A Unified Benchmark for Out-of-Distribution Detection for Autonomous Driving Systems

Xiangyu Li
SeysoAI
Suzhou, China
xiangyu.li3@mail.mcgill.ca

Jingyu Zhang*
Department of Electronic Engineering and Computer Science, Hong Kong Metropolitan University
HongKong, China
fzhang@hkmu.edu.hk

Jacky Keung
Department of Computer Science, City University of Hong Kong
HongKong, China
jacky.keung@cityu.edu.hk

Xiaoxue Ma
Department of Electronic Engineering and Computer Science, Hong Kong Metropolitan University
HongKong, China
kxma@hkmu.edu.hk

Yihan Liao
Department of Computer Science, City University of Hong Kong
HongKong, China
yihanliao4-c@my.cityu.edu.hk

Abstract

Autonomous Driving Systems (ADS) can fail when they encounter inputs that differ from their training data, known as out-of-distribution (OOD) conditions. Such OOD inputs lead ADS to make incorrect driving decisions, resulting in serious safety risks. Reliable OOD detection is therefore essential for enhancing system robustness and preventing hazardous behavior. However, existing literature in the autonomous driving field examines a relatively narrow scope of OOD detectors (e.g., reconstruction-based only) under limited OOD conditions.

To address these limitations, we present a unified benchmark that evaluates nine widely used OOD detectors on both simulated and real-world driving images. Specifically, we assess detectors from four families: reconstruction-based, density-based, distance-based, and Generative Adversarial Network (GAN)-based. To examine generalization ability, we propose an automated evaluation pipeline with statistical analysis that evaluates OOD detectors by automatically generating OOD samples (three weather transformations and four adversarial attacks) and assessing them based on both effectiveness (AUC-ROC, AUC-PR, F1 scores at five threshold values) and efficiency (inference time).

Our experiments reveal that distance-based detectors consistently deliver the highest overall performance and robustness across datasets and all OOD scenarios, followed closely by density-based detectors. In contrast, reconstruction-based and GAN-based methods exhibit greater performance fluctuations and reduced reliability under adversarial and complex perturbations. These findings suggest that OOD detection in autonomous driving benefits most from methods that leverage feature-space distance or probabilistic density estimation, as they capture intrinsic data distribution properties more effectively.

Overall, our work guides the selection of OOD detectors for real-world ADS deployment and emphasizes the importance of robustness, generalization, and inference efficiency. Our implementation is available at this `link`.

CCS Concepts

• **Software and its engineering → Software testing and debugging**.

Keywords

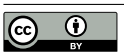
out-of-distribution detection, autonomous driving, software testing, benchmarking, machine learning systems.

ACM Reference Format:
Xiangyu Li, Jingyu Zhang, Jacky Keung, Xiaoxue Ma, and Yihan Liao. 2026. A Unified Benchmark for Out-of-Distribution Detection for Autonomous Driving Systems. In *7th ACM/IEEE International Conference on Automation of Software Test (AST '26), April 13–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3793654.3793758

1 Introduction

Autonomous driving systems (ADS) are widely expected to enhance road safety, reduce traffic congestion, and extend mobility to underserved populations [26, 44]. A remarkable proportion of ADS relies on autonomous driving models to understand the vehicle surroundings and propose corresponding driving decisions [3, 6, 13]. Most autonomous driving models are trained under various scenarios to ensure the model generalization [8, 10, 38]. However, in deployment, these models must handle a much broader range of environmental and adversarial image transformations. Inputs under adverse weather conditions, varying brightness, or adversarial perturbations may deviate significantly from the training distribution. Without awareness of such distribution shifts, models may produce unreliable predictions, such as a remarkable derivation of steering angle from the model's normal output that can cause serious safety

*Corresponding author



AST '26, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-2476-3/26/04
https://doi.org/10.1145/3793654.3793758

concerns and risks [4, 5, 12, 17, 18]. As such, detecting OOD inputs becomes a crucial safeguard mechanism. Prompt OOD detection could raise the software reliability by catching inputs the model was never trained for, and preventing silent data-driven faults that ordinary code tests miss. When OOD conditions appear, OOD detectors can inform the ADS to propose a protection strategy, either by issuing warnings, reducing control authority, or engaging fallback systems to reduce the risk caused by OOD conditions. OOD detectors continuously check incoming data, which could reduce potential ADS failure and safeguard the stability and reliability of ADS.

