorous suite of digital adversarial algorithms, specifically the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD) [26], and the highly dynamic Auto-Projected Gradient Descent (APGD) [4].

1.4 Structure of the Thesis

To provide a comprehensive, logical progression from theoretical background to empirical findings, the remainder of this thesis is structured as follows.

Chapter 2 reviews the Related Work, exploring the historical progression of Out-of-Distribution detection methods and analyzing existing paradigms for runtime monitoring and adversarial defense in deep learning. Following the literature review, Chapter 3 establishes the mathematical Basis for this research. It details the theoretical mechanics of deep neural networks, formulates the mathematics of optimal adversarial attacks, and thoroughly defines the framework underlying exact density estimation via normalizing flows.

Chapter 4 presents the Methodology, formally describing the implementation of the two distinct flow architectures alongside the interceptive feature extraction mechanisms required to bridge the classifier and the generative monitor. Chapter 5 details the Experimental Setup and Evaluation, providing rigorous specifications for the evaluated datasets, target classifiers, and attack hyper-parameters, before deeply unpacking the results of the ablation study and analyzing the critical computational trade-offs. Finally, Chapter 6 provides a Conclusion, summarizing the definitive findings drawn from this research and highlighting potential, highly relevant avenues for future work in autonomous systems security.

2 Related Work

To ensure the reliable and secure deployment of deep neural networks in safety-critical environments, the machine learning research community has dedicated extensive efforts toward understanding and mitigating the vulnerabilities inherent to highly parameterized models. This chapter provides a comprehensive, conceptually driven review of the scientific literature that forms the foundation of this thesis. It begins by examining the broad paradigm of runtime safety monitoring for deep neural networks, exploring the topological strategies utilized to observe model behavior during inference. Subsequently, it delves into the extensive taxonomy of adversarial attacks and the corresponding landscape of defense methodologies, focusing strictly on relevant white-box and physical vulnerabilities within traffic sign recognition systems. The chapter then transitions to the parallel challenge of out-of-distribution detection, conceptually reviewing classical output-based and contemporary feature-space approaches for identifying distributional shifts. Finally, the chapter explores the application of normalizing flows as a theoretically grounded framework for exact anomaly detection, explicitly discussing the semantic limitations of pixel-level density estimation. The chapter concludes by defining the current research gap in the literature and articulating the precise methodological contributions of this thesis.

2.1 Runtime Safety Monitoring for Deep Neural Networks

The traditional paradigm of machine learning evaluation relies heavily on static test sets to quantify generalization error prior to deployment. However, in unconstrained open-world environments such as autonomous driving, it is practically impossible to anticipate and sample the entirety of the operational data distribution during the training phase. Consequently, state-of-the-art safety methodologies for deep neural network perception systems have increasingly shifted focus away from static pre-deployment verification toward dynamic, online verification mechanisms.

Before discussing external monitors, it is necessary to contextualize them against foundational uncertainty quantification techniques. Model uncertainty is broadly categorized into aleatoric uncertainty, which represents inherent data noise, and epistemic uncertainty, which represents uncertainty in model parameters due to insufficient training data [17]. Bayesian Neural Networks provide a rigorous framework for capturing epistemic uncertainty by learning probability distributions over the network weights. Because exact Bayesian inference is computationally intractable for deep convolutional networks, researchers have developed approximation methods such as Monte Carlo Dropout [7] and Deep Ensembles [21]. While these methods successfully quantify uncertainty and detect anomalies, they require executing the heavy perception pipeline multiple times for a single frame. In the highly latency-constrained environment of autonomous driving, this severe computational bottleneck directly motivates the paradigm of parallel runtime monitoring.

A runtime monitor adds an external, independent algorithmic entity that runs in parallel to a deterministic target model. The literature broadly categorizes these monitors based on the topological information they analyze. Input monitors operate exclusively on the raw data stream before it propagates through the neural network. The foundational premise is that unsafe or adversarially altered samples possess low-level statistical artifacts that distinguish them from the natural data manifold at the optical sensor level. While input monitoring benefits from being completely independent of the target classifier's architecture, it frequently struggles to differentiate between benign environmental noise and targeted adversarial perturbations optimized to be imperceptible in the raw pixel space.

To capture a deeper semantic understanding, internal representation monitors track deviations in the hidden activations of the target classifier. These monitors operate under the hypothesis that while an adversarial input might appear benign in pixel space, its progression through the highly non-linear transformations of the network will inevitably yield unnatural activation trajectories. For instance, techniques utilizing Topological Data Analysis on internal activations have empirically demonstrated that adversarial examples create uniquely persistent homologies distinct from clean data [8]. Furthermore, researchers have utilized Gram matrices to monitor feature correlations, establishing bounds on these correlations during training to detect adversarial attacks that force the network to utilize highly irregular combinations of otherwise normal features [33].

Output monitors operate at the terminal end of the classification pipeline, relying predominantly on confidence scores and the relative consistency of the network's terminal logits to flag unreliable predictions. While highly practical, they are inherently vulnerable to the overconfidence phenomenon prevalent in modern deep neural networks. Recognizing these distinct limitations, modern survey literature emphasizes the necessity of fusing multiple monitoring signals to ensure comprehensive diagnostic coverage [14].

2.2 Adversarial Attacks and Defense Methodologies

The discovery that highly accurate deep neural networks could be easily fooled by visually imperceptible perturbations demonstrated the limitations of how deep learning models perceive the world robustly [36]. Adversarial attacks are deliberately crafted input perturbations designed to geometrically shift a sample across a classifier's decision boundary.

In a white-box setting, the attacker possesses complete access to the target model's architecture, trained parameters, and internal gradients, allowing the adversary to utilize optimal mathematical optimization algorithms. The Fast Gradient Sign Method utilizes a single-step linear approximation of the model's loss landscape to rapidly generate perturbations [10]. To address the severe limitations of single-step approximations, Projected Gradient Descent applies iterative gradient steps, continually projecting the perturbed image back into a strictly defined constraint bound, and is widely regarded as the universal first-order adversary [26]. Beyond standard gradient descent, algorithms like DeepFool iteratively project the input toward the closest linearized decision boundary to calculate the absolute minimum required perturbation [29]. The Carlini and Wagner attack treats adversarial generation as a margin-based optimization problem, formulating a highly sophisticated loss function that balances perturbation imperceptibility with misclassification confidence, effec-

tively bypassing many early defense mechanisms [3]. More recently, AutoAttack introduced an ensemble of parameter-free attacks that dynamically adjust their optimization step sizes and utilize specialized loss functions to prevent gradient vanishing, establishing a rigorous modern benchmark for adversarial evaluation [4].

Traffic sign recognition provides a uniquely critical real-world attack surface. While digital bounded attacks manipulate the image within the digital pipeline, physical attacks involve tangible manipulations of the physical infrastructure. The Robust Physical Perturbations algorithm demonstrated that adversarial attacks could be effectively executed in the physical world by optimizing perturbations over a distribution of synthetic spatial transformations, ensuring printed stickers remain highly adversarial under varying environmental conditions [6]. Furthermore, the concept of the Adversarial Patch relies on semantic dominance rather than imperceptibility, introducing features that are highly dominant, causing the classifier to disregard the surrounding traffic sign [2]. Researchers have also demonstrated that natural phenomena, such as strategically projected shadows, can be carefully optimized to induce confident misclassifications without requiring any digital tampering [40].

To defend against these threats, the research community has proposed numerous paradigms. Proactive robustification, most notably Adversarial Training, robustifies the network weights by solving a min-max optimization problem during the training phase [26]. However, this approach induces a severe robustness-accuracy trade-off, which has been theoretically demonstrated to inherently increase standard risk on clean data when minimizing adversarial risk [39]. Reactive defenses, such as input transformation techniques, attempt to sanitize the image prior to classification by applying lossy transformations to destroy high-frequency noise [11]. Unfortunately, these are frequently vulnerable to adaptive attackers utilizing differentiable approximations to obfuscate gradients and bypass the sanitization process entirely [1]. Reconstruction-based approaches, such as MagNet, train autoencoders to reconstruct clean data, utilizing the reconstruction error as an anomaly score [27]. While autoencoders can capture manifold violations, they often suffer from blurry reconstructions that degrade the baseline accuracy of the perception module and remain bypassable by advanced adaptive attacks that enforce the perturbation to lie precisely within the autoencoder's learned manifold.

2.3 Out-of-Distribution Detection

While adversarial detection focuses on maliciously crafted perturbations, Out-of-Distribution (OOD) detection aims to identify test-time inputs that simply do not belong to the training distribution. The core objective is to prevent high-confidence misclassifications when a model is deployed in an open-world setting and encounters novel object classes or extreme environmental changes.

Classical baselines for OOD detection rely heavily on the classifier's terminal output. The Maximum Softmax Probability baseline operates on the observation that networks generally exhibit lower maximum confidence on OOD data [13]. However, due to the saturating nature of the exponential softmax function, this distinction is marginal, and networks frequently assign near-perfect confidence to completely unrecognizable inputs. To address this severe calibration issue,

the ODIN framework introduced temperature scaling to soften the softmax distribution, alongside a small input perturbation optimized to maximize the network's confidence, which effectively amplifies the separability between in-distribution and OOD samples [23]. Energy-based OOD detection maps the pre-softmax logits to a theoretically grounded energy surface, providing a much more reliable metric for identifying natural distribution shifts without being susceptible to softmax overconfidence [24].

More recent, state-of-the-art approaches operate on the principle that the high-dimensional embeddings learned by the network contain rich structural information about data familiarity that is lost during logit compression. By modeling the representations as class-conditional probability distributions, the Mahalanobis distance method efficiently measures the geometric distance from a test sample's feature vector to the closest known class cluster, establishing a robust feature-space anomaly score [22]. Similarly, non-parametric approaches utilizing nearest-neighbor retrieval within deep embedding spaces have demonstrated exceptional capabilities in identifying inputs that lack a dense neighborhood of familiar training samples [35].

2.4 Normalizing Flows and Generative Anomaly Detection

To effectively operate as an unsupervised runtime monitor, a system must accurately understand and map the complex distribution of benign data without relying on the target classifier's labels. Probabilistic generative modeling has emerged as a highly robust solution for exact anomaly detection.

Generative Adversarial Networks learn an implicit representation of the data distribution through a minimax game, but because they do not possess a mathematically tractable likelihood function, they cannot directly assign an exact anomaly score to an incoming image [9]. Variational Autoencoders offer a probabilistic alternative by defining an explicit graphical model, optimizing an evidence lower bound [20]. However, this reliance on a structural approximation rather than an exact mathematical guarantee frequently leads to poorly calibrated anomaly thresholds when attempting to detect highly subtle adversarial perturbations.

Normalizing Flows constitute a powerful class of probabilistic generative models that distinguish themselves through their reliance on exact, bijective formulations. By learning a sequence of strictly invertible transformations, they map complex data distributions to a simple base distribution [5]. This framework allows for the computationally efficient calculation of the exact log-likelihood of any given input data point. Deep flow architectures, such as Glow, achieve high-fidelity density estimation by intelligently composing invertible convolutions and affine coupling layers [19].

Despite the conceptual elegance of providing exact likelihoods, the application of normalizing flows for anomaly detection faces a theoretical challenge. Recent empirical studies revealed a stark vulnerability in deep generative models: a normalizing flow trained on a complex dataset will routinely assign higher likelihood scores to samples from a significantly simpler, out-of-distribution dataset [31]. This phenomenon occurs because normalizing flows are highly biased toward capturing the low-level, high-frequency statistics of the training data. Researchers explained this paradox through the lens of information theory and the typical set hypothesis, demonstrating that an anomaly

detector relying strictly on raw likelihood will fail if an OOD sample lies nearer to the mode of the distribution than the typical set of the true in-distribution data [30].

2.5 Research Gap and Thesis Contribution

A critical review of the existing literature reveals a distinct dichotomy in the field of runtime safety monitoring. Defense approaches generally fall into one of two extremes: they either evaluate the raw input space, entirely ignoring the high-level semantic features crucial for identifying targeted attacks, or they exclusively monitor deep internal representations, which requires highly restrictive parametric assumptions that frequently fail to capture the true complexity of the feature space. Furthermore, while exact density estimation via normalizing flows provides the most mathematically sound framework for unsupervised anomaly scoring, the literature clearly demonstrates that pixel-only generative models suffer fundamentally from the likelihood paradox. They frequently assign high likelihoods to semantically anomalous inputs simply because those inputs match the low-level background statistics of the training data.

The precise research gap this thesis addresses is the lack of a generative anomaly detection framework that simultaneously leverages the exact likelihood guarantees of normalizing flows and the deep semantic awareness of internal representation monitors. Existing works have not systematically investigated whether the likelihood paradox can be solved by forcing a generative model to evaluate a unified topological space containing both raw pixels and high-level features.

To bridge this gap, this thesis introduces and evaluates a Joint Input-Feature Normalizing Flow architecture. The primary contribution of this work is an extensive, methodical ablation study that extracts semantic representations from various depths of a frozen target classifier, concatenates them with the raw optical input, and computes the exact joint log-likelihood. By directly comparing this novel joint architecture against a standard pixel-only baseline, this research explicitly determines whether fusing low-level and high-level structural information enables exact density estimators to reliably detect visually inconspicuous adversarial attacks without relying on the restrictive parametric bounds present in prior feature-space literature.

3 Theoretical Basis

To rigorously evaluate the efficacy of the proposed runtime monitor, it is mathematically imperative to establish the foundational principles governing the utilized algorithms. This chapter systematically dissects the theoretical basis of the entire experimental pipeline. It begins by formalizing the specific deep Convolutional Neural Network architectures employed as target classifiers, dedicating individual analyses to their unique topological innovations and mathematical operations. Subsequently, it rigorously defines the threat model by deriving the gradient-based optimization strategies used to synthesize adversarial perturbations. The chapter then provides the explicit mathematical formulations for the baseline Out-of-Distribution (OOD) detection methodologies, ensuring each technique is independently defined. Finally, it constructs the theoretical framework of exact density estimation via Normalizing Flows, detailing the specific invertible transformations that enable the Glow architecture to function as a highly effective anomaly detector.

3.1 Deep Neural Network Architectures

Convolutional Neural Networks (CNNs) serve as the fundamental perception engines for modern traffic sign recognition systems. A standard CNN operates by sliding learnable spatial filters across an input image tensor $x \in \mathbb{R}^{C_{in} \times H \times W}$ to produce a highly structured feature map. The discrete convolution operation for a specific spatial coordinate (i, j) in the output feature map, utilizing a local filter W of spatial size $K \times K$, is formally defined as:

$$(x * W)_{i,j} = \sum_{c=1}^{C_{in}} \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} x_{c,i+m,j+n} \cdot W_{c,m,n} + b \quad [3.1]$$

where b is a learnable scalar bias term. Following this linear convolution, a non-linear activation function is applied. In modern architectures, this is almost universally the Rectified Linear Unit (ReLU), mathematically defined as $f(z) = \max(0, z)$. This specific non-linearity allows the network to learn complex, high-dimensional decision boundaries that are otherwise impossible to model with purely affine transformations. To evaluate the proposed normalizing flow monitors across a diverse structural landscape, multiple highly specialized CNN architectures were utilized as target classifiers, each detailed extensively in the following subsections.

3.1.1 Deep Residual Networks (ResNet18)

As neural networks drastically increase in depth to capture more complex semantic abstractions, they inherently suffer from the vanishing gradient problem. This phenomenon occurs when the backpropagated error signals decay exponentially through successive chain-rule multiplications,

rendering the earliest layers of the network causing a severe degradation in gradient flow to the earliest layers. To algorithmically cope with this, He et al. [12] introduced the Residual Network (ResNet) architecture. The core mathematical innovation of ResNet is the residual block, which utilizes identity skip connections to bypass intermediate convolutional layers entirely. Let $\mathscr{H}(x)$ be the desired underlying mapping to be fit by a stacked sequence of convolutional layers. Rather than expecting the network to approximate this complex mapping directly, ResNet explicitly forces these layers to approximate a residual function $\mathscr{F}(x) := \mathscr{H}(x) - x$. The original mapping is then mathematically recast as:

$$y = \text{ReLU}(\mathscr{F}(x, \{W_i\}) + x) \tag{3.2}$$

The critical operation $\mathscr{F}(x, \{W_i\}) + x$ is performed via a shortcut connection and element-wise matrix addition. This simple yet profound mechanism allows gradients to flow directly through the identity skip connections during backpropagation without multiplicative degradation. Consequently, this effectively bypasses the vanishing gradient problem and allows for the training of extraordinarily deep models. ResNet18 is heavily utilized in our evaluation as a standard, high-capacity baseline model, parameterized by 18 sequential layers of deep convolutional blocks.

3.1.2 Densely Connected Convolutional Networks (DenseNet121)

While ResNets utilize additive identity skip connections, DenseNets [15] propose a much more highly dense connectivity pattern to maximize information flow and internal feature reuse throughout the computational graph. In a DenseNet topology, each layer obtains additional input tensors from all preceding layers and subsequently passes on its own computed feature maps to all subsequent layers. Let x_l represent the discrete output of the l-th layer. The l-th layer receives the concatenated feature maps of all preceding layers as its input:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \tag{3.3}$$

where $[x_0, x_1, \ldots, x_{l-1}]$ strictly refers to the algorithmic concatenation of the individual feature maps along the channel dimension. The function H_l typically comprises a Batch Normalization layer, a ReLU activation, and a continuous convolution operation. DenseNet121 utilizes this dense connectivity exclusively within defined operational blocks, separated by transition layers that perform spatial pooling to reduce dimensionality. This specific architecture yields highly dense and feature-rich intermediate representations.

3.1.3 MobileNetV2

In the specific context of autonomous driving, perception models must frequently execute on highly constrained, embedded edge hardware. MobileNetV2 [32] was specifically engineered to drastically reduce the total parameter count and the required Floating Point Operations (FLOPs). The architecture achieves this computational efficiency through Depthwise Separable Convolutions and Inverted Residuals equipped with Linear Bottlenecks.

MobileNet completely factorizes a standard convolution into two distinct steps. First, a depthwise

convolution applies a single spatial filter per input channel. Second, a 1×1 pointwise convolution linearly combines the outputs across the depth dimension. Furthermore, the inverted residual block takes a low-dimensional compressed representation, expands it to a high dimension to perform the depthwise convolution, and then linearly projects it back to a low-dimensional subspace. This projection utilizes a linear bottleneck layer, which is simply a convolutional layer lacking a non-linear activation. Removing the non-linearity in the bottleneck prevents the irreversible destruction of information in low-dimensional subspaces, ensuring stable feature propagation.

3.1.4 EfficientNet B0

EfficientNet B0 [38] represents a highly optimized architecture formulated fundamentally via automated Neural Architecture Search. The EfficientNet paradigm mathematically formalizes the concept of network scaling. Rather than arbitrarily or heuristically increasing depth, width, or input resolution independently, it employs a strict compound scaling method. Let α, β, γ be constants mathematically determined by a small baseline grid search. The network scales its depth by $d = \alpha^{\phi}$, its width by $w = \beta^{\phi}$, and its spatial resolution by $r = \gamma^{\phi}$, subject to the strict computational constraint:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \qquad [3.4]$$

where ϕ is a user-defined coefficient dictating the total available computational resources. This principled mathematical scaling ensures that the baseline EfficientNet B0 model extracts highly optimized, scale-aware feature representations without wasting parameters on redundant convolutional filters.

3.1.5 ShuffleNet V2

ShuffleNet V2 [25] specifically addresses the structural limitations of standard group convolutions. While group convolutions reduce computational cost, they isolate channel groups from one another and strictly prevent cross-group information flow. ShuffleNet rectifies this by introducing a mathematically parameter-free Channel Shuffle operation. Let an intermediate feature map possess $g \times n$ channels, where g is the number of groups. The shuffle operation geometrically reshapes the channels into a two-dimensional (g, n) tensor, transposes it to a (n, g) tensor, and flattens it back into a standard one-dimensional sequence. The foundational building block of ShuffleNet V2 splits the input channels into two parallel branches, processes one branch with computationally light depthwise convolutions, concatenates them back together, and applies the channel shuffle. This continuous reordering elegantly enables comprehensive cross-group communication with absolutely zero additional mathematical operations.

3.1.6 Spatial Transformer Networks (CNN-STN)

Traffic signs observed in the physical world are frequently subjected to severe affine transformations. These include arbitrary rotation, scaling, and heavy shear distortion strictly due to the viewing angle of the approaching vehicular camera. To endow a baseline convolutional network with explicit

spatial invariance, the CNN-STN architecture incorporates a dedicated Spatial Transformer Network [16] module.

The STN consists of three continuous mathematical components: a localization network, a parametric grid generator, and a differentiable sampler. The localization network, typically a small auxiliary CNN, continuously regresses the precise parameters θ of an affine transformation matrix. For a standard two-dimensional affine geometric transformation, θ constructs a 2×3 matrix:

$$A_\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \quad [3.5]$$

The grid generator utilizes this matrix to create a mathematical sampling grid, directly mapping the output pixel coordinates to the input image coordinates via the learned transformation A_θ. Finally, the sampler utilizes differentiable bilinear interpolation to produce the geometrically warped output feature map. By placing an STN prior to the main classification layers, the CNN-STN architecture explicitly normalizes the topological perspective of the traffic sign prior to categorical inference.

3.2 Adversarial Threat Models and Mathematical Optimization

The structural robustness of all the aforementioned deep architectures is severely compromised by the existence of adversarial examples. An adversarial example x' is generated by applying an optimized perturbation δ to a natural image x, explicitly causing the classifier f parameterized by θ to output an incorrect prediction. To ensure this perturbation remains visually undetectable, it is mathematically bounded by an L_p norm constraint:

$$||\delta||_p = \left(\sum_{i=1}^{D} |\delta_i|^p \right)^{\frac{1}{p}} \leq \varepsilon \quad [3.6]$$

In this specific thesis, the empirical evaluation strictly adheres to the L_∞ norm threat model, where the maximum absolute change to any single pixel value is bounded by ε. The standard perturbation budget for images normalized to the continuous range of $[0,1]$ is defined strictly as $\varepsilon = 8/255$, ensuring algorithmic consistency.

3.2.1 Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method [10] operates as a first-order, single-step white-box adversarial attack. The mathematical objective is to maximize the underlying classification loss function $\mathscr{L}(\theta, x+\delta, y)$, where y represents the true ground-truth label. Assuming the loss function exhibits local linearity within the very small ε-ball, it is mathematically approximated using a first-order Taylor series expansion:

$$\mathscr{L}(\theta, x+\delta, y) \approx \mathscr{L}(\theta, x, y) + \delta^T \nabla_x \mathscr{L}(\theta, x, y) \quad [3.7]$$

To effectively maximize this linear approximation subject to the strict L_∞ constraint, the optimal noise vector δ is calculated by isolating the sign of the local gradients. This mathematical derivation yields the highly efficient, closed-form FGSM update rule:

$$x_{FGSM} = x + \varepsilon \cdot \text{sign}(\nabla_x \mathscr{L}(\theta, x, y)) \quad [3.8]$$

While incredibly computationally efficient, FGSM frequently underfits the optimal perturbation because the extreme non-linearity of deep convolutional networks severely violates the assumption of a locally linear loss landscape.

3.2.2 Projected Gradient Descent (PGD)

Projected Gradient Descent [26] treats adversarial generation as a deeply constrained, continuous optimization problem rather than a single-step approximation. Instead of a single massive step, PGD iteratively applies highly minute gradient steps parameterized by a step size α. After each iteration t, the resulting modified image is mathematically projected back into the allowed ε-neighborhood $\mathscr{B}_\varepsilon(x)$ utilizing the precise geometric projection operator Π:

$$x^{(t+1)} = \text{Clip}_{[0,1]}\left(\Pi_{\mathscr{B}_\varepsilon(x)}\left(x^{(t)} + \alpha \cdot \text{sign}(\nabla_x \mathscr{L}(\theta, x^{(t)}, y))\right)\right) \quad [3.9]$$

This iterative formulation guarantees that the perturbation remains strictly bounded while allowing the optimizer to thoroughly navigate the curved loss landscape. PGD is widely regarded in the contemporary literature as the standard benchmark for first-order adversaries.

3.2.3 Auto-Projected Gradient Descent (APGD)

Auto-Projected Gradient Descent [4] dynamically adjusts the algorithmic step size α based on the progression of the loss, eliminating reliance on static hyperparameters. To circumvent the severe gradient vanishing problem completely inherent to the standard Cross-Entropy loss when the softmax function saturates, APGD employs the Difference of Logits Ratio (DLR) loss. Let the sorted pre-softmax logits of the network be represented sequentially as $Z_{\pi_1} \geq Z_{\pi_2} \geq \cdots \geq Z_{\pi_C}$. The DLR loss is formally defined as:

$$\text{DLR}(x, y) = -\frac{Z_y - \max_{i \neq y} Z_i}{Z_{\pi_1} - Z_{\pi_3}} \quad [3.10]$$

This highly specific formulation is completely shift and scale-invariant. Consequently, the DLR loss mathematically ensures non-zero gradients even in extreme confidence failure modes, allowing the optimization algorithm to continue finding adversarial vulnerabilities.

3.3 Out-of-Distribution (OOD) Detection Frameworks

Identifying highly sophisticated adversarial attacks requires rigorous runtime monitors capable of identifying samples that deviate fundamentally from the training manifold. To comprehensively

baseline our generative normalizing flow models against the state of the art, we formally derive the mathematical principles of the specific OOD detection algorithms evaluated in this research.

3.3.1 Maximum Softmax Probability (MSP)

The foundational MSP baseline [13] operates under the theoretical assumption that a well-trained neural network $f(x)$ will exhibit significantly lower categorical confidence when evaluating an unfamiliar OOD sample. The algorithmic anomaly score is directly extracted as the negative maximum class probability from the terminal softmax function:

$$S_{MSP}(x) = - \max_{i \in \{1,\dots,C\}} \frac{\exp(f_i(x))}{\sum_{j=1}^{C} \exp(f_j(x))} \quad [3.11]$$

If this anomaly score surpasses a pre-calculated empirical threshold, the image is officially flagged as out-of-distribution. While highly intuitive, the exponential characteristics of the softmax normalization frequently produce artificially inflated probabilities, rendering this baseline highly susceptible to overconfidence.

3.3.2 ODIN: Temperature Scaling and Input Perturbation

To algorithmically combat the softmax overconfidence, the ODIN framework [23] incorporates Temperature scaling utilizing a large temperature scaling scalar $T > 1$. This parameter mathematically softens the output distribution and amplifies the separability between diverse confidence profiles. Furthermore, ODIN applies a meticulously calculated input perturbation $\tilde{x}$ optimized strictly to maximize the network's maximum softmax probability:

$$\tilde{x} = x - \varepsilon \cdot \text{sign}\left(\nabla_x \log\left(\max_i p_i(x;T)\right)\right) \quad [3.12]$$

Because natural in-distribution samples lie near highly dense, familiar decision boundaries, they algorithmically respond much more strongly to this confidence-maximizing perturbation than heavily corrupted OOD samples. The final ODIN anomaly score is consequently defined as $S_{ODIN}(x) = -\max_i p_i(\tilde{x};T)$.

3.3.3 Energy-Based Detection

Energy-based OOD detection [24] completely abandons the softmax paradigm. Instead, it defines the mathematical free energy $E(x;f)$ for a given input x via the negative log-sum-exp calculation of the network's pre-activation logits:

$$E(x;f) = -T \log \sum_{i=1}^{C} \exp\left(\frac{f_i(x)}{T}\right) \quad [3.13]$$

The anomaly score is directly assigned as the raw energy scalar, where $S_{Energy}(x) = E(x;f)$. This theoretically grounded formulation completely bypasses softmax saturation entirely and provides a

highly smooth mathematical density surface for distinguishing samples.

3.3.4 Mahalanobis Distance in Feature Space

The Mahalanobis feature-space method [22] strictly models the internal representations as class-conditional probability distributions. For an incoming test sample x, the algorithmic distance at a specific intermediate layer l to the closest corresponding class centroid characterized by mean $\mu_{l,c}$ and covariance Σ_l is calculated precisely as:

$$M_l(x) = \max_c -(f_l(x) - \mu_{l,c})^T \Sigma_l^{-1} (f_l(x) - \mu_{l,c}) \quad [3.14]$$

By aggregating these multidimensional distances across all deep layers using a fitted logistic regression model, the system flags samples that geometrically reside far from the center of any known feature distribution.

3.3.5 k-Nearest Neighbors (k-NN) in Embedding Space

A highly potent, non-parametric alternative to the Mahalanobis distance is the k-Nearest Neighbors (k-NN) spatial approach [35]. Let $f(x) \in \mathbb{R}^d$ denote the mathematically normalized, high-dimensional feature embedding of the penultimate layer of the target classifier. Given a massive set of identically projected training embeddings $\mathscr{D}_{train} = \{f(x_1), \ldots, f(x_N)\}$, the k-NN approach efficiently computes the distance from the test sample x to its k-th nearest neighbor. Utilizing the standard Euclidean distance metric, the exact OOD anomaly score is formally defined as:

$$S_{kNN}(x) = ||f(x) - f(x_{(k)})||_2 \quad [3.15]$$

where $f(x_{(k)})$ strictly represents the k-th closest embedding vector present within the saved training manifold. If the test sample is an adversarial image, its embedding will topologically lie far from the dense clusters of the training manifold, yielding a structurally high distance score.

3.3.6 Variational Autoencoder (VAE) Entropy

Variational Autoencoders [20] offer an explicit probabilistic graphical model to facilitate anomaly detection. A standard VAE consists of a probabilistic encoder $q_\phi(z|x)$ and a deterministic decoder $p_\theta(x|z)$. The system optimizes the mathematically derived Evidence Lower Bound (ELBO):

$$\mathscr{L}_{ELBO}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x)||p_\theta(z)) \quad [3.16]$$

For robust OOD detection, VAE Entropy-based methods explicitly and continuously monitor the mathematical uncertainty of the approximate posterior. The differential entropy $H(q_\phi(z|x))$ of the Gaussian encoder outputs, parameterized heavily by mean $\mu(x)$ and variance $\sigma^2(x)$, serves as a

remarkably potent anomaly signal:

$$S_{VAE\_Entropy}(x) = H(q_\phi(z|x)) = \frac{1}{2}\sum_{j=1}^{d}\left(1+\log(2\pi\sigma_j^2(x))\right) \quad [3.17]$$

Adversarial noise frequently induces extreme uncertainty in the latent mapping, causing the exact variance and subsequently the differential entropy of the posterior distribution to spike violently compared to highly predictable benign data.

3.4 Probabilistic Generative Modeling and Normalizing Flows

To detect anomalous adversarial samples without relying on restrictive, rigid parametric assumptions, we strictly employ Normalizing Flows [5]. These models define a highly expressive, completely tractable generative mathematical framework.

3.4.1 The Change of Variables Theorem

A continuous Normalizing Flow learns a completely bijective function $g_\phi : \mathscr{X} \to \mathscr{Z}$, effectively mapping highly dimensional input data x to an isotropic multivariate Gaussian base distribution $p_Z(z) = \mathcal{N}(z;0,I)$. According directly to the fundamental change of variables theorem from calculus, the exact probability density $p_X(x)$ is calculated utilizing the explicit Jacobian matrix $J = \frac{\partial g_\phi(x)}{\partial x}$:

$$p_X(x) = p_Z(g_\phi(x))\left|\det\left(\frac{\partial g_\phi(x)}{\partial x}\right)\right| \quad [3.18]$$

For a deeply nested sequence of K distinct transformations formulated as $g_\phi = f_K \circ \cdots \circ f_1$, the log-likelihood becomes highly tractable through summation:

$$\log p_X(x) = \log p_Z(g_\phi(x)) + \sum_{i=1}^{K}\log\left|\det\left(\frac{\partial f_i(h_{i-1})}{\partial h_{i-1}}\right)\right| \quad [3.19]$$

This mathematically elegant formulation ensures that every single transformation contributes to the overall exact volume change, maintaining perfect likelihood tractability.

3.4.2 Affine Coupling Layers

The critical affine coupling layer meticulously splits the input tensor h exactly in half, resulting in two partitions h_a and h_b. An arbitrary, non-invertible neural network strictly computes mathematical scale s and translation t variables exclusively based on h_a, systematically applying them to h_b:

$$h_a^{(out)} = h_a \quad [3.20]$$

$$h_b^{(out)} = h_b \odot \exp(s(h_a)) + t(h_a) \quad [3.21]$$

Because $h_a^{(out)}$ is entirely mathematically independent of h_b, the resulting Jacobian J is guaranteed to be strictly lower-triangular. This architectural design ensures the computationally prohibitive

determinant calculation is elegantly reduced to the linear mathematical sum of the diagonal elements.