In software testing fields, tests can be classified into different categories according to test goal, such as functional tests used to verify the system's functionality, and non-functional tests that evaluate the performance-related aspects, like latency, reliability. Our study acts as a robustness test [1] to evaluate how detectors behave under distributional shift, rare events, and unexpected inputs. OOD detection test reveals the limitations of current OOD detectors and contributes to designing more robust architectures.

Due to the variation of OOD scenarios, a massive number of OOD detection techniques have been proposed in prior studies. Most OOD detectors were developed for a specific type of OOD scenario, such as Pinggera *et al.* [30] proposed an algorithm that was designed to identify small obstacles as the OOD factor, and SelfOracle [36] was designed to identify adverse weather images. Other studies mainly focus on detecting OOD images on multiclass classification tasks, where OOD images are defined as images not in the current category or class, like Peng *et al.* [29] and Ruff *et al.* [31] proposed OOD detectors on MNIST [15] and CIFAR [24] datasets for identifying whether samples are placed in the correct class. However, OOD conditions in autonomous driving fields are greatly different from traditional classification datasets. Most ADS require selecting OOD detectors with great robustness that could identify various OOD conditions at a fast speed in real-time autonomous driving scenarios.

To summarize OOD detectors' features and capabilities, Yang *et al.* [48] define three OOD detection families in their paper: reconstruction-based, density-based, and distance-based. We added GAN-based as a fourth family in our study, which was first introduced by Zenati *et al.* [49], where they found that GAN-based models could show state-of-the-art results in the OOD detection scenario. The family of each detector is defined based on the detector's mechanism for determining OOD samples. Specifically, reconstruction-based family reconstructs the input sample and uses the reconstruction loss to determine OOD samples, detectors in the density-based and distance-based families detect OOD samples based on the density and distance of features in the latent space, respectively. GAN-based detectors rely on the generative adversarial network to identify OOD samples.

Stocco *et al.* [36] have proposed a study to compare different detectors on handling different weather, luminance, and anomalous behavior (out-of-road and collision) images collected by a camera in the center of a vehicle. Hussain *et al.* [21] extended SelfOracle [36] to further consider the strategy when an OOD situation happens. However, their detector selection is unbalanced (4 reconstruction-based, 1 distance-based). Also, they did not consider the detectors' performance in detecting adversarial attacked images.

Therefore, ensuring the system's reliability in handling exceptional situations is a key challenge in the validation of ADS. Traditional software testing techniques assume deterministic logic, whereas modern perception modules rely on learned models that may fail silently under unforeseen inputs. In this paper, we present a validation framework that treats OOD detection as a software testing problem. Using our constructed automated standardized test pipeline, we design experiments to evaluate the performance and explore characteristics of detectors and their corresponding family. Our pipeline only requires InD driving images as input and it can automatically generate test samples from various OOD conditions and provide a comprehensive quantitative analyze of detectors OOD detection ability.

Compared with previous studies, we applied adversarial attacks on the evaluation dataset to assess the detectors' ability to identify adversarial-attacked images. We also included GAN-based OOD detectors in our comparison that are commonly used in general OOD detection tasks. OOD detectors were trained on the InD images in an unsupervised manner and were tested on the test dataset that contains both InD and OOD images. Our contributions are summarized as follows:

- We design an automated model-independent evaluation pipeline and apply it to nine widely used OOD detectors drawn from the four major families: reconstruction-based, density-based, distance-based, and GAN-based, so that every method is trained and tested under identical pre- and post-processing, permitting a genuinely fair cross-family comparison. Our pipeline can be deployed as an automated tool to generate OOD samples and help evaluate the accuracy and robustness of the target detector.
- We create a large test dataset that covers normal images, weather-transformed images, black-box adversarial-attacked images, and white-box adversarial-attacked images collected from both the driving simulator and real-world scenarios. Due to the large number of OOD images and scenarios included in the test dataset, the findings of testing OOD detectors on the test dataset can be extended to a more general domain.
- We reveal that distance-based detectors perform best overall for OOD detection due to strong separability and stability, while density-based detectors show compatible robustness. However, reconstruction-based and GAN-based detectors show serious volatility under various OOD scenarios.

2 Related Work

2.1 OOD Data

OOD data broadly denotes inputs whose statistics deviate meaningfully from those seen during training. In the field of autonomous driving, prior work has explored weather-induced shifts in CARLA [16]. Pinggera *et al.* [30] have proposed the Lost & Found dataset that includes both static and moving on-road obstacles as OOD factors. Stocco *et al.* [36] were the first to improve the Udacity Simulator by introducing multiple environment factors to the simulator, such as weather, day/night, and terrain. This improvement enriches the diversity of the collected driving images and contributes to testing the robustness of OOD detectors.

Several studies have involved adversarial-attacked data as OOD data. Li *et al.* [45] implemented white-box, grey-box, and black-box to assess the adversarial robustness, domain generalization, and dataset biases. Zhao *et al.* [52] examined the impact of adversarial perturbations and assessed defense approaches. However, previous work mainly focuses on a specific type of OOD detectors with limited OOD conditions. Studies on adversarial attacks, like the work proposed by Zhao *et al.* [52], explicitly state the effects of each adversarial attack, which lack generalization in selecting OOD defense approaches. OOD defense approach refers to reducing the effects on ADS caused by OOD images. Common OOD defense approaches are like training the model with adversarial processed images, and denoising the input images. Our work demonstrates a more robust result with a broader view of OOD detector approaches.

2.2 Autonomous Driving Systems

Pure-vision end-to-end ADS has developed into various structures in recent years. Pure-vision end-to-end ADS means models only use images to make driving decisions and directly control the vehicle. The models within this scope mainly take one or more images from the center or side cameras of vehicles as input to predict vehicle control variables such as steering angles, throttle, or accelerations. DAVE-2 [6] pioneered this line by coupling a lightweight proposal head with an attribute-prediction branch, learning generic road semantics while jointly regressing steering commands. Epoch [41] retains the same principle. The convolutional feature extraction, followed by dense steering layers, streamlines training with a single continuous loss. Latent work expanded prior models by adjusting training steps or adding transformer blocks to improve the robustness and precision. ChauffeurNet [3] adds command-conditional branches and imitation learning, CILRS [13] injects route instructions to improve intersection handling, and LBC [11] defines a privileged teacher to teach the purely vision-based agent and enhance the performance. We implemented the ADS to help generate adversarial-attacked images and assessed the attack strength.

2.3 OOD Detection Approach

We classified OOD detectors into four categories based on Yang *et al.*'s [48] and Zenati *et al.*'s study [49]: reconstruction-based, density-based, distance-based, and GAN-based.

Simple Autoencoder (SAE) is the simplest reconstruction-based detector to identify OOD images. An advanced reconstruction-based OOD detector, VAE proposed by Kingma & Welling use latent likelihood to flag OOD data on top of that. Laakom *et al.* [25] have proposed a Convolution Autoencoder, which is trained to reconstruct positive input and learning representations on a multimodal macroinvertebrate image classification dataset. Siddalingappa *et al.* [34] also combined convolutional neural networks with autoencoders to identify CT scan images in the medical field.