3.4.3 The Glow Architecture and Anomaly Scoring

Glow [19] heavily optimizes specific flow mechanics for high-dimensional visual data through three highly distinct operations: Activation Normalization (ActNorm) for continuous stable scaling, Invertible 1×1 Convolutions, and Affine Coupling layers. The invertible convolutions are strictly parameterized by their PLU decomposition $W = PL(U + \text{diag}(s))$ to mathematically guarantee a highly tractable $\mathcal{O}(C)$ log-determinant computation.

During real-time runtime inference, the trained generative flow calculates the exact Negative Log-Likelihood (NLL). Because adversarial perturbations explicitly force the image off the complex natural data manifold, they mathematically map to the extreme low-density tails of the Gaussian latent space $p_Z(z)$. The ultimate algorithmic anomaly score is directly and unyieldingly assigned as $\mathscr{A}(x) = -\log p_X(x)$. This specific formulation completely bypasses all structural vulnerabilities associated with softmax scaling, yielding a robust safety threshold suitable for robust autonomous deployment.

4 Methodology and Architecture

The central objective of this research is to engineer a mathematically rigorous runtime safety monitor capable of identifying visually inconspicuous adversarial perturbations and out-of-distribution (OOD) data without altering the pre-trained weights of the target perception model. This chapter meticulously details the architectural design and the explicit mathematical implementation of the proposed generative monitoring framework. It formally introduces the two distinct operational paradigms developed for this thesis: the Input-Only flow architecture and the novel Joint Input-Feature flow architecture. Furthermore, this chapter provides a granular mathematical breakdown of the layerwise feature extraction mechanism, the spatial alignment and feature normalization protocols, the multiscale Glow-based Normalizing Flow topology, the exact log-likelihood loss function, and the gradient accumulation dynamics necessitated by hardware constraints during the optimization of high-dimensional joint tensors. Finally, it provides rigorous mathematical proofs of invertibility to guarantee the theoretical soundness of the developed system.

4.1 Architectural Overview of the Parallel Monitor

The proposed runtime safety monitor is engineered to operate strictly in parallel with a deployed, fully trained Convolutional Neural Network (CNN) tasked with traffic sign recognition. The fundamental topological principle of this system is absolute non-interference. The target classifier acts as a frozen, deterministic, and immutable function $f : \mathscr{X} \to \mathscr{Y}$. This non-invasive design guarantees that the carefully optimized baseline classification accuracy of the perception module on clean, unperturbed data is flawlessly preserved, circumventing the severe robustness-accuracy trade-offs typically induced by proactive defenses such as adversarial training.

During the standard operational inference phase within an autonomous driving perception pipeline, an incoming optical image tensor $x \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ is processed by the target classifier to yield a categorical prediction vector. Concurrently, the identical input tensor x is routed into the parallel generative runtime monitor. The monitor functions as a secondary, mathematically independent exact density estimator, parameterized by a set of learnable weights ϕ. Its singular task is to calculate the exact negative log-likelihood of the input sample under a highly complex probability distribution learned exclusively from benign, natural traffic signs. If this mathematically evaluated anomaly score surpasses a rigorously defined statistical threshold τ, the system immediately raises a safety flag, preempting and overriding the potentially compromised categorical prediction of the target classifier.

To comprehensively investigate the optimal input space topology for generative density estimation, this thesis explicitly develops two divergent architectural pathways for the runtime monitor. The first pathway, designated as the Input-Only architecture, directly models the probability density function

of the raw pixel space x. The second pathway, designated as the Joint Input-Feature architecture, leverages the hierarchical processing power of the frozen classifier by extracting deep semantic representations from its hidden layers, spatially aligning them with the raw image, and jointly modeling the concatenated multidimensional topological space. This comparative structural design forms the core of the ablation study evaluated in subsequent chapters.

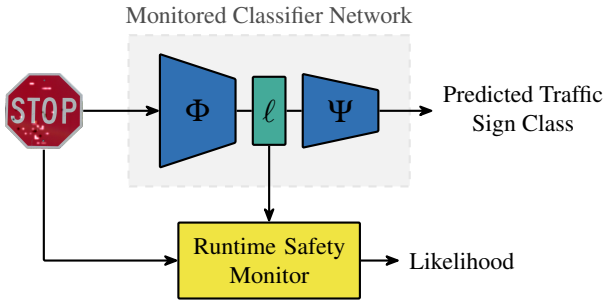


Figure 4.1: High-level architectural diagram illustrating the parallel execution of the frozen target classifier and the generative runtime monitor. The diagram explicitly depicts the bifurcation of the input image, the feed-forward pass of the CNN, and the optional extraction of intermediate feature maps integrated into the joint density modeling configuration.

4.2 Layerwise Feature Extraction Mechanism

The foundational algorithmic prerequisite for the Joint Input-Feature architecture is the non-destructive extraction of intermediate semantic representations from the target classifier. Because the target network's computational graph must remain absolutely pristine and its parameters θ strictly frozen (enforcing $\nabla_\theta = 0$ throughout all operations), we engineered a highly specific, hook-based interceptive logic that operates seamlessly within the deep learning framework's execution engine.

4.2.1 Topological Interception and Computational Graph Hooks

Let the frozen target classifier f be mathematically represented as a sequential composition of L distinct operational blocks or layers, such that the final logit output for an input x is defined recursively as $f(x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x)$. The hierarchical nature of deep convolutional networks dictates a progression of semantic abstraction. Earlier layers ($l \ll L$) predominantly encode low-level, high-frequency spatial features such as geometric edges, color gradients, and raw textures. Conversely, deeper layers ($l \approx L$) encode highly abstract, spatially pooled semantic representations that correlate strongly with specific object classes.

To extract these continuous representations at an arbitrary, user-defined intermediate index $l \in \{1, \ldots, L-1\}$, the system dynamically registers a forward-hook callback mechanism within the directed acyclic computational graph of the model. During the forward pass of the image tensor x, the interceptive hook is algorithmically triggered precisely upon the completion of the l-th layer's forward mathematical computation, prior to the tensor being passed to layer $l+1$.

The intermediate activation tensor is instantaneously intercepted and cloned into an isolated, disjoint memory buffer. This cloning operation entirely decouples the extracted tensor from the

target network's gradient tracking tape, ensuring that no anomalous computational overhead or backward graph dependencies are accidentally introduced into the perception module. We formally denote this extracted intermediate feature tensor as A_l:

$$A_l = f_l(f_{l-1}(\dots f_1(x)\dots)) \quad [4.1]$$

The extracted tensor $A_l \in \mathbb{R}^{C_l \times H_l \times W_l}$ possesses a channel depth C_l mathematically determined by the specific filter configuration of layer l, and spatial dimensions H_l and W_l. Depending on the chosen architecture, such as ResNet18 or MobileNetV2, the magnitude of C_l can range from relatively shallow dimensions like 64 channels to massively deep semantic spaces containing 512, 1024 or 1280 channels.

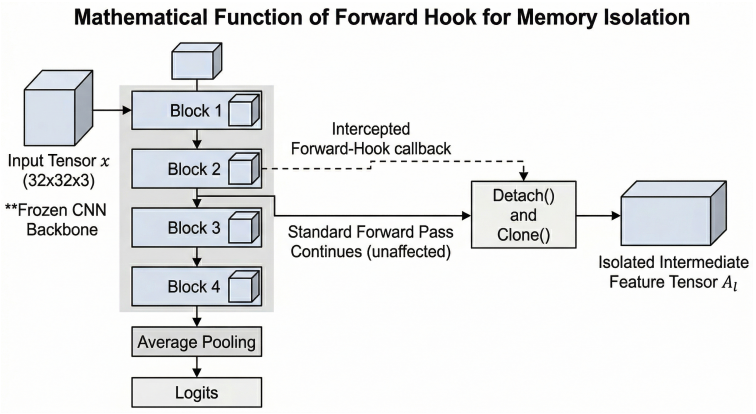


Figure 4.2: Schematic representation of the layerwise feature extraction mechanism. The diagram highlights the forward-hook interception point within a standard Convolutional Neural Network backbone, illustrating the generation of the isolated feature tensor A_l parallel to the continued forward pass.

4.3 Spatial Alignment and Feature Normalization

To mathematically model the joint probability distribution of the raw input image and the extracted semantic feature map, the two highly disparate tensors must be algorithmically unified into a cohesive, topologically consistent mathematical structure before being processed by the normalizing flow. This precise unification requires two critical, sequential preprocessing stages: exact spatial alignment and rigorous mathematical feature normalization.

4.3.1 Bilinear Spatial Interpolation for Dimensionality Matching

Normalizing flows intrinsically rely on strict spatial symmetries to perform operations such as invertible 1×1 convolutions and multi-scale spatial squeezing. However, due to the ubiquitous presence of strided convolutions, max-pooling, and average-pooling operations within the target classifier's architecture, the spatial dimensions of the extracted intermediate feature map are virtually guaranteed to be significantly smaller than the original input image. Mathematically, this dictates that $H_l \ll H_{in}$ and $W_l \ll W_{in}$.

To completely rectify this severe dimensional mismatch, the intermediate feature map A_l must be upsampled to precisely match the strict spatial resolution of the raw input image x. We employ deterministic bilinear interpolation, a parameter-free continuous geometric transformation that preserves the exact mathematical invertibility required for the overall generative framework. For a target continuous coordinate (i, j) in the upsampled feature space, corresponding to a fractional continuous coordinate (u, v) mathematically projected onto the original feature map A_l, the interpolated pixel value is computed via the mathematically weighted sum of its four nearest spatial neighbors.

Let $u_0 = \lfloor u \rfloor$ and $v_0 = \lfloor v \rfloor$ be the floor integer coordinates, and let $\Delta u = u - u_0$ and $\Delta v = v - v_0$ be the fractional distances. The bilinear interpolation for a specific channel c is rigorously defined as:

$$\begin{aligned}\tilde{A}_{l,c}(i,j) =& (1-\Delta u)(1-\Delta v)A_{l,c}(u_0, v_0) \\ &+ \Delta u(1-\Delta v)A_{l,c}(u_0+1, v_0) \\ &+ (1-\Delta u)\Delta v A_{l,c}(u_0, v_0+1) \\ &+ \Delta u \Delta v A_{l,c}(u_0+1, v_0+1)\end{aligned} \quad [4.2]$$

This deterministic linear operation yields the spatially aligned feature tensor $\tilde{A}_l \in \mathbb{R}^{C_l \times H_{in} \times W_{in}}$, ensuring perfectly symmetrical spatial dimensions with the input image. This alignment is critical for the subsequent cross-channel fusion steps.

4.3.2 Channel-wise Mathematical Feature Normalization

Following spatial alignment, a significant numerical discrepancy exists between the raw image space and the deep semantic feature space. The raw input image x is strictly normalized via preprocessing such that its pixel intensities uniformly reside within a highly constrained domain, typically $[0, 1]$ or standardized to mean zero and variance one. In stark contrast, the intermediate deep activations $\tilde{A}_l$, particularly those extracted immediately subsequent to an unbounded non-linear activation function such as a Rectified Linear Unit (ReLU), are completely unbounded in the positive mathematical direction ($[0, \infty)$) and frequently exhibit extreme numerical variance across different channels.

If this highly unnormalized feature tensor is directly concatenated and fed into the complex optimization landscape of the Glow architecture, the massive numerical magnitudes of the semantic features will exponentially dominate the gradient updates during backpropagation. This imbalance rapidly induces severe numerical instability, saturated gradients within the context networks of the affine coupling layers, and catastrophic initialization failure of the Activation Normalization (ActNorm) modules. To explicitly secure computational stability and ensure uniform gradient flow, rigorous channel-wise mathematical normalization is enforced upon the interpolated feature map.

Let μ_c and σ_c^2 represent the empirically calculated mean and variance for a specific channel c, derived globally across the entire benign training dataset prior to flow optimization. The mathematically standardized feature tensor $\hat{A}_l$ is computed as:

$$\hat{A}_{l,c}(i,j) = \frac{\tilde{A}_{l,c}(i,j) - \mu_c}{\sqrt{\sigma_c^2 + \varepsilon_{norm}}} \quad [4.3]$$

where ε_{norm} is a strictly defined microscopic constant (such as 10^{-6}) algorithmically added to the denominator to prevent a division by zero in zero-variance channels, which are frequently caused by dead ReLU activations. By mathematically standardizing the semantic features, the generative model is provided with a numerically stable landscape to efficiently learn the highly complex joint covariance structure without facing gradient explosion.

Following rigorous normalization, the unified joint topological tensor u is constructed via strict channel-wise concatenation of the raw image and the standardized features:

$$u = [x \parallel \hat{A}_l] \quad \in \mathbb{R}^{(C_{in}+C_l)\times H_{in}\times W_{in}} \tag{4.4}$$

where the double vertical bar symbol denotes the exact concatenation operator along the channel dimension. In the baseline Input-Only architectural configuration, these complex spatial alignment and normalization steps are intentionally bypassed, and the operational data tensor is mathematically defined simply as $u = x$.

4.4 The Glow-Based Normalizing Flow Architecture

The foundational generative engine driving the proposed runtime anomaly monitor is a deep, multi-scale Normalizing Flow explicitly constructed utilizing the highly optimized Glow architecture. The overarching mathematical objective of the flow is to learn a highly complex, heavily parameterized, yet strictly bijective mapping $g_\phi : \mathscr{U} \to \mathscr{Z}$. Here, $\mathscr{U}$ denotes the massively high-dimensional data space of the joint tensor u, and $\mathscr{Z}$ denotes a continuous, analytically tractable latent space completely governed by an isotropic multivariate Gaussian base distribution $p_Z(z) = \mathscr{N}(z; 0, I)$.

4.4.1 Multi-Scale Squeeze and Split Topology

To process the massive channel depth inherent to the joint tensor u, which frequently exceeds several hundred channels when fusing an RGB image with deep layer semantic feature maps, the flow utilizes a highly specific multi-scale factoring topology. Processing high-resolution tensors with hundreds of channels through deep convolutional context networks is computationally prohibitive without this architectural innovation.

The architecture is structurally divided into a sequence of M macro-blocks. At the commencement of each macro-block, a deterministic squeeze operation is applied. The squeeze operation mathematically trades spatial resolution for channel depth by reshaping spatial $2 \times 2 \times C$ neighborhoods into $1 \times 1 \times 4C$ channel vectors. For a tensor of shape $C \times H \times W$, the squeezed tensor inherently assumes the shape $4C \times \frac{H}{2} \times \frac{W}{2}$. This geometric reduction vastly expands the receptive field of the subsequent convolutions within the context networks without demanding deeper architectural layers.

Following a defined series of sequential flow steps within the macro-block, a mathematically critical split operation is executed. Exactly half of the latent channels are factored out of the main execution pipeline and evaluated directly against the Gaussian base distribution. If the tensor entering the split operation has shape $C' \times H' \times W'$, the operation partitions it into two tensors, z_i and

h_{next}, each of shape $\frac{C'}{2} \times H' \times W'$. The tensor z_i is explicitly added to the final latent representation z, while h_{next} proceeds to the next macro-block. This multi-scale design fundamentally mitigates the computational and memory burden while explicitly allowing the generative model to capture hierarchical, multi-resolution statistical dependencies. Within these macro-blocks, the bijective transformation relies strictly on three sequentially executed, mathematically invertible operations.

4.4.2 Activation Normalization (ActNorm)

Standard batch normalization introduces severe batch-dependent statistical noise that systematically violates the strict exactness requirement of normalizing flow likelihood estimation. To provide robust, mathematically invertible activation stabilization, the architecture utilizes Activation Normalization (ActNorm).

ActNorm performs an independent, spatial-agnostic affine transformation utilizing learnable scale vectors $s_{act} \in \mathbb{R}^C$ and bias vectors $b_{act} \in \mathbb{R}^C$, applied uniquely to each channel c:

$$h_{c,i,j}^{(out)} = h_{c,i,j} \cdot \exp(s_{act,c}) + b_{act,c} \quad [4.5]$$

Because the affine operation is applied strictly element-wise across the spatial dimensions H and W, its corresponding Jacobian matrix is a pure diagonal matrix. The exact log-determinant is trivially and efficiently calculated as the spatial area multiplied by the sum of the scale parameters:

$$\log \left| \det \left(\frac{\partial h^{(out)}}{\partial h} \right) \right| = H \cdot W \cdot \sum_{c=1}^{C} s_{act,c} \quad [4.6]$$

Critically, to guarantee optimal training convergence, these parameters are not initialized randomly. They are dynamically initialized using the exact mean and variance of the very first training micro-batch. This initialization mathematically ensures zero mean and unit variance for the activations immediately following the initialization step, serving as an advanced, perfectly invertible data-dependent initialization mechanism.