Density-based detectors are frequently used for classification tasks. Erdil *et al.* have proposed a task-agnostic KDE[19] on the traditional multiclass image classification task. Scrucca *et al.* proposed the entropy-based GMM[33] to identify OOD on both simulated and real-world datasets under significant noise effects. Peng *et al.* [29] provided an unbiased tractable estimator using the Monte Carlo-based importance sampling technique to detect OOD images on CIFAR-100 [24] and ImageNet-1K[14].

In terms of distance-based detectors, Deep Support Vector Data Description [40] involves a deep neural network that extracts features and helps to detect OOD data. It shows an impressive result on MNIST [15] and CIFAR-10. Papernot *et al.* exploited the structure of deep learning and introduced the Deep K-Nearest Neighbor (DeepKNN) that combines neighbors with representations of learned data and separates them according to distance. They evaluate the method on multiclass classification datasets. Zhang *et al.* [50] proposed the DeepRoad OOD detector that involves VGG [35] and Principal Component Analysis(PCA) for detecting OOD images in the autonomous driving field. For GAN-based detectors, Schlegl *et al.*'s AnoGAN[32] measures reconstruction error in GAN latent space, its encoder variant (F-AnoGAN) accelerates the prediction process. Wolleb *et al.* [46] proposed DeScarGAN learns jointly from both normal and anomalous images in the training step to distinguish chest X-ray images. Akcay *et al.* [2] proposed Ganomaly, which learn knowledge from high-dimensional space and inference on the latent space with two encoders.

Transform-based detectors are also used to detect OOD conditions like Koner *et al.* [23]. But due to the increase in the model's complexity and a large demand for computational resources, they are not widely leveraged as a part of ADS to detect OOD in real-time cases. In this paper, we have implemented several detectors within each family to evaluate the detectors themselves and summarize general findings that can be applied to the entire detector family. Our selected OOD detectors are explicitly described in Section 3.3.

2.4 OOD Detection Benchmarks

OOD detection is a long-standing topic that has been explored in different fields. OpenOOD is a benchmark for general OOD detection proposed by Yang *et al.* [47]. It includes common image datasets like CIFAR-10, CIFAR-100, and ImageNet-1k, but lacks resources for OOD detection in the autonomous driving field. Other benchmarks like COCO-O proposed by Mao *et al.* [28] mainly focus on the nature distribution shift of OOD data, while in the autonomous driving scenario, OOD conditions are more complicated. Benchmarks on the autonomous driving case may lack the variation of the OOD condition, like the "SegmentMeIfYouCan" benchmark proposed by Chan *et al.* [9] concentrate on using the segmentation strategies to detect unseen objects and road obstacles.

Compared with previous benchmarks, our study broadened the OOD conditions (weather-transformed and adversarial-attacked) and the driving scenario (driving simulator and real-world) to enable a more comprehensive and realistic validation of detectors' robustness and enhance the test coverage to provide stronger evidence of the system's reliability under non-nominal, safety-critical conditions.

3 Study Design

3.1 Overview

The study follows a three-stage pipeline represented in the Figure 1: OOD dataset construction, OOD detection, and Evaluation. In stage 1 (red), we construct evaluation sets with two main OOD scenarios: weather-transformed and adversarial-attacked scenarios. The

former contains images under fog, rain, and snow conditions. The adversarial-attacked scenario perturbs the InD images with black-box and white-box attacks. Stage 2 (green) is our main assessment process that tests OOD detectors on the generated OOD dataset. Each OOD dataset (each dataset contains one type of OOD image) is separately scored on detectors from our defined families: density-based, distance-based, reconstruction-based, and GAN-based. An adaptive threshold τ defines InD and OOD images from OOD predictions generated from the OOD detectors training stage. Stage 3 (blue) evaluates the OOD detectors in terms of effectiveness and efficiency. This unified pipeline enables a fair, end-to-end comparison of detectors across both weather and adversarial attack conditions.

3.2 OOD Dataset Construction

3.2.1 Weather Transformation. We adopt three weather transformations to construct the unexpected driving condition. Specifically, we implement rain, snow, and fog effects. For the implementation details, please refer to Section 4.2. Examples of weather-transformed images are shown in Figure 2.

3.2.2 Adversarial Attack. To study OOD detectors' sensitivity to different adversarial attack techniques and to observe whether the category of attack techniques affects the performance of OOD detectors, we implemented two white-box attack techniques and two black-box attack techniques in our study. An equal number of attack techniques under two common test categories can reduce the test bias. White-box attacks in the machine learning field assume the knowledge of the network [51]. For the white-box attack technique, we provide the gradient and the loss collected in the training stage.

For white-box adversarial attack techniques, we chose a very efficient attack strategy: FGSM[20], which only needs the input gradient with a single back propagation step. Adversarial perturbations can be directly added to the image. We also chose an enhanced attack technique, PGD[27], that iterates the attack step multiple times to maximize the model's loss. We adopted Salt and Pepper Noise (SP) [22] and Simultaneous Perturbation Stochastic Approximation (SPSA) [43] as our black-box adversarial attack. SP benefits from its simplicity, which does not require querying the model. SPSA is a stronger and effective gradient-free method. It applies attacks with a strategic search on the input space instead of random noise.

Fast Gradient Sign Method [20] (FGSM) perturbs images in the direction that maximally increases the model's loss, causing model failure on predictions. It directly adds perturbation to images, affecting the model's prediction based on gradient loss.

Projected Gradient Descent [27] (PGD) is an iterative, white-box adversarial-attack method that refines the one-step FGSM approach into a multi-step optimization routine. Starting from the original image. PGD repeatedly affects the input in the direction of the loss gradient, taking small steps of size α that maximize the model's classification loss.

Salt and Pepper Noise [22] (SP) is a simple, black-box adversarial attack technique used in adversarial robustness testing. It randomly alters a fixed proportion of image pixels by setting them to the minimum or maximum possible value, simulating impulse noise.

Simultaneous Perturbation Stochastic Approximation [43] (SPSA) is a query-efficient, gradient-free black-box adversarial attack approach. At every iteration, it requires the *two* model queries, independent of the input dimension.

In the Table 1, we denote $\mathbf{x}$ as the InD input sample and y its true label. The model parameters are represented by $\boldsymbol{\theta}$, and $\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)$ is the loss function used for training or evaluation. The adversarial perturbation magnitude is controlled by ϵ, defining the maximum allowable change under an ℓ_∞ norm constraint. For iterative attacks such as PGD, α is the step size and $\Pi_{\mathcal{B}\epsilon(\mathbf{x})}$ denotes the projection operator that ensures the perturbed sample remains within the ϵ-ball centered at $\mathbf{x}$. In the SP attack, $I(x, y)$ is the original pixel intensity at position (x, y), with Imax and $I_{\min}$ indicating the maximum and minimum intensity values, and p_s, p_p being the probabilities of salt and pepper noise, respectively. For the SPSA attack, c_k is a small finite-difference scaling factor, Δ_k is a random perturbation vector, and $\hat{\mathbf{g}}_k$ denotes the stochastic gradient estimate derived from the difference in losses computed at symmetrically perturbed inputs.

3.3 OOD Detectors

This section elaborates on the OOD detectors implemented in our study. We described the structure of each detector, as well as the mathematical mechanism used to distinguish OOD data from InD data in Table 2. In this section, we defined $\mathbf{x}^{(i)}$ as the InD image, $\hat{\mathbf{x}}^{(i)}$ as the OOD images. For each input image i (out of N total), we compute a loss value l_i for optimizing OOD detectors in the training process and use it to compare with threshold τ to determine whether the image is OOD in the inference process. L is the average train loss of the OOD detector obtained as $L = \frac{1}{N}\sum_{i=1}^{N} l_i$ in the training process to optimize the OOD detector and obtain the threshold τ value according to the proportion rate ε. L can be computed in various ways according to the detector family type. For example, reconstruction-based detectors compute L as the difference between the input image and the reconstructed image.