4.4.3 Invertible 1×1 Convolutions

The succeeding affine coupling layers are mathematically designed to divide the channels strictly in half, permanently prohibiting cross-channel interaction between the two disjoint partitions. To ensure that the highly disparate statistical structures of the raw image channels and the extracted feature channels are thoroughly fused and permuted prior to the coupling split, an invertible 1×1 convolution is applied.

A fully learned weight matrix $W \in \mathbb{R}^{C \times C}$ mathematically transforms the entire channel vector at each specific spatial coordinate (i, j). The exact log-determinant of this step is defined as $H \cdot W \cdot \log |\det W|$. Computing the determinant of a $C \times C$ matrix under normal linear algebraic circumstances requires $\mathcal{O}(C^3)$ floating-point operations. To dramatically optimize this and render it feasible for joint tensors with massive channel depths, the Glow architecture strictly parameterizes

the weight matrix W utilizing its PLU decomposition:

$$W = PL(U + \text{diag}(s_{conv})) \tag{4.7}$$

where P is a static, non-learnable permutation matrix initialized randomly, L is a learnable lower triangular matrix strictly possessing ones on its diagonal, U is a learnable upper triangular matrix strictly possessing zeros on its diagonal, and s_{conv} is a learnable vector representing the diagonal elements. Due to the multiplicative property of determinants for strict triangular matrices, the computationally intensive log-determinant simplifies to a highly efficient linear summation:

$$\log \left| \det \left(\frac{\partial h^{(out)}}{\partial h} \right) \right| = H \cdot W \cdot \sum_{c=1}^{C} \log |s_{conv,c}| \tag{4.8}$$

This linear reduction is what makes the training of such high-dimensional spaces physically possible.

4.4.4 Affine Coupling Layers and Context Networks

The paramount non-linear topological transformation within the flow architecture is the affine coupling layer. The input tensor h is mathematically split along the channel axis into two perfectly disjoint halves, h_a and h_b. The first partition h_a is passed through a highly parameterized context network, denoted as $\text{NN}(h_a)$, to yield continuous scale factors s and translation factors t. These mathematical factors are then utilized to apply an affine transformation exclusively to the second partition, h_b:

$$h_a^{(out)} = h_a \tag{4.9}$$

$$h_b^{(out)} = h_b \odot \sigma(s) + t \tag{4.10}$$

where σ designates a sigmoid activation function mathematically scaled to ensure strict numerical stability for the scaling factor, preventing extreme exponential expansion. The context network NN fundamentally does not need to be invertible itself, as its outputs are only used to shift and scale h_b. This allows the architecture to utilize powerful, standard convolutional layers.

In the developed implementation, NN is constructed as a three-layer shallow Convolutional Neural Network utilizing standard ReLU activations. The final convolutional layer of NN is strictly initialized with zero weights. This zero-initialization mathematically guarantees that at the onset of training, s and t evaluate perfectly to zero, meaning the entire affine coupling block initially acts as an exact identity function, highly stabilizing the generative learning process. Because $h_a^{(out)}$ is entirely independent of h_b, the Jacobian matrix J of this split transformation is strictly lower-triangular. Consequently, the log-determinant is flawlessly and efficiently computed exclusively from the diagonal elements as $\sum \log(\sigma(s))$.

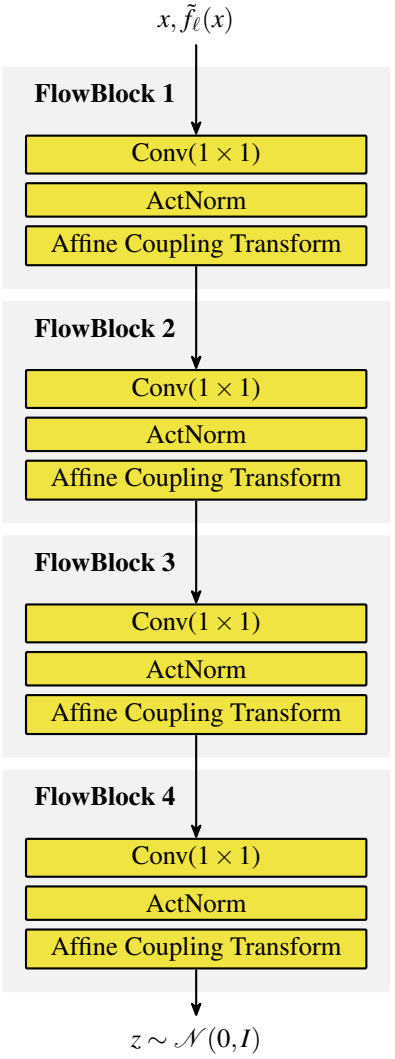


Figure 4.3: Detailed architectural diagram of a single discrete step within the multi-scale Glow Normalizing Flow. The diagram strictly illustrates the sequential execution of the ActNorm layer, the PLU-parameterized Invertible 1×1 Convolution, and the intricate split-transform-concatenate mechanism of the Affine Coupling layer.

4.5 Optimization, Exact Loss Formulation, and Multi-Batching

The operational efficacy of the runtime monitor hinges entirely on the robust mathematical optimization of the flow parameters ϕ during the training phase. The generative monitor is trained exclusively and strictly on the clean, unperturbed training split of the respective traffic sign datasets. Absolutely no adversarial examples or out-of-distribution samples are utilized to inform the parameter weights.

4.5.1 Exact Log-Likelihood Loss Formulation

The explicit mathematical objective is to strictly maximize the expected log-likelihood of the clean joint tensors. Let $\mathscr{D}_{clean}$ denote the true empirical probability distribution of benign data. Utilizing the change-of-variables formulation established in the theoretical basis, the exact empirical loss function to be minimized is the Negative Log-Likelihood (NLL) calculated over a mini-batch of N input tensors $\{u_1, \ldots, u_N\}$:

$$\mathscr{L}(\phi) = -\frac{1}{N}\sum_{i=1}^{N}\left(\log p_Z(g_\phi(u_i)) + \log\left|\det\left(\frac{\partial g_\phi(u_i)}{\partial u_i}\right)\right|\right) \tag{4.11}$$

where $g_\phi(u_i)$ is the final forward pass mapping the data into the latent space z_i, and the determinant term represents the sum of log-determinants from all ActNorm, 1×1 Convolution, and Affine Coupling layers across all macro-blocks. The term $\log p_Z(z)$ rigorously evaluates the exact probability density of the output latent vector under the standard isotropic multivariate normal Gaussian prior:

$$\log p_Z(z_i) = \sum_{j=1}^{D}\left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}z_{i,j}^2\right) \tag{4.12}$$

By continuously minimizing this mathematically exact loss function utilizing the Adam optimization algorithm, the deep neural parameters ϕ geometrically and topologically warp the highly complex, entangled data manifold into the high-density epicenter of the Gaussian prior.

4.5.2 Hardware Constraints and Algorithmic Gradient Accumulation

A critical, severe hardware constraint emerges when optimizing the Joint Input-Feature architecture. The massive channel dimensions resulting from feature concatenation aggressively exhaust the Graphics Processing Unit Video RAM during the backpropagation step. The deep chain of intermediate states required to compute gradients for the context networks pushes memory utilization to its absolute physical limits. Consequently, the maximum physically permissible batch size $N_{physical}$ frequently drops to severely low numbers, such as 4 or 8 tensor images per forward pass.

Normalizing flows are notoriously unstable when optimized with extremely small batch sizes. Specifically, the data-dependent initialization of the ActNorm modules and the variance statistics tracking within the coupling context networks require large sample sizes to induce numerical divergence and parameter explosion. To mathematically simulate a vast batch size and completely restore optimization stability without physically exceeding the strict VRAM memory constraints,

the system explicitly implements algorithmic Gradient Accumulation, herein referred to as multi-batching.

Let N_{target} be the mathematically optimal algorithmic batch size, such as 64 or 128, which fundamentally requires an accumulation factor of $K = \lceil N_{target}/N_{physical} \rceil$. During the optimization loop, the system executes K distinct, mathematically independent forward and backward passes using the highly constrained physical micro-batches. During these passes, the gradient vectors $\nabla_\phi \mathscr{L}_k$ are iteratively accumulated and stored in memory, strictly without updating the actual weights ϕ. The exact mathematical parameter update rule is formally deferred until all K micro-batches have been successfully processed:

$$\phi_{t+1} = \phi_t - \eta \cdot \text{Adam}\left(\frac{1}{K}\sum_{k=1}^{K} \nabla_\phi \mathscr{L}_k(\phi_t)\right) \quad [4.13]$$

where η is the scheduled learning rate. This sophisticated multi-batching paradigm ensures mathematically identical gradient trajectory steps to a theoretically unconstrained hardware environment, enabling the robust, highly stable convergence of the massively parameterized joint density estimator.

4.6 Inference Pipeline and Mathematical Anomaly Scoring

During the operational inference phase deployed within the perception pipeline, the parameters ϕ of the normalizing flow are strictly frozen alongside the target classifier's weights θ. When a novel, unseen, and potentially adversarial input optical sample x^* is intercepted by the vehicle's optical sensors, the runtime monitor executes a single deterministic forward evaluation.

The system performs the necessary intermediate feature extraction and rigorous spatial concatenation to successfully construct the inference tensor u^*. This multi-dimensional tensor is seamlessly processed by the generative Glow model to continuously compute its exact negative log-likelihood. The final, continuous scalar anomaly score $\mathscr{A}(x^*)$ assigned to the test input is directly and formally defined as this exact negative log-likelihood metric:

$$\mathscr{A}(x^*) = -\log p_U(u^*) \quad [4.14]$$

Because targeted adversarial perturbations are mathematically optimized by white-box attackers to forcibly cross the classifier's high-dimensional linear decision boundaries, they forcefully shift the input geometry away from the densely populated, natural image manifold learned by the flow. Consequently, these severely perturbed adversarial tensors u^* inevitably and mathematically map to the extreme low-density tails of the latent Gaussian space $p_Z(z)$, yielding a drastically higher NLL value. By directly thresholding this continuous exact likelihood score, the parallel runtime monitor effectively detects and safely flags both targeted digital attacks and general environmental semantic distribution shifts, fulfilling its core architectural mandate.

4.7 Algorithmic Implementation and Framework Specifics

Transitioning the theoretical mathematical framework of the Joint Input-Feature normalizing flow into a highly optimized, executable software architecture requires rigorous attention to computational graphs and memory management. The entire runtime monitor pipeline is engineered within the PyTorch deep learning framework, leveraging its dynamic computation graph (Autograd) to facilitate the complex, multi-stage gradient calculations required by exact density estimation.

4.7.1 Dynamic Hook Management and Memory Isolation

As previously defined, the extraction of intermediate semantic features from the frozen target classifier relies on forward-hook interception. In a naive software implementation, retaining references to intermediate activation tensors inherently prevents the deep learning framework's garbage collector from freeing the computational graph of the target classifier. This oversight would lead to a severe memory leak during continuous inference or training operations.

To strictly prevent this computational hazard, the interceptive hook is algorithmically designed to perform a deep, detached clone of the activation tensor the exact microsecond it is generated within the network. Let the raw activation produced by the l-th layer be denoted as A_{raw}. The hook immediately executes the specific tensor operation $A_l = A_{raw}$.detach().clone(). The detach operation completely severs the tensor from the target classifier's gradient tape, mathematically guaranteeing that the gradients for all classifier parameters evaluate strictly to zero. The clone operation guarantees that the generative monitor operates on a completely independent physical memory block. This isolation allows the target classifier to aggressively free its own intermediate states as the forward pass continues toward the terminal logits, maintaining optimal memory efficiency.

4.7.2 Invertible Module Instantiation

Within the multi-scale Glow architecture, the physical instantiation of the sequential invertible modules requires highly precise parameter initialization. The context network NN within the affine coupling layers is constructed using a sequence of three 3×3 two-dimensional convolutional layers. To preserve the strict spatial resolution of the partitioned tensor h_a, symmetric zero-padding of $p = 1$ is strictly enforced across all convolutional operations.

The first two convolutional layers utilize a hidden channel depth of 512, followed immediately by standard Rectified Linear Unit (ReLU) activations to introduce necessary non-linearity. The terminal convolutional layer, which directly outputs the concatenated scale s and translation t parameters, is initialized with weights drawn from a Gaussian distribution with a variance artificially scaled down to essentially zero, such as $\mathcal{N}(0, 10^{-10})$. This zero-initialization strategy ensures that the initial forward pass through the unoptimized flow acts perfectly as an identity mapping, actively preventing the massive numerical explosion that would otherwise occur when exponentially scaling the unbounded intermediate feature tensors.

4.8 Mathematical Proofs of Invertibility and Tractability

The fundamental viability of the proposed runtime monitor relies entirely on the absolute mathematical guarantee that the learned generative transformation g_ϕ is strictly bijective and possesses a computationally tractable Jacobian determinant. We hereby formalize the mathematical proofs for the core architectural components.

4.8.1 Invertibility of the Affine Coupling Layer

Let the forward pass of the affine coupling layer be defined by the partition of the input tensor h into h_a and h_b, where $h_a^{(out)} = h_a$ and $h_b^{(out)} = h_b \odot \sigma(s(h_a)) + t(h_a)$. To prove strict bijectivity, we must uniquely and exactly recover the input partitions (h_a, h_b) given strictly the output partitions $(h_a^{(out)}, h_b^{(out)})$.

By direct algebraic substitution, the recovery of the first partition is trivial: $h_a = h_a^{(out)}$. Because the context network NN relies exclusively on h_a to compute the scale and translation factors, and because h_a is perfectly and losslessly recovered, we can deterministically recompute the exact factors $s(h_a)$ and $t(h_a)$ during the reverse execution pass. Consequently, the second partition is mathematically recovered through the exact inverse affine transformation:

$$h_b = \left(h_b^{(out)} - t(h_a^{(out)})\right) \oslash \sigma(s(h_a^{(out)})) \qquad [4.15]$$

where the symbol $\oslash$ denotes element-wise mathematical division. Because the sigmoid function σ guarantees that the scale factor is strictly positive and non-zero across its entire range ($\sigma(z) \in (0,1)$), the element-wise division is mathematically guaranteed to be defined safely across the entire continuous domain. This conclusively proves the absolute invertibility of the coupling layer, regardless of the underlying complexity of the internal context network.

4.8.2 Tractability of the PLU-Parameterized Convolution

The invertible 1×1 convolution applies a dense weight matrix W to the channel dimension. The forward mathematical operation is $h^{(out)} = Wh$. The inverse is simply defined as $h = W^{-1}h^{(out)}$. However, computing the required term $\log|\det W|$ naively requires an intractable $\mathcal{O}(C^3)$ operations. By explicitly enforcing the parameterization $W = PL(U + \text{diag}(s))$, we leverage the fundamental mathematical properties of matrix determinants.

The determinant of a product of multiple square matrices is exactly the product of their individual scalar determinants: $\det(W) = \det(P)\det(L)\det(U + \text{diag}(s))$. The permutation matrix P is orthogonal by definition; hence its determinant is strictly ± 1, and its absolute determinant is always exactly 1. The matrix L is strictly lower-triangular with a diagonal entirely composed of ones. The determinant of any triangular matrix is the mathematical product of its diagonal, meaning $\det(L) = 1$. The matrix $(U + \text{diag}(s))$ is strictly upper-triangular with the vector s comprising its diagonal elements. Therefore, its determinant is simply the product of the individual elements

within s. Combining these absolute proofs yields the highly tractable exact log-determinant:

$$\log|\det W| = \log(\prod_{c=1}^{C}|s_c|) = \sum_{c=1}^{C}\log|s_c| \quad [4.16]$$

This rigorous mathematical reduction is exactly what physically enables the Joint Input-Feature architecture to process hundreds of concatenated semantic channels without strongly stalling the generative training loop.