We pick τ according to the L where ε proportion of the images in the training set have loss exceeding the τ. Detailed results and proportion rate are illustrated in Tables 5 and 6. N is the number of input samples, and i is the index of the current image. All the OOD detectors are trained on the InD dataset, and inference on OOD dataset.

3.3.1 Reconstruction-based Approach. Reconstruction-based methods train a model to reproduce InD samples, using reconstruction quality as a measure of normality. During inference, a model reconstructs input images and computes the reconstruction error, which serves as the anomaly score. Intuitively, OOD inputs are more challenging to reconstruct, resulting in higher reconstruction errors. A Gamma distribution can approximate the distribution of reconstruction differences. Representative methods include the **Simple Autoencoder (SAE)**, which learns a deterministic encoder–decoder mapping for reconstruction, and the **Variational Autoencoder (VAE)**, which models inputs in a probabilistic latent space using variational inference to capture uncertainty in reconstruction.

3.3.2 Density-based Approach. Density-based OOD detectors rely on the estimated probability density of InD data in a latent space.

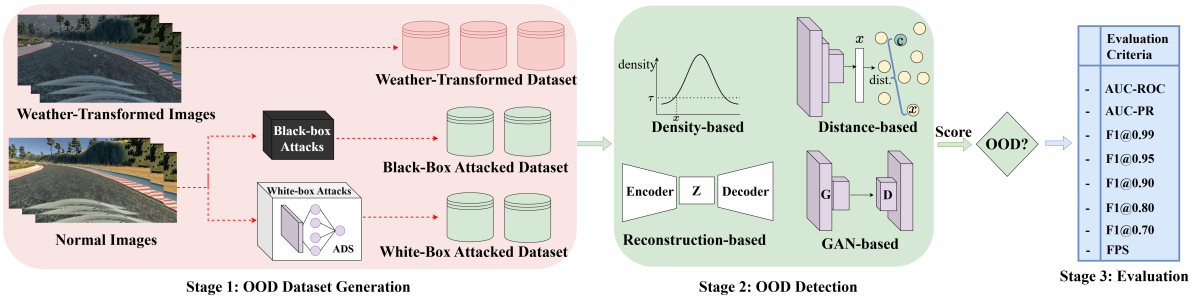


Figure 1: The pipeline for evaluating OOD detectors.

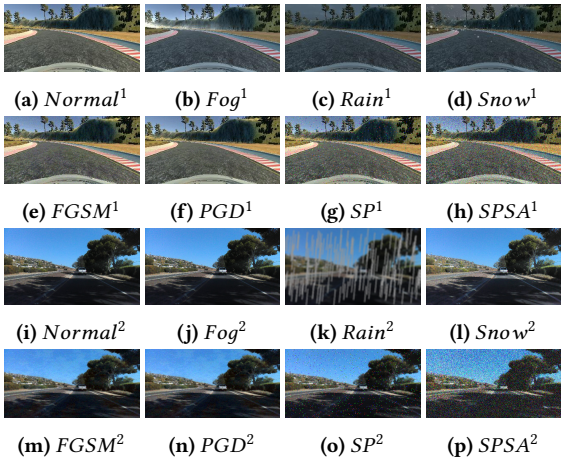
Attack	Category	Description	Formula
FGSM	White-box	One-step gradient-sign attack that perturbs the input in the direction that maximally increases the loss.	$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y))$
PGD	White-box	Iterative FGSM with projection onto the ϵ-ball to enforce perturbation bounds.	$\mathbf{x}^{(t+1)} = \Pi_{\mathcal{B}_\epsilon(\mathbf{x})}(\mathbf{x}^{(t)} + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}^{(t)}, y)))$
SP	Black-box	Randomly replaces pixels with $I_{\max}$ or $I_{\min}$ with probabilities p_s, p_p.	$I'(x,y) = \begin{cases} I_{\max}, & \text{w.p. } p_s \\ I_{\min}, & \text{w.p. } p_p \\ I(x,y), & \text{w.p. } 1 - p_s - p_p \end{cases}$
SPSA	Black-box	Gradient-free method estimating gradients by finite differences over random perturbations.	$\hat{\mathbf{g}}_k = \frac{\mathcal{L}(\mathbf{x}_k + c_k \Delta_k) - \mathcal{L}(\mathbf{x}_k - c_k \Delta_k)}{2c_k} \Delta_k^{-1}$

Table 1: Summary of adversarial attack methods.