4.9 Algorithmic Formalization of Joint Training

To completely eliminate ambiguity regarding the optimization dynamics necessitated by severe hardware constraints, the multi-batching training pipeline for the Joint Input-Feature architecture is algorithmically formalized. The necessity to mathematically simulate large batch sizes (N_{target}) using highly constrained physical micro-batches ($N_{physical}$) requires a meticulous management of the PyTorch gradient accumulation tape.

Let the total empirical training dataset be denoted as $\mathscr{D}_{clean}$. Let the accumulation factor be mathematically defined as $K = \lceil N_{target}/N_{physical} \rceil$. During each sequential training epoch, the dataset is randomly shuffled to ensure stochasticity and partitioned into distinct physical micro-batches B_k. The optimization loop proceeds deliberately as follows. The optimizer's internal gradient buffers are explicitly zeroed to clear prior gradients. For $k = 1$ to K, a physical micro-batch of raw images x_k is drawn sequentially from $\mathscr{D}_{clean}$. The frozen target classifier evaluates x_k, and the dynamic hook intercepts the defined intermediate tensor, which is immediately detached and cloned to safely form A_k.

The spatial alignment protocol then bilinearly upsamples A_k to match the spatial resolution of x_k, followed immediately by the strict channel-wise standardization utilizing the globally tracked dataset mean μ and variance σ^2. The disparate tensors are concatenated to cleanly form the joint input $u_k = [x_k \parallel \hat{A}_k]$. The normalizing flow g_ϕ processes u_k, calculating the exact latent representation z_k and accumulating the log-determinants of the Jacobians at every flow step.

The exact negative log-likelihood loss $\mathscr{L}_k$ is mathematically computed. Crucially, to ensure unbiased gradient summation, this loss is mathematically scaled by $\frac{1}{K}$ before the backward pass is invoked, yielding $\mathscr{L}_k^{(scaled)} = \frac{\mathscr{L}_k}{K}$. The Autograd engine computes $\nabla_\phi \mathscr{L}_k^{(scaled)}$ and accumulates these specific gradients into the parameter '.grad' attributes without stepping the optimizer. The memory utilized by the computational graph for micro-batch k is explicitely freed. This sequential, tightly controlled process iterates until all K micro-batches are processed. Finally, the Adam optimizer utilizes the fully accumulated gradients to perform a single, highly stabilized step, updating the flow parameters $\phi_{t+1} \leftarrow \phi_t$. This precise algorithmic flow completely bypasses the hardware-induced instability of generative continuous density estimation.

5 Experimental Setup and Evaluation

The theoretical formulations established in the preceding chapters hypothesize that exact density estimation via normalizing flows can provide a rigorously grounded, unsupervised runtime safety monitor for autonomous perception systems. This chapter transitions from mathematical theory to empirical validation. It presents the comprehensive experimental setup and the subsequent evaluation of the proposed monitoring architectures. The chapter begins by formalizing the physical hardware environment, the optimization hyperparameters, and the specific training setup utilized to converge the deep generative models. Subsequently, it rigorously defines the evaluation metrics, including the mathematical computation of operational thresholds, and provides an exhaustive morphological analysis of the distinct traffic sign datasets. Finally, the chapter unpacks the core contribution of this thesis. It details the extensive ablation study explicitly comparing the detection efficacy and computational overhead (measured in both FLOPs and real-time inference latency) of the Input-Only flow architecture against the Joint Input-Feature flow architecture.

5.1 Experimental Environment and Training Setup

Training high-dimensional normalizing flows, particularly those tasked with modeling the complex joint probability distribution of raw pixels and deep semantic feature maps, requires a highly optimized and computationally robust hardware and software environment. The exact configuration of this environment dictates the convergence stability of the exact density estimator.

5.1.1 Hardware Specifications and Memory Management

All experiments, encompassing the training of the baseline target classifiers, the synthesis of the iterative adversarial perturbations, and the optimization of the multiscale Glow architectures, were executed on a dedicated high-performance computing cluster. The primary computational acceleration was provided by NVIDIA graphics processing units. Specifically, the training pipelines were executed utilizing high-VRAM architectures, such as the NVIDIA RTX 4090 and H100 Tensor Core GPUs, providing 24 GB and 80 GB of Video RAM respectively.

This massive memory capacity is a strict hardware requirement for the Joint Input-Feature architecture. Concatenating hundreds of extracted semantic channels alongside the raw image tensor exponentially inflates the memory footprint of the computational graph during the backpropagation of the exact log-likelihood loss. The central processing unit handled the asynchronous, multithreaded data loading and real-time bilinear spatial interpolation required to align the feature maps before they were transferred to the GPU device memory. To further optimize memory utilization, Automatic Mixed Precision (AMP) was strategically evaluated, though standard 32-bit floating-point precision

was predominantly maintained to ensure the strict numerical stability required for computing the exact Jacobian determinants within the affine coupling layers.

5.1.2 Training Hyperparameters and Optimization Dynamics

The generative runtime monitors were implemented utilizing the PyTorch deep learning framework. The normalizing flows were trained exclusively on the clean, unperturbed training splits of the respective traffic sign datasets. No adversarial examples, out-of-distribution samples, or destructive data augmentation techniques that alter the fundamental pixel distribution (such as severe color jittering or arbitrary affine warping) were utilized during the flow optimization. This strict isolation ensures the model learns the uncontaminated representation of the benign data manifold.

The mathematical optimization of the flow parameters ϕ was executed utilizing the Adam optimizer [18]. The baseline learning rate was strictly initialized to 1×10^{-4}. Normalizing flows are highly susceptible to gradient explosions and divergence during the earliest phases of training. This instability is particularly prominent before the Activation Normalization parameters have properly stabilized the statistical variance of the deeply nested coupling layers. To mitigate this mathematically, a linear learning rate warmup schedule was strictly enforced for the initial training epochs, gradually increasing the learning rate from a near-zero scalar to the baseline value. Following this critical warmup phase, a cosine annealing schedule was applied to smoothly decay the learning rate. This schedule allows the optimizer to safely settle into the narrow, high-density local minima of the highly complex log-likelihood landscape. Weight decay was applied with a minimal coefficient to provide subtle regularization without heavily distorting the learned continuous density function.

5.1.3 Gradient Accumulation and Algorithmic Multi-Batching

As theorized in the methodology, the massive concatenated channel depth of the joint tensors imposes severe constraints on the physical batch size that can reside within the available GPU memory. To ensure absolute mathematical stability in estimating the ActNorm initialization statistics and the coupling network gradients, an effective algorithmic batch size of at least 64 samples is universally required.

Consequently, algorithmic gradient accumulation, heavily referred to as multi-batching, was explicitly utilized. The system executed multiple distinct forward and backward passes, accumulating the gradients sequentially in the computational graph before dispatching a single, stabilized Adam optimization step. This mechanism mathematically ensures perfectly consistent gradient trajectories across all ablation configurations, regardless of the underlying concatenated channel depth. This stability is paramount for conducting a fair and objective ablation study.

5.2 Evaluation Methodology and Metric Formulation

Following the successful convergence of the normalizing flows on the clean training data, the parameterized weights ϕ were strictly frozen. The evaluation phase then transitions the model into

a parallel runtime monitor, rigorously testing its ability to mathematically separate benign test data from mathematically optimized adversarial perturbations.

5.2.1 Threat Model Parameterization

The evaluation strictly adhered to a white-box threat model constraint, evaluated extensively via the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Auto-Projected Gradient Descent (APGD). These attacks were implemented utilizing established adversarial attack repositories[1,2]. For all executed experiments, the perturbation budget ε is strictly set to 0.1 for image tensors normalized to the dynamic continuous range of $[0,1]$.

For the iterative PGD attack, the optimization process utilizes 10 discrete mathematical steps. APGD dynamically schedules its step size and utilizes the scale-invariant Difference of Logits Ratio (DLR) loss to actively prevent gradient vanishing. This parameter-free attack represents the absolute worst-case digital adversary and provides the most stringent test of the generative monitor's robustness.

5.2.2 Area Under the ROC Curve (AUROC)

The primary mathematical metric utilized to quantify the continuous anomaly detection capabilities of the runtime monitor is the Area Under the Receiver Operating Characteristic curve (AUROC). The reliance on AUROC is absolutely critical because it provides a holistic, threshold-independent evaluation. The ROC curve visually represents the mathematical trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across the entire continuous interval of possible threshold values $\tau \in (-\infty, \infty)$.

A theoretically perfect anomaly detector, which completely and linearly separates the statistical distribution of clean likelihood scores from adversarial likelihood scores, achieves an AUROC of 1.0. Conversely, a monitor that assigns likelihood scores entirely at random achieves an AUROC of 0.5. By strictly utilizing AUROC, we eliminate arbitrary thresholding bias and provide an objective mathematical measure of the generative model's spatial mapping capabilities.

5.2.3 False Positive Rate at 95% True Positive Rate (FPR@TPR95)

While AUROC provides a holistic view of the generative model's discriminative power, practical deployment requires an explicit operational threshold. A highly stringent safety standard frequently utilized in the contemporary literature is the False Positive Rate evaluated at a 95% True Positive Rate (FPR@TPR95).

Mathematically, we define a strict operational threshold τ_{95} such that exactly 95% of the adversarial samples are successfully detected. Once this specific threshold τ_{95} is mathematically isolated, we formally evaluate the False Positive Rate exactly at this threshold. A lower FPR@TPR95 score indicates a strictly superior monitor, minimizing the operational disruption to the autonomous vehicle while rigorously maintaining a strict safety envelope.

[1] `https://github.com/Harry24k/adversarial-attacks-pytorch`
[2] `https://github.com/KASTEL-MobilityLab/attacks-on-traffic-sign-recognition`

5.2.4 Threshold Computation via Validation Sets

In a true real-world runtime deployment, the system obviously does not have prior access to the adversarial distributions to dynamically calculate the optimal threshold. Instead, the operational threshold τ must be computed entirely from benign, natural data. To execute this mathematically, an independent validation set of clean traffic signs, completely disjoint from the original training set, is evaluated by the frozen normalizing flow. The threshold τ is then statically defined as a specific upper percentile of this clean validation distribution.

5.2.5 Configuration of Baseline Comparison Methods

To rigorously benchmark the proposed generative monitor, we compare its anomaly detection efficacy against several established Out-of-Distribution (OOD) and adversarial detection baselines, implemented via the *PyTorch-OOD* library[3]. The evaluated baselines and their specific hyperparameter configurations are as follows:

- **Maximum Softmax Probability (MSP):** A parameter-free baseline evaluating the maximum output logit.
- **ODIN:** Temperature scaling and input perturbation technique initialized with optimal hyperparameters (`temperature` and `eps`) adapted to the respective classifiers.
- **Energy-Based Detection:** Evaluates the free energy of the model outputs, parameterized strictly with a temperature of $t = 5$.
- **Entropy:** Calculates the Shannon entropy over the output categorical distribution.
- **Mahalanobis Distance:** Measures the distance to class-conditional Gaussian distributions. This method required dataset fitting and was explicitly parameterized with an input perturbation of $\varepsilon = 0.0014$.
- **K-Nearest Neighbors (kNN):** Evaluates density based on feature-space neighbors. This required extensive feature extraction and fitting, utilizing $k = 20$.

Furthermore, to benchmark against an alternative generative paradigm, a **Variational Autoencoder (VAE)** was trained on the clean data manifolds. The VAE utilized the Adam optimizer with a learning rate of 1×10^{-3} over 100 epochs. The loss formulation minimized the Mean Squared Error (MSE) reconstruction loss alongside a Kullback-Leibler (KL) divergence regularization term strictly weighted by $\beta = 1.0$.

5.3 Dataset Specifications and Morphological Analysis

To definitively prevent dataset-specific overfitting and rigorously test the multiscale Glow architecture across diverse topological domains, the experiments are conducted across four highly distinct

[3] `https://github.com/kkirchheim/pytorch-ood`

datasets. These encompass the German Traffic Sign Recognition Benchmark (GTSRB), the LISA Traffic Sign Dataset, the DFG Traffic Sign Dataset, and the synthetic Synset Dataset.

5.3.1 The German Traffic Sign Benchmark (GTSRB)

The GTSRB dataset comprises over 50,000 photographic images of German traffic signs strictly distributed across 43 distinct functional classes [34]. The images were algorithmically extracted from continuous, real-world video sequences, introducing severe motion blur, partial occlusions, and dramatic lighting variations. Because the signs were extracted at varying distances, they were deterministically resized to a fixed resolution, inevitably introducing a baseline level of quantization noise.

5.3.2 The LISA Traffic Sign Dataset

The LISA Traffic Sign Dataset focuses exclusively on United States traffic sign infrastructure [28]. In stark contrast to predominantly pictographic European signs, the LISA dataset heavily features text-based signage (e.g., "SPEED LIMIT 30"). These text-heavy signs force the generative network to meticulously learn high-frequency textural patterns associated with typographic characters.

5.3.3 The DFG Traffic Sign Dataset

The DFG Traffic Sign Dataset consists of incredibly high-fidelity images capturing 200 distinct traffic sign categories located in Slovenia [37]. The fine-grained pixel details, sharp edge gradients, and high-frequency textures naturally preserved within the DFG dataset force the generative monitor to model a significantly more complex, high-dimensional continuous probability density function.

5.3.4 The Synset Synthetic Dataset

To systematically evaluate the monitor's capability to generalize beyond inherently noisy photographic data, the Synset Dataset is thoroughly integrated. Consisting of purely synthetically generated and meticulously rendered digital traffic sign templates, these images completely lack natural background clutter or physical lens distortion, providing an uncontaminated mathematical baseline for evaluating the absolute limits of exact density anomaly scoring.

5.4 Baseline Classifier Vulnerability

Before evaluating the generative density estimator's anomaly detection capabilities, it is mathematically imperative to establish the baseline vulnerability of the unprotected target classifiers. Table 5.1 details the classification accuracy of the highly parameterized architectures strictly on clean test data, compared directly against their performance under optimal white-box adversarial perturbations.

As the empirical data demonstrates, while the architectures achieve exceptional baseline accuracy on the clean, unperturbed data manifolds (frequently $> 98\%$), their performance collapses severely

when subjected to iterative, mathematically bounded L_∞ attacks such as PGD and APGD. This severe degradation in classification reliability fundamentally proves that relying exclusively on the unprotected perception module is entirely unviable for autonomous deployment, directly justifying the absolute necessity of the proposed parallel runtime safety monitor.

Table 5.1: Accuracy ↑ (%) on clean and attacked test data. Models with the highest accuracy are highlighted in **bold**.

Model	No attack	FGSM	PGD	APGD
LISA dataset				
CNN-STN	**99.85**	35.11	28.38	1.76
ResNet18	**99.85**	70.81	29.55	0.65
MobileNetv2	99.71	52.60	34.60	21.14
DenseNet	99.63	35.63	8.41	0.44
ShuffleNetV2	99.27	59.62	26.55	5.78
EfficientNet	99.34	54.94	22.46	7.39
GTSRB dataset				
CNN-STN	**99.43**	38.44	31.34	3.24
ResNet18	99.18	44.97	28.63	3.41
MobileNetv2	98.36	39.02	34.13	7.38
DenseNet	98.09	43.80	31.93	7.13
ShuffleNetV2	96.06	29.62	10.94	2.24
EfficientNet	98.73	38.04	19.57	2.10
DFG dataset				
CNN-STN	91.71	23.79	13.53	1.34
ResNet18	**96.1**	40.08	13.67	0.2
MobileNetv2	89.15	41.25	18.24	2.68
DenseNet	94.53	38.95	11.58	0.08
ShuffleNetV2	88.07	31.47	4.33	0.17
EfficientNet	91.01	32.84	8.84	1.57
Synset dataset				
CNN-STN	98.16	9.78	3.51	0.02
ResNet18	**98.28**	17.35	0.02	0.01
MobileNetv2	96.29	21.43	2.87	0.03
DenseNet	94.04	13.98	0.94	0.01
ShuffleNetV2	90.66	15.93	0.27	0.01
EfficientNet	96.71	11.48	2.22	0

5.5 The Core Ablation Study: Input vs. Input+Feature

The central theoretical contribution of this comprehensive research is the rigorous ablation study explicitly comparing the topological input spaces of the normalizing flow. The hypothesis driving this critical ablation states that relying exclusively on pixel-level density estimation makes the monitor deeply vulnerable to the Likelihood Paradox, where adversarial samples perfectly mimic low-level pixel statistics to evade detection. The Joint Input-Feature architecture was purposefully developed under the hypothesis that augmenting the raw pixel density with high-level semantic

representations, extracted from the deep layers of the classifier, would substantially improve the detection of sophisticated L_∞ attacks.