OOD Detector	Category	Description	Formula	Classification
SAE	Reconstruction-based	Learns a low-dimensional representation and reconstructs the input; large reconstruction error indicates OOD.	$\mathcal{L}_{\text{SAE}} = \frac{1}{N} \sum_{i=1}^{N} \|x^{(i)} - \hat{x}^{(i)}\|^2, \quad \hat{x}^{(i)} = D(E(x^{(i)}))$	$s(\mathbf{x}) > \tau \Rightarrow$ OOD, $s(\mathbf{x}) < \tau \Rightarrow$ InD.
VAE	Reconstruction-based	Variational autoencoder estimating data likelihood via ELBO; lower likelihood implies OOD.	$\mathcal{L}_{\text{VAE}}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}\|\mathbf{x})}[\log p_\theta(\mathbf{x}\|\mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z}\|\mathbf{x}) \| p(\mathbf{z}))$	$-\text{ELBO}(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
KDE	Density-based	Estimates input density using kernel smoothing; samples with low density are OOD.	$\hat{p}(\mathbf{x}) = \frac{1}{Nh^d} \sum_{j=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}^{(j)}}{h}\right)$, $K(u) = \frac{1}{(2\pi)^{d/2}} e^{-\|u\|^2/2}$	$-\log \hat{p}(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
GMM	Density-based	Fits a Gaussian mixture in feature space; low mixture likelihood signals OOD.	$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$	$-\log p(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
DeepKNN	Distance-based	Computes average feature distance to k nearest training samples; large distances imply OOD.	$s(\mathbf{x}) = \frac{1}{k} \sum_{j=1}^{k} \|\phi(\mathbf{x}) - \text{NN}_j(\phi(\mathbf{x}))\|$	$s(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
DeepSVDD	Distance-based	Learns a hypersphere in feature space; samples far from the center are OOD.	$\mathcal{L}_{\text{DeepSVDD}} = \frac{1}{N} \sum_{i=1}^{N} \|\phi(\mathbf{x}^{(i)}; W) - \mathbf{c}\|^2$	$\|\phi(\mathbf{x}; W) - \mathbf{c}\|^2 > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
DeepRoad	Distance-based	Measures minimal distance to manifold prototypes (e.g., road/weather scenes); large distance $\rightarrow$ OOD.	$s(\mathbf{x}) = \min_{i=1,\ldots,m} \min_{\mathbf{u} \in \mathcal{T}} \|\mathbf{f}_i - \mathbf{u}\|_2$	$s(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
FAnoGAN	GAN-based	Uses a generator and feature extractor to measure reconstruction and feature discrepancies.	$l_{\text{AnoGAN}}(\mathbf{x}) = \|\mathbf{x} - G(z^*)\| + \lambda \|f(\mathbf{x}) - f(G(z^*))\|$	$l_{\text{AnoGAN}}(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.
Ganomaly	GAN-based	Combines adversarial, reconstruction, and latent consistency losses for anomaly detection.	$\mathcal{L}_{\text{Ganomaly}} = \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{\text{con}} \|\mathbf{x} - \hat{\mathbf{x}}\|_1 + \lambda_{\text{lat}} \|z - \hat{z}\|_1, \quad s(\mathbf{x}) = \|z - \hat{z}\|_1$	$s(\mathbf{x}) > \tau \Rightarrow$ OOD, $< \tau \Rightarrow$ InD.

Table 2: Summary of OOD detectors.