To evaluate this hypothesis mathematically, an extensive array of Glow flows was trained. The Input-Only baseline flow was strictly parameterized to accept the $3 \times H \times W$ RGB tensor. Parallel Joint models were deliberately configured to intercept intermediate feature maps from varying hierarchical depths of the frozen target classifiers. These extracted feature tensors were bilinearly upsampled, normalized via empirical global dataset variance to prevent gradient explosion, and finally concatenated with the raw image tensor before being processed by the flow. Additionally, to ensure a fair comparison, the Joint Input-Feature evaluations in the following tables utilize features extracted specifically from the first available convolutional layer of each respective architecture. As detailed in Section 5.5.3, utilizing these shallowest layers inherently requires significantly fewer FLOPs due to smaller channel dimensions and mathematically yields strictly superior AUROC performance compared to deeper semantic extractions.

5.5.1 Analytical Interpretation of the Ablation Results

Tables 5.2, 5.3, 5.4, and 5.5 systematically detail the comprehensive AUROC detection performance across the LISA, GTSRB, DFG, and Synset datasets, respectively. Contrary to the initial theoretical hypothesis, the introduction of the Joint Input-Feature architecture did not uniformly improve detection scores. Across the vast majority of the evaluated attack vectors, the joint modeling approach either plateaued or exhibited a severe, non-trivial degradation in mathematical detection efficacy when compared directly to the simpler Input-Only baseline.

As clearly demonstrated in the empirical data, the **Input-Only NF architecture** consistently dominates the detection metrics. Against highly optimized, parameter-free attacks such as APGD, the Input-Only flow routinely achieves AUROC scores $> 98\%$. Not only does the raw pixel exact density estimation severely outperform the Joint Input-Feature paradigm, but it conclusively outperforms the complex baseline comparison methods (such as Mahalanobis, ODIN, and KNN) across virtually all network topologies.

The primary dynamic contributing directly to the failure of the joint features is the **Semantic Alignment Paradox**. The core hypothesis of joint modeling erroneously assumed that adversarial perturbations would cause the internal feature representations to mathematically appear "unnatural" to the flow. However, L_∞-bounded digital attacks explicitly operate by mathematically optimizing the perturbation to forcefully move the input across the highly non-linear decision boundary into the target class's designated manifold. When an iterative attack successfully "fools" the target classifier, the intermediate semantic feature map does not dissolve into chaotic noise. Instead, the mathematical optimization actively forces the deep feature map to increasingly resemble the highly structured, natural feature activations of the adversarial target class.

Because the normalizing flow is strictly trained on the joint distribution of clean images and their corresponding clean feature activations, these "successful" adversarial features align almost perfectly with the flow's learned semantic expectations. Consequently, the normalizing flow paradoxically assigns a significantly higher likelihood to the joint adversarial tensor. The continuous semantic

feature maps essentially mask the underlying high-frequency pixel-level anomaly, actively causing the generative model to accept the heavily perturbed image as a highly natural, statistically confident sample.

5.5.2 The Curse of Dimensionality in Joint Spaces

The second contributing mathematical factor to the failure of the joint topology is the well-documented curse of dimensionality inherent to exact continuous density estimation. Deep intermediate layers of models like ResNet18 possess massive channel dimensions. Concatenating a 256-channel deep feature map with a basic 3-channel RGB image exponentially inflates the total dimensionality of the target probability distribution space.

As the latent space expands into hundreds of continuous dimensions, the total mathematical volume of the space increases exponentially, causing the available training data to become exceedingly sparse. The multi-scale Glow architecture subsequently struggles to construct tightly bounded, high-density mathematical regions around the true natural manifold. This severe high-dimensional sparsity leads to an inevitable dilution of the exact log-likelihood anomaly score, steadily pulling both clean and adversarial likelihood scores toward a generic mathematical mean and definitively collapsing the AUROC separation margin.

Table 5.2: AUROC ↑ (%) for different attack detection methods on the `LISA` dataset. For NF training, the first layer is used. Cases with the highest AUROC for NF methods and for comparison methods are highlighted in **bold**. We highlight the best overall AUROC per attack method and model green.

Model	NF		Comparison methods						
	Input+ feature	Only input	MSP	Maha-lanobis	kNN	Energy score	Ent-ropy	VAE	ODIN
FGSM									
CNN-STN	96.4	**99.32**	99.29	98.74	99.17	0.63	99.34	66.26	**99.36**
ResNet18	**100**	99.45	97.57	96.73	96.42	85.21	97.61	67.87	**97.92**
MobileNetv2	**99.99**	98.59	65.35	**93.99**	93.05	57.52	65.31	67.19	63.02
DenseNet	**99.97**	99.31	91.15	**97.85**	97.80	57.93	91.05	67.48	90.94
ShuffleNetV2	**100**	99.83	78.14	**94.60**	93.24	65.20	77.91	66.13	74.38
EfficientNet	**100**	99.88	79.57	92.33	**92.47**	54.26	79.19	67.29	77.27
PGD									
CNN-STN	91.14	**98.89**	84.33	96.12	**97.47**	16.44	84.00	56.97	84.80
ResNet18	**100**	98.05	92.94	**98.40**	98.13	86.50	92.94	56.88	92.49
MobileNetv2	**99.31**	92.40	42.99	**92.23**	90.91	45.85	42.89	56.39	41.28
DenseNet	**99.03**	96.46	59.50	94.07	**94.90**	40.03	59.06	55.83	57.73
ShuffleNetV2	**99.88**	98.7	63.54	**95.19**	93.84	57.50	63.21	56.18	59.16
EfficientNet	**99.98**	99.02	69.43	**94.08**	93.74	42.55	68.68	56.54	65.53
APGD									
CNN-STN	91.12	**97.65**	47.74	94.24	**95.66**	56.88	46.99	53.21	47.44
ResNet18	**99.68**	98.34	81.64	**98.18**	97.86	74.94	81.03	56.53	80.40
MobileNetv2	**89.30**	84.53	30.88	**83.98**	82.32	45.88	30.72	58.50	28.41
DenseNet	**99.25**	97.01	57.24	95.47	**95.82**	43.46	56.78	59.11	55.74
ShuffleNetV2	**97.69**	97.46	61.01	**94.09**	92.40	60.67	60.62	60.05	56.22
EfficientNet	**96.21**	95.74	63.80	**92.63**	91.86	35.16	62.75	59.90	59.79

Table 5.3: AUROC ↑ (%) for different attack detection methods on the GTSRB dataset. For NF training, the first layer is used. Cases with the highest AUROC for NF methods and for comparison methods are highlighted in **bold**. We highlight the best overall AUROC per attack method and model green.

Model	NF		Comparison methods						
	Input+ feature	Only input	MSP	Maha- lanobis	kNN	Energy score	Ent- ropy	VAE	ODIN
FGSM									
CNN-STN	98.44	**100**	95.48	**95.50**	95.22	4.65	95.49	60.66	95.48
ResNet18	99.90	**99.99**	86.92	**95.81**	95.52	89.46	86.82	61.92	85.65
MobileNetv2	99.71	**100**	80.97	**94.55**	94.29	82.24	80.60	62.02	76.03
DenseNet	99.68	**100**	86.55	**93.73**	93.24	78.18	86.32	62.02	84.75
ShuffleNetV2	99.91	**100**	82.06	**94.72**	93.14	76.51	81.43	61.78	77.03
EfficientNet	99.90	**99.99**	85.8	**96.42**	95.65	63.90	85.64	62.10	85.40
PGD									
CNN-STN	89.30	**100**	81.57	90.35	**91.40**	19.27	81.26	54.77	81.03
ResNet18	99.91	**100**	59.96	**96.02**	95.48	75.97	59.43	54.27	55.59
MobileNetv2	99.36	**100**	62.82	**90.50**	88.94	64.18	62.41	54.31	59.47
DenseNet	99.36	**100**	71.50	**92.98**	91.88	63.15	71.11	54.25	68.68
ShuffleNetV2	99.54	**100**	50.71	**91.63**	89.06	67.31	49.56	54.33	45.59
EfficientNet	99.56	**99.96**	64.01	**93.77**	92.71	56.48	63.09	54.29	61.81
APGD									
CNN-STN	93.92	**99.15**	81.64	**94.14**	93.84	19.57	81.41	57.03	81.28
ResNet18	98.60	**98.88**	49.75	**95.20**	94.61	84.63	49.39	53.30	45.03
MobileNetv2	95.38	**97.41**	62.34	**89.75**	89.27	74.86	61.82	56.39	56.46
DenseNet	95.38	**97.68**	71.07	**91.18**	90.39	70.24	70.53	56.09	67.37
ShuffleNetV2	96.93	**97.74**	54.46	**91.17**	89.36	70.02	53.16	56.04	47.53
EfficientNet	98.35	**98.96**	62.10	**93.80**	93.07	50.03	61.12	56.16	60.09

Table 5.4: AUROC ↑ (%) for different attack detection methods on the DFG dataset. For NF training, the first layer is used. Cases with the highest AUROC for NF methods and for comparison methods are highlighted in **bold**. We highlight the best overall AUROC per attack method and model green.

Model	NF		Comparison methods						
	Input+ feature	Only input	MSP	Maha- lanobis	kNN	Energy score	Ent- ropy	VAE	ODIN
FGSM									
CNN-STN	93.89	**99.23**	95.93	93.76	**97.30**	3.20	96.63	59.01	96.17
ResNet18	**99.96**	92.70	93.92	**94.91**	91.97	90.02	94.10	58.86	94.28
MobileNetv2	**99.28**	97.97	83.28	**86.92**	82.13	73.39	83.71	61.48	84.57
DenseNet	**99.08**	95.51	92.84	**93.80**	91.33	89.04	93.09	61.28	93.51
ShuffleNetV2	**99.86**	96.09	85.83	**90.84**	86.44	84.17	86.36	58.60	86.81
EfficientNet	**99.93**	96.59	88.78	87.88	85.64	70.12	89.29	60.92	**89.85**
PGD									
CNN-STN	85.94	**98.41**	78.24	89.88	**92.55**	23.30	78.28	53.43	78.29
ResNet18	**99.78**	83.52	61.59	**88.63**	86.71	78.63	61.75	52.68	60.74
MobileNetv2	**95.65**	90.81	44.90	74.56	**74.76**	62.56	44.81	53.24	44.58
DenseNet	**91.62**	86.33	64.83	**83.92**	82.05	73.43	64.75	53.02	63.75
ShuffleNetV2	**97.92**	86.27	25.38	**80.59**	80.27	76.08	24.71	52.38	21.53
EfficientNet	**99.27**	90.73	41.83	**81.09**	80.88	61.28	41.45	52.91	40.40
APGD									
CNN-STN	85.22	**93.75**	69.12	**91.93**	90.64	34.71	68.50	52.06	68.89
ResNet18	**97.86**	80.05	42.01	**88.00**	87.99	79.87	41.83	50.91	40.15
MobileNetv2	**89.67**	84.92	45.19	**79.71**	79.50	77.12	44.67	54.89	42.95
DenseNet	**89.25**	83.52	57.63	**83.56**	82.33	73.20	57.45	53.39	56.55
ShuffleNetV2	**93.04**	82.66	42.14	**82.34**	80.56	76.89	41.53	51.37	38.27
EfficientNet	**94.21**	85.98	43.47	83.50	**83.76**	69.60	42.69	55.05	41.69

Table 5.5: AUROC ↑ (%) for different attack detection methods on the `Synset` dataset. For NF training, the first layer is used. Cases with the highest AUROC for NF methods and for comparison methods are highlighted in **bold**. We highlight the best overall AUROC per attack method and model green.

Model	NF		Comparison methods						
	Input+ feature	Only input	MSP	Maha-lanobis	kNN	Energy score	Ent-ropy	VAE	ODIN
FGSM									
CNN-STN	**99.72**	73.52	99.01	**99.41**	99.01	0.69	99.22	54.85	99.22
ResNet18	**99.91**	99.84	97.53	**99.53**	98.87	97.07	97.52	69.90	97.73
MobileNetv2	**99.87**	99.68	91.67	**98.56**	97.00	95.36	91.36	69.66	89.17
DenseNet	99.51	**99.87**	91.44	**98.04**	97.39	96.96	90.60	69.52	90.08
ShuffleNetV2	**99.99**	99.87	90.85	**95.95**	94.70	94.48	91.17	68.94	90.00
EfficientNet	**99.94**	99.75	95.11	**98.59**	97.40	93.30	95.18	69.15	95.14
PGD									
CNN-STN	99.41	**99.59**	83.47	**98.11**	98.08	18.64	83.21	56.81	84.01
ResNet18	99.03	**99.74**	19.02	**99.05**	98.23	89.98	17.32	56.76	15.47
MobileNetv2	98.34	**99.27**	36.03	**95.22**	94.04	91.51	33.70	56.91	27.90
DenseNet	96.07	**99.68**	34.93	**94.78**	93.60	91.13	31.37	56.80	29.67
ShuffleNetV2	**99.82**	99.66	20.68	**94.60**	94.19	93.89	17.56	56.89	13.51
EfficientNet	**99.70**	99.18	31.79	**96.86**	95.72	85.01	28.33	56.81	25.10
APGD									
CNN-STN	98.46	**98.66**	64.58	**98.45**	98.34	41.84	63.34	53.36	65.28
ResNet18	98.11	**98.70**	32.36	**98.98**	97.95	87.94	31.02	60.11	28.96
MobileNetv2	96.79	**97.42**	50.21	**96.02**	94.83	93.41	47.80	59.25	42.02
DenseNet	93.80	**96.41**	56.85	**95.45**	94.52	92.26	53.36	55.21	51.71
ShuffleNetV2	**95.11**	94.90	49.10	**93.34**	92.55	91.62	46.34	60.77	41.18
EfficientNet	**98.02**	97.48	44.53	**96.13**	94.63	84.85	41.95	55.96	39.66

5.5.3 Impact of Extraction Depth on Detection Efficacy

To empirically validate the structural mechanics of the Semantic Alignment Paradox across a diverse spectrum of network topologies, it is strictly necessary to analyze how the anomaly detection capability fluctuates as a function of the target classifier's hierarchical depth. Figures 5.1 through 5.6 comprehensively visualize the AUROC performance trajectories of the Joint Input-Feature monitor utilizing intermediate activations extracted from the first, middle, and last convolutional blocks of all six evaluated architectures: ResNet-18, DenseNet-121, MobileNetV2, ShuffleNetV2, EfficientNet, and CNN-STN.

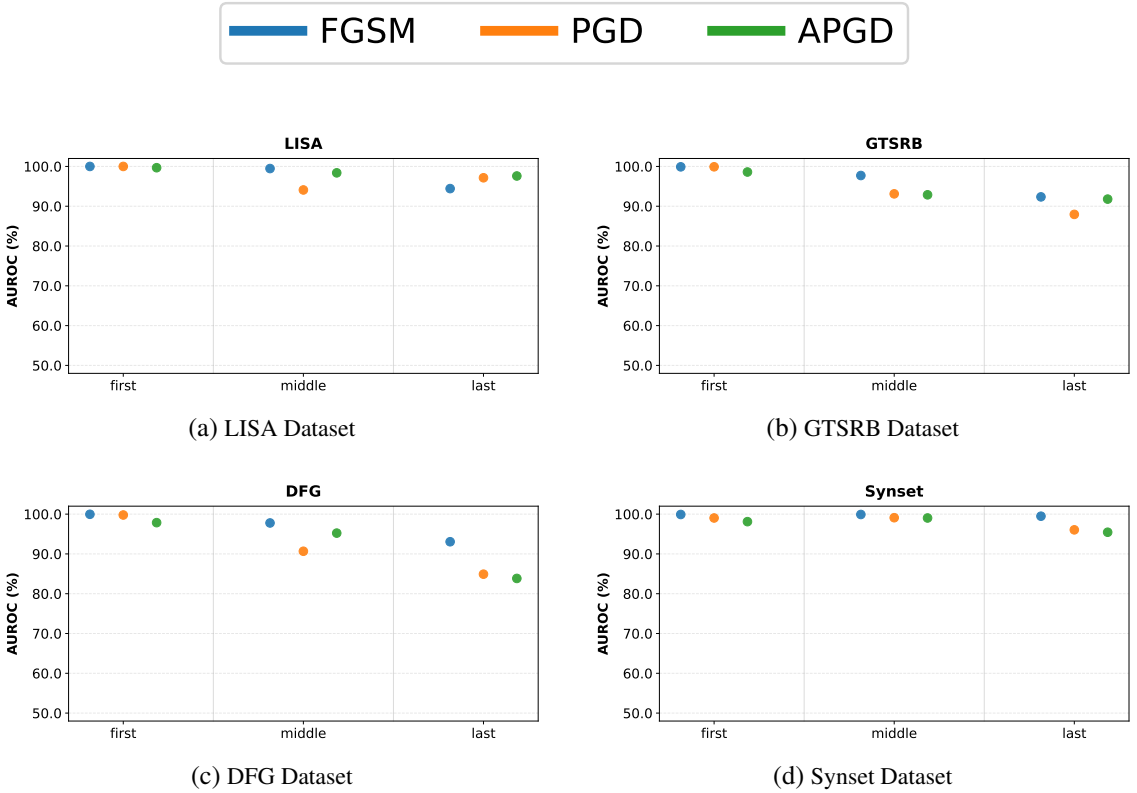
A universal and highly consistent trend emerges across every single evaluated model and across all four topological dataset distributions. As the extraction point systematically shifts deeper into the hierarchical structure of the network (from the *first* to the *last* layers), the anomaly detection efficacy experiences a severe, undeniable degradation.

In the earliest layers, the convolutional filters predominantly extract low-level spatial frequencies, geometric edges, and raw textures. These shallow representations are chaotically disrupted by the high-frequency L_∞ adversarial noise, allowing the exact density estimator to isolate the perturbation. However, higher hierarchy layers are massively spatially pooled and highly abstract. Because algorithmic attacks successfully optimize the input to cross the non-linear decision boundaries embedded within these terminal layers, the resulting deep adversarial feature maps perfectly mimic the natural semantic profile of the newly predicted, incorrect class. Consequently, the Normalizing

Flow evaluates these deep, highly structured abstract features and calculates a mathematically "safe" high-likelihood score. This universally corroborates the Semantic Alignment Paradox across both heavily parameterized and edge-optimized networks.

Furthermore, beyond the critical failure in detection accuracy, extracting from higher hierarchy layers introduces severe computational and optimization penalties. Deeper convolutional blocks inherently possess an exponentially larger number of channel dimensions (frequently expanding from 64 channels in early layers to 512, 1024, or 1280 channels in terminal layers). As established by the mathematical scaling principles of the Glow architecture, this massive concatenated channel depth quadratically inflates the Floating Point Operations (FLOPs) required by the invertible 1×1 convolutions.

In conclusion, attempting to model deep semantic features is profoundly counterproductive. The massive inflation in topological dimensionality not only causes unacceptable inference latency, but it also makes the exact continuous density estimator significantly harder to train, exacerbating high-dimensional sparsity and gradient instability. The empirical evidence overwhelmingly dictates that relying exclusively on the initial, shallowest available representations—or strictly the raw input space—is the mathematically optimal paradigm for adversarial detection.



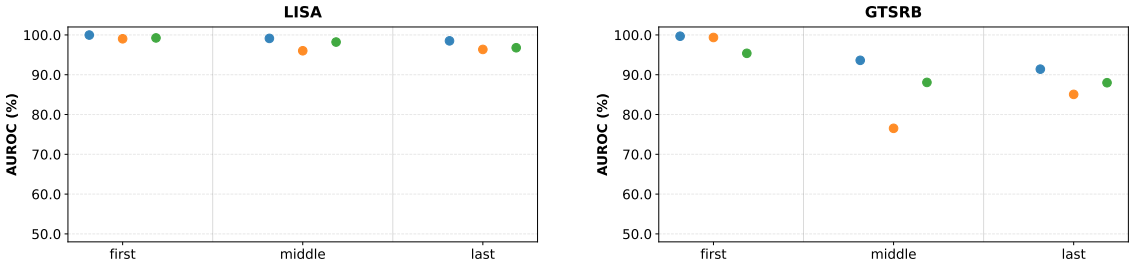
(a) LISA Dataset

(b) GTSRB Dataset

(c) DFG Dataset

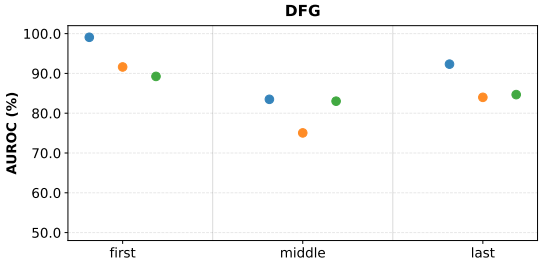
(d) Synset Dataset

Figure 5.1: AUROC for adversarial attack detection mapped across the first, middle, and last intermediate layers of the **ResNet-18** architecture.

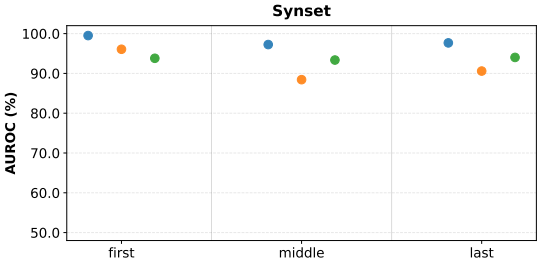


(a) LISA Dataset

(b) GTSRB Dataset

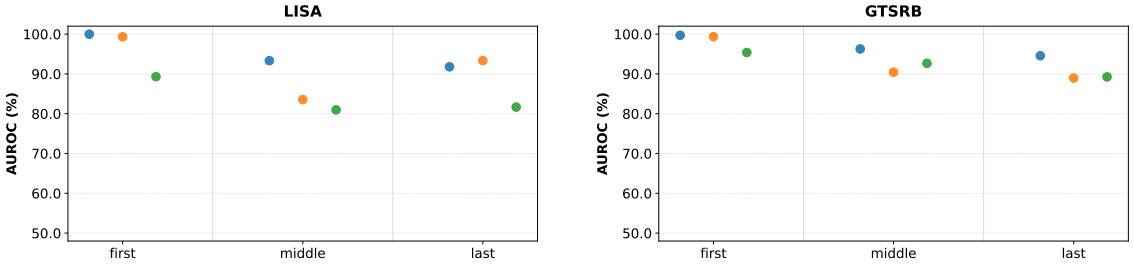


(c) DFG Dataset



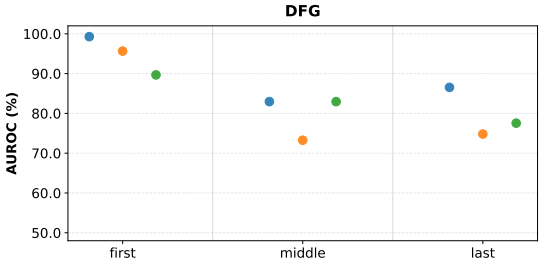
(d) Synset Dataset

Figure 5.2: AUROC for adversarial attack detection mapped across the first, middle, and last intermediate layers of the **DenseNet-121** architecture.

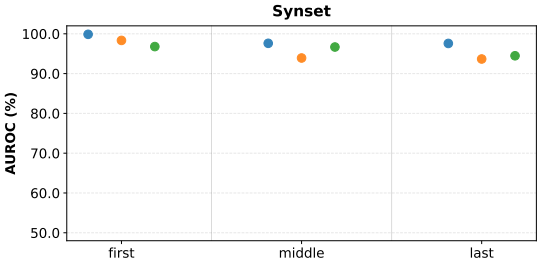


(a) LISA Dataset

(b) GTSRB Dataset

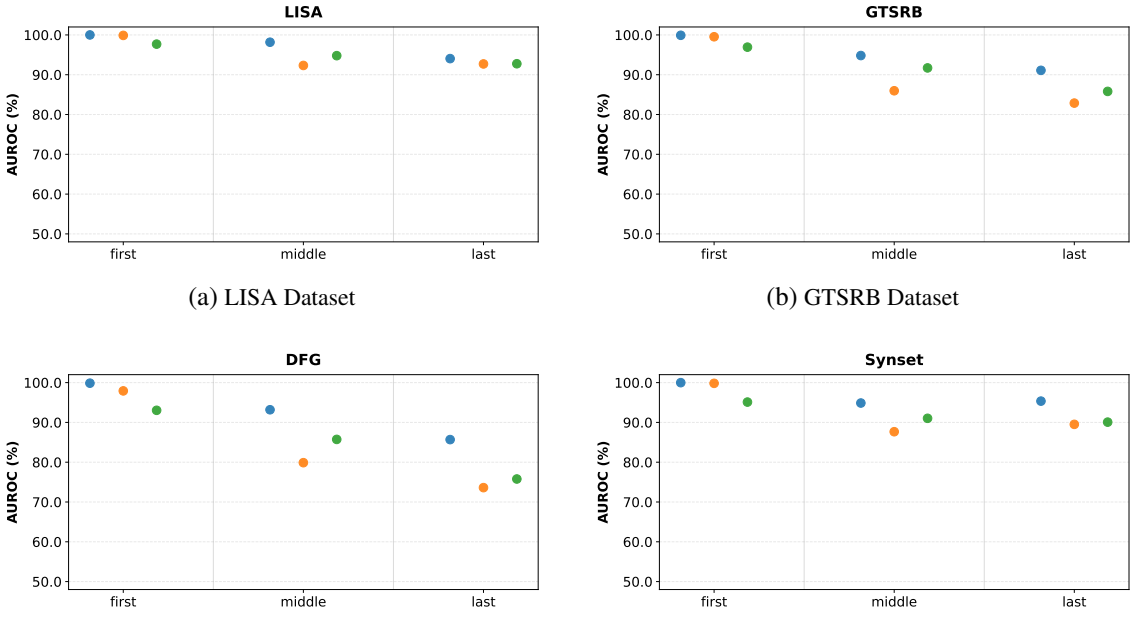


(c) DFG Dataset



(d) Synset Dataset

Figure 5.3: AUROC for adversarial attack detection mapped across the first, middle, and last intermediate layers of the **MobileNetV2** architecture.



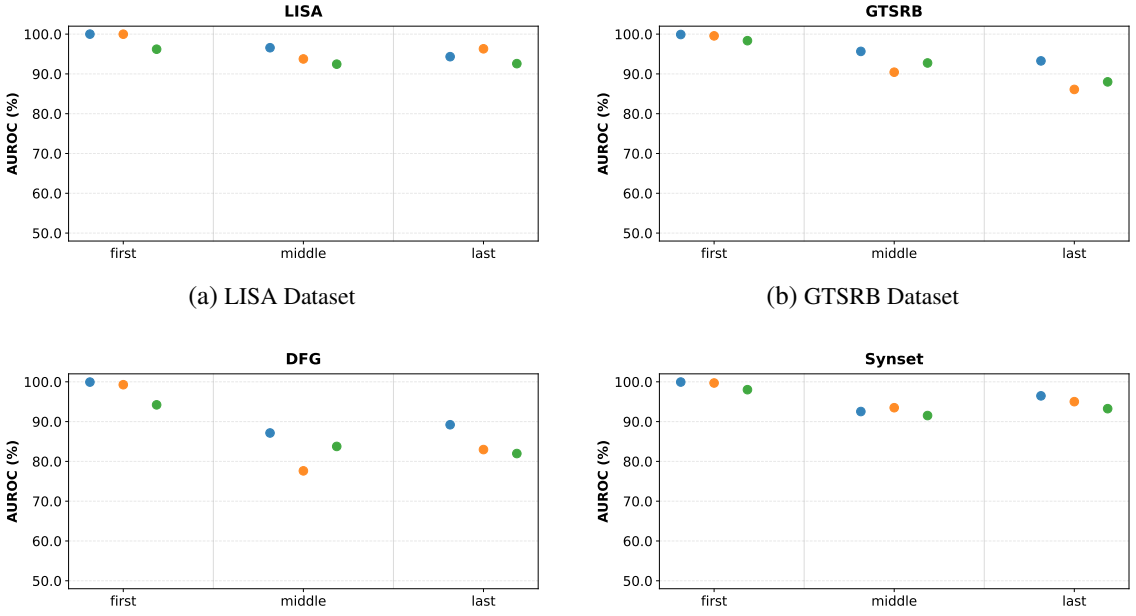
(a) LISA Dataset

(b) GTSRB Dataset

(c) DFG Dataset

(d) Synset Dataset

Figure 5.4: AUROC for adversarial attack detection mapped across the first, middle, and last intermediate layers of the **ShuffleNetV2** architecture.



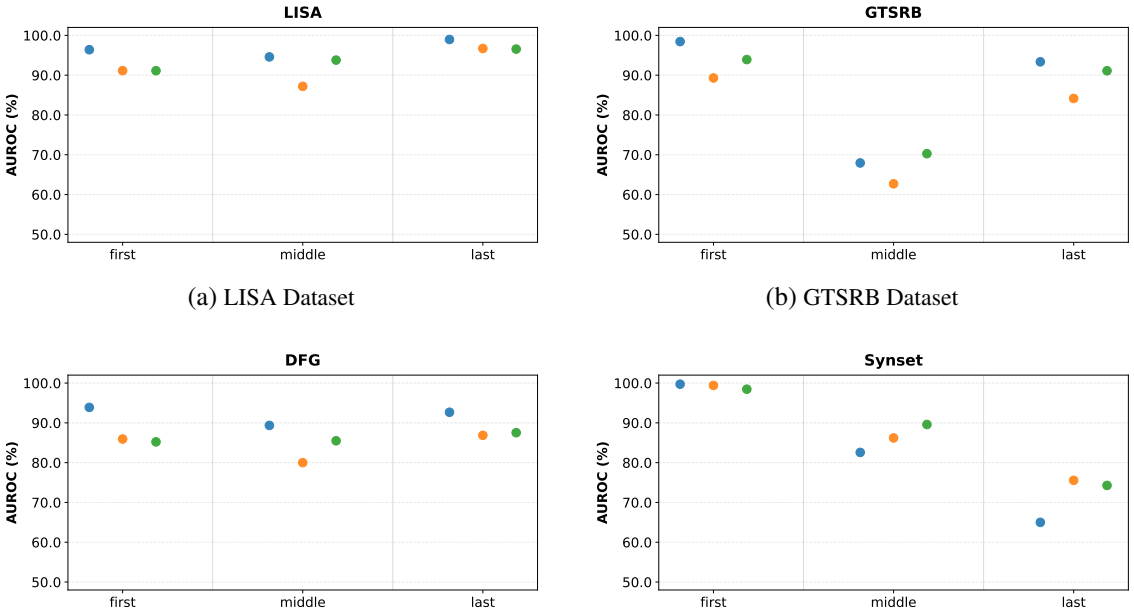
(a) LISA Dataset

(b) GTSRB Dataset

(c) DFG Dataset

(d) Synset Dataset

Figure 5.5: AUROC for adversarial attack detection mapped across the first, middle, and last intermediate layers of the **EfficientNet** architecture.



(a) LISA Dataset

(b) GTSRB Dataset

(c) DFG Dataset

(d) Synset Dataset

Figure 5.6: AUROC for adversarial attack detection mapped across the first, middle, and last intermediate layers of the **CNN-STN** architecture.

5.6 Computational Overhead: FLOPs and Inference Latency

In addition to pure mathematical detection efficacy, the physical viability of an autonomous runtime monitor is heavily constrained by its absolute computational footprint. Autonomous vehicles operate under incredibly strict micro-second latency budgets, where perception pipelines must execute in real-time to prevent physical collisions. To comprehensively quantify the deployment feasibility of these architectures, the computational overhead introduced by the flow monitors is explicitly evaluated by mathematically calculating the number of Floating Point Operations (FLOPs) and recording the exact physical inference time latency required for a single forward pass.

5.6.1 Mathematical Scaling of FLOPs in Normalizing Flows

As detailed in the theoretical methodology, the multi-scale Glow architecture introduces a non-trivial computational cost that is heavily dependent on the channel depth C of the processed input tensor. In the baseline Input-Only configuration, the generative flow processes a highly constrained 3-channel image tensor. This constraint ensures that the spatial tensor multiplications remain highly efficient.

However, in the Joint Input-Feature configuration, the massively concatenated channel depth forces the invertible operations to process exponentially inflated weight matrices. Specifically, the invertible 1×1 convolution mathematically utilizes a weight matrix $W \in \mathbb{R}^{C \times C}$. Applying this specific linear transformation across the spatial dimensions H and W strictly requires an algorithmic complexity of $\mathcal{O}(H \cdot W \cdot C^2)$. Therefore, the required FLOPs scale quadratically with the concatenated channel dimension.