These methods typically extract deep features using a pretrained encoder (e.g., ResNet) and fit a statistical density model to them. At inference, low likelihood or density indicates that a sample lies outside the high-density region of InD data. **Kernel Density Estimation (KDE)** estimates the data distribution non-parametrically using



(a) $Normal^1$ **(b)** Fog^1 **(c)** $Rain^1$ **(d)** $Snow^1$
(e) $FGSM^1$ **(f)** PGD^1 **(g)** SP^1 **(h)** $SPSA^1$
(i) $Normal^2$ **(j)** Fog^2 **(k)** $Rain^2$ **(l)** $Snow^2$
(m) $FGSM^2$ **(n)** PGD^2 **(o)** SP^2 **(p)** $SPSA^2$

Figure 2: Sample images in the evaluation datasets ([1] Udacity dataset, [2] Dave dataset.)

kernel functions, while the **Gaussian Mixture Model (GMM)** assumes a mixture of Gaussian components, providing a parametric estimate of the feature-space density. Both methods compute an anomaly score based on the (negative) log-likelihood of the input feature.

3.3.3 Distance-based Approach. Distance-based methods measure how far a test sample's representation lies from the learned InD feature region. They assume that InD samples cluster tightly in latent space, whereas OOD samples lie farther away. **DeepKNN** measures the average distance to the k nearest neighbors in the feature space; smaller distances indicate InD behavior. **DeepSVDD** maps samples into a hypersphere by minimizing distances to a learned center, such that larger deviations indicate OOD. **DeepRoad** extends this idea by comparing distances between local patch-level features and learned manifolds, making it suitable for detecting OOD in autonomous driving scenarios.

3.3.4 GAN-based Approach. GAN-based methods employ generative adversarial networks trained only on InD data to model their distribution implicitly. During inference, the reconstruction or feature discrepancy between a real sample and its generated counterpart serves as an anomaly score. **FAnoGAN** introduces an encoder into the classic AnoGAN framework, enabling efficient mapping from image space to latent space and faster inference. **Ganomaly** further refines this framework with an encoder–decoder–encoder architecture that enforces both image and latent consistency, using adversarial, reconstruction, and latent losses to improve OOD sensitivity.

3.4 Evaluation Criteria

We measure the performance of OOD detectors in both effectiveness and efficiency. For effectiveness, we select Area under the Curve of Receiver Operating Characteristics (AUC-ROC), Area under the Curve of Precision-Recall (AUC-PR), and F1 scores at five $1-\varepsilon$ threshold values (F1@0.99, F1@0.95, F1@0.90, F1@0.80, F1@0.70) as the evaluation criteria to assess the effectiveness of OOD detectors. Note that AUC-ROC and AUC-PR are two threshold-independent metrics. F1-score F1 score is a measure of a classification model's accuracy, calculated as the harmonic mean of precision and recall: $F1 = 2\frac{Precision \times Recall}{Precison + Recall}$, where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$. Note that we denote OOD as positive and InD as negative. For all criteria, a higher score indicates a higher accuracy. We also deploy ScottKnott Effect Size Difference (ESD) test [39] to distribute a statistical analysis on the collected results, shown by using different cell colors to represent ranks in Table 5 and Table 6. We leveraged each detector to inference the evaluation dataset 5 times and collected the metrics' value as samples to perform the statistical test. To evaluate the efficiency of OOD detectors, we chose Frames Per Second (FPS) to measure the number of images the detector can process in one second. Higher FPS indicates a faster inference speed.

4 Experimental Setup

4.1 Research Questions

This section outlines the research questions investigated in this paper. **RQ1. Effectiveness of OOD Detectors Among and Across Families**: This research question studies effectiveness (1) among families: whether detectors within the same family (for example, SAE and VAE) show similar effectiveness patterns in detecting OOD images, and (2) across families: whether detectors of different families show different patterns. To clarify these, we assess the effectiveness of nine detectors (across four families) by computing AUC