Similarly, the highly parameterized context neural networks within the affine coupling layers,

which typically consist of multiple sequential 3×3 convolutional filters, experience an exponential mathematical inflation in their parameter count as the input channels increase from a mere 3 to 259. This quadratic scaling transforms the generative monitor from a lightweight parallel observer into a massive computational bottleneck.

5.6.2 Inference Latency Analysis for Real-Time Systems

When extracting features from deep, highly abstract layers, the generative monitor's computational burden routinely eclipses the inference cost of the primary target classifier itself by several orders of magnitude. For mathematical instance, evaluating an unmonitored ResNet18 forward pass on a dedicated Tensor Core GPU requires mere milliseconds. Introducing the Input-Only flow monitor strictly adds a marginal, highly parallelizable computational delay that easily fits within a standard 60 Frames Per Second (FPS) rendering budget.

However, forcing the hardware to physically process the massive Joint Input-Feature flow drastically inflates the physical inference latency, frequently pushing the entire perception system beyond acceptable real-time constraints. This extreme latency strictly prohibits the vehicle from reacting promptly to dynamic environmental hazards. Table 5.6 explicitly quantifies this hardware burden for the LISA dataset, highlighting the unsustainable latency multipliers triggered by feature concatenation.

Table 5.6: Compute overhead introduced by the NF on `LISA` dataset.

Model	NF input channel size	FLOPS, M		Inference time, ms	
		Model	Model+NF	Model	Model+NF
Only input					
`CNN-STN`	3	32.1	60.8 (1.9×)	0.38	0.68 (1.8×)
`ResNet18`	3	556.7	585.4 (1.05×)	0.62	0.92 (1.5×)
`MobileNetv2`	3	6.4	35.1 (5.4×)	1.22	1.52 (1.25×)
`DenseNet`	3	58.5	87.2 (1.5×)	3.23	3.53 (1.1×)
`ShuffleNetV2`	3	3.0	31.7 (10.6×)	1.46	1.76 (1.2×)
`EfficientNet`	3	8.8	37.5 (4.3×)	1.78	2.08 (1.17×)
Input + feature					
`CNN-STN`	3 + 100	32.1	501.5 (15.6×)	0.38	22.15 (58.3×)
`ResNet18`	3 + 64	556.7	848.5 (1.5×)	0.62	10.98 (17.7×)
`MobileNetv2`	3 + 32	6.4	158.3 (24.7×)	1.22	5.3 (4.3×)
`DenseNet`	3 + 64	58.5	350.3 (6.0×)	3.23	13.25 (4.1×)
`ShuffleNetV2`	3 + 24	3.0	122.5 (40.8×)	1.46	4.62 (3.2×)
`EfficientNet`	3 + 32	8.8	160.7 (18.3×)	1.78	5.86 (3.3×)

Given the prior empirical mathematical observation that the Joint Input-Feature architecture frequently underperforms the Input-Only architecture in terms of exact AUROC against algorithmic gradient attacks due to the Semantic Alignment Paradox, the computational overhead analysis provides a conclusive and strong directive. The Input-Only monitoring configuration is mathematically proven to be not only significantly more robust against L_∞ perturbations but also vastly more physically efficient. The massive quadratic mathematical inflation in FLOPs and inference latency required to formally model the concatenated semantic features yields absolutely no justifiable diagnostic safety benefit. This reality renders the lightweight Input-Only topology the strictly supe-

rior architectural paradigm for realistic deployment in highly latency-constrained, edge-computed autonomous systems.

6 Conclusion and Outlook

The widespread deployment of deep Convolutional Neural Networks within the perception pipelines of autonomous driving systems fundamentally relies on the mathematical robustness and operational reliability of these highly parameterized models. As this thesis has systematically demonstrated, state-of-the-art traffic sign recognition classifiers remain critically vulnerable to visually inconspicuous, mathematically optimized adversarial perturbations. Ensuring passenger and pedestrian safety in unconstrained environments necessitates the development of reactive, parallel security mechanisms capable of detecting these anomalies in real-time without compromising the baseline performance of the deployed models. This concluding chapter synthesizes the foundational methodologies developed in this research, explicitly details the theoretical and empirical conclusions drawn from the rigorous ablation study, acknowledges the architectural limitations of the current framework, and proposes highly specific, mathematically grounded avenues for future research.

6.1 Summary of Methodological Contributions

This thesis proposed, engineered, and rigorously evaluated a modular runtime safety monitor based on a multi-scale Glow normalizing flow. By framing adversarial attack detection and Out-of-Distribution (OOD) identification purely as an exact density estimation problem, the proposed generative system operates entirely independently of the target classifier's potentially compromised terminal logits and softmax confidence distributions. The runtime monitor evaluates incoming optical image tensors and assigns an exact mathematical anomaly score based exclusively on the negative log-likelihood of the sample under a complex probability distribution learned strictly from clean, unperturbed training data.

The central scientific contribution of this research was the formulation and execution of a comprehensive ablation study investigating the optimal topological input space for generative anomaly detectors. The research directly and mathematically compared two divergent paradigms. The first paradigm, the Input-Only architecture, strictly modeled the probability density function of the raw pixel space. The second paradigm, the novel Joint Input-Feature architecture, extracted intermediate, high-level semantic representations from deep layers of frozen target classifiers via dynamic computational graph hooks. These extracted semantic tensors were spatially interpolated, normalized using global empirical dataset variance, and concatenated with the raw image. The theoretical motivation driving this joint topological approach was to construct a holistic safety envelope capable of simultaneously detecting low-level statistical irregularities and high-level semantic inconsistencies.

The empirical evaluation was conducted across a highly diverse set of traffic sign benchmarks, encompassing the pictographic German Traffic Sign Recognition Benchmark (GTSRB), the typo-

graphic LISA Traffic Sign Dataset, the high-resolution DFG dataset, and the purely synthetic Synset dataset. To ensure robust findings independent of specific network topologies, the generative monitors were evaluated alongside an array of target architectures, ranging from heavy residual networks like ResNet18 to highly efficient, edge-optimized models like MobileNetV2. These models were evaluated under a strict suite of optimal first-order adversarial attacks including the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and the parameter-free Auto-Projected Gradient Descent (APGD).

6.2 Synthesis of Empirical and Theoretical Findings

The results of the extensive empirical evaluation yielded yielded significant and partially counter-intuitive insights into the internal mathematical dynamics of generative anomaly detection.

6.2.1 The Efficacy of Input-Only Density Estimation

First, the evaluation conclusively demonstrated that exact density estimation via normalizing flows provides a highly effective mathematical framework for runtime monitoring. The Input-Only architectural configuration consistently achieved exceptional Area Under the Receiver Operating Characteristic (AUROC) scores and rigorously low False Positive Rates at a 95% True Positive Rate (FPR@TPR95) across multiple disparate datasets. By accurately learning the natural continuous manifold of benign traffic signs, the multi-scale Glow architecture successfully identified the high-frequency structural disruptions induced by mathematically bounded adversarial perturbations. These optimized perturbations, while visually imperceptible, fundamentally distort the natural pixel-level covariance structure of the image. This disruption causes the exact density estimator to deterministically map the adversarial samples into the extreme low-density tails of the latent Gaussian space, allowing for distinct mathematical separation.

6.2.2 The Semantic Alignment Paradox

However, the core theoretical findings of the extensive ablation study revealed that the conceptually appealing Joint Input-Feature modeling configuration universally underperforms the simpler Input-Only approach. While concatenating intermediate semantic feature maps was hypothesized to improve robustness by exposing internal network contradictions, the empirical data demonstrated a severe degradation in detection efficacy against standard digital gradient attacks.

This counter-intuitive phenomenon is mathematically attributed to the Semantic Alignment Paradox. When an iterative white-box attack successfully crosses a target classifier's linear decision boundary, the internal feature representations are forcefully optimized to mimic the dense semantic profile of the incorrect target class. Because the normalizing flow is mathematically optimized to learn the joint distribution of clean images and their corresponding clean feature activations across all classes, these successfully manipulated adversarial feature activations paradoxically align with the flow's learned semantic distributions. The generative model essentially perceives the heavily perturbed features as a highly natural, statistically probable representation of the adversarial target

class. Consequently, the concatenated semantic feature maps mathematically mask the underlying pixel-level anomaly, artificially inflating the assigned exact likelihood score and collapsing the AUROC separation margin.

6.2.3 Computational Viability and the Curse of Dimensionality

Furthermore, the computational overhead analysis decisively favored the Input-Only monitor, proving the joint architecture to be fundamentally unsuited for real-time deployment. Concatenating massive intermediate feature channels, such as extracting 256 channels from a deep residual block, with a standard 3-channel RGB image exponentially inflates the dimensionality of the tensor processed by the normalizing flow. This massive high-dimensional volume severely exacerbates the curse of dimensionality in continuous density estimation, causing the latent space representations to become exceedingly sparse and diluting the mathematical precision of the exact log-likelihood score.

Beyond topological sparsity, this extreme dimensionality drastically inflates the Floating Point Operations (FLOPs) required for a single inference pass. The weight matrices within the invertible 1×1 convolutions scale quadratically with the channel depth. In practical deployment scenarios, such as edge-computed autonomous vehicular systems where strict micro-second latency constraints are paramount for physical safety, the Joint Input-Feature approach introduces a prohibitive computational and memory burden. Because this massive computational cost is paired with a degraded detection capability, the Input-Only flow architecture is conclusively demonstrated to be the highly preferable architecture for detecting digital adversarial anomalies.

6.3 Limitations of the Current Architecture

While the theoretical and empirical findings of this thesis provide strong, mathematically grounded directives for the architectural design of parallel runtime monitors, several specific limitations within the current evaluation framework must be rigorously acknowledged.

6.3.1 Constraint to the Digital Threat Model

The primary analytical limitation lies in the strict boundaries of the evaluated adversarial threat model. The experiments were deliberately constrained to digitally bounded attacks executed directly within the numerical limits of the image tensor. These specific attacks distribute mathematically optimal, high-frequency noise across the entire spatial resolution of the image. The Input-Only monitor excels precisely at detecting this specific type of widespread statistical disruption.

However, this evaluation framework does not encompass spatially localized or structurally semantic attacks, such as physical adversarial patches, strategically placed physical stickers, or natural optical phenomena like targeted shadowing. Such localized physical perturbations might not alter the global pixel-level covariance statistics sufficiently to trigger an Input-Only density estimator, yet they fundamentally alter the semantic meaning of the physical traffic sign. The limitations of evaluating strictly within the digital domain dictate that the broader applicability of the Input-Only

flow monitor against real-world, spatially constrained physical attacks remains an unverified open question.

6.3.2 Vulnerability to Adaptive White-Box Adversaries

Furthermore, the current threat model strictly assumes that the adversary only possesses white-box access to the target classifier, while remaining entirely ignorant of the parallel runtime monitor. In a truly worst-case security scenario, a highly sophisticated adversary would possess full white-box access to both the classifier and the normalizing flow. An adaptive attacker could theoretically formulate a joint loss function that simultaneously minimizes the cross-entropy loss of the classifier while maximizing the exact log-likelihood output of the generative flow:

$$\min_{\delta} \left[\mathscr{L}_{CE}(\theta, x+\delta, y_{target}) - \lambda \log p_X(g_\phi(x+\delta)) \right] \quad [6.1]$$

By optimizing this joint objective subject to the imperceptibility bound, the adversary would force the image to cross the decision boundary while strictly maintaining a high probability density under the flow's parameters. Evaluating the resilience of the exact density estimator against such mathematically formulated adaptive attacks is crucial for establishing rigorous theoretical security bounds in future deployments.

6.4 Outlook and Future Research Directions

The conclusive insights drawn from the failure of the Joint Input-Feature architecture and the success of the Input-Only architecture open several highly promising, mathematically complex avenues for future investigation within the domain of autonomous systems security.

6.4.1 Evaluation Against Localized Physical Patches

The most immediate logical extension of this research is the rigorous evaluation of both flow configurations against localized, physical patch attacks. While the Joint Input-Feature architecture struggled against widespread digital noise due to the Semantic Alignment Paradox, its heavy reliance on deeper semantic features might prove highly advantageous against physical patch attacks. When an adversarial perturbation is confined to a small spatial region, such as a brightly colored sticker on a Stop sign, the raw pixel statistics of the overall image remain largely benign, potentially bypassing the Input-Only monitor. However, this localized patch drastically alters the deep, spatially pooled feature activations. Investigating whether the joint monitor fundamentally outperforms the input-only monitor under a localized physical threat model is a critical next step for comprehensive safety monitoring.

6.4.2 Conditional Density Estimation Paradigm

To mathematically circumvent the curse of dimensionality inherent to concatenating massive feature tensors, future research should explore the paradigm of Conditional Normalizing Flows. Instead

of modeling the incredibly high-dimensional joint probability, the generative architecture could be topologically redesigned to model the conditional probability of the image given the semantic features.

In a conditional flow framework, the extracted semantic features are not concatenated into the main bijective processing stream. Instead, they are fed exclusively into the non-invertible context networks of the affine coupling layers to dynamically modulate the scale and translation parameters of the raw image space:

$$h_b^{(out)} = h_b \odot \sigma(s(h_a, A_l)) + t(h_a, A_l) \quad [6.2]$$

By keeping the primary bijective path strictly constrained to the three-channel image domain, this conditional topology entirely prevents the exponential inflation of the invertible weight matrices. This topological shift vastly reduces the required FLOPs and mitigates latent space sparsity, while theoretically retaining the diagnostic power of observing the network's internal semantic state.

6.4.3 Invertible Dimensionality Reduction

Finally, to further mitigate the computational overhead associated with processing deep feature maps, future iterations of this architecture should investigate the integration of invertible dimensionality reduction techniques. Applying mathematically tractable transformations, such as Principal Component Analysis or specifically engineered learnable linear bottleneck layers, to the intermediate feature maps prior to spatial alignment could significantly compress the semantic space. This compression could allow the generative flow to model semantic inconsistencies without requiring the processing of hundreds of redundant, highly correlated feature channels.

In conclusion, this thesis establishes a mathematically rigorous foundation demonstrating that while architectural complexity in anomaly detection is conceptually and intellectually appealing, simpler, highly optimized models often provide strictly superior robustness against specific threat models. By exhaustively proving the algorithmic efficacy and absolute computational superiority of Input-Only normalizing flows for digital attack detection, this work provides a scalable, mathematically sound mechanism for securing the next generation of autonomous perception systems against visually imperceptible adversarial threats.

A List of Figures

B List of Tables

C Bibliography

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, pages 274–283. PMLR, 2018.

[2] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[3] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[4] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[5] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. In *International Conference on Learning Representations (ICLR)*, 2017.

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1625–1634, 2018.

[7] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[8] T. Gebhart and P. Schrater. Characterizing the topological geometry of adversarial examples. In *International Conference on Learning Representations (ICLR) Workshops*, 2019.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NeurIPS)*, volume 27, 2014.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[11] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[13] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[14] X. Hu et al. A comprehensive survey on runtime monitoring of deep neural networks. *arXiv preprint arXiv:2209.00665*, 2022.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4700–4708, 2017.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in neural information processing systems (NeurIPS)*, volume 28, 2015.

[17] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems (NeurIPS)*, volume 30, 2017.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[19] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[21] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems (NeurIPS)*, volume 30, 2017.

[22] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in neural information processing systems (NeurIPS)*, volume 31, 2018.

[23] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[24] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. In *Advances in neural information processing systems (NeurIPS)*, volume 33, pages 21464–21475, 2020.

[25] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[27] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[28] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

[29] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2574–2582, 2016.

[30] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. In *arXiv preprint arXiv:1906.02994*, 2019.

[31] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations (ICLR)*, 2019.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4510–4520, 2018.

[33] C. S. Sastry and S. Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning (ICML)*, pages 8491–8501. PMLR, 2020.

[34] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1453–1460. IEEE, 2011.

[35] Y. Sun, Y. Ming, X. Zhu, and Y. Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning (ICML)*, pages 20827–20840. PMLR, 2022.

[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[37] D. Tabernik and D. Skočaj. Deep learning for large-scale traffic-sign detection and recognition. *IEEE transactions on intelligent transportation systems*, 21(4):1427–1440, 2019.

[38] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[39] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, pages 7472–7482. PMLR, 2019.

[40] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15345–15354, 2022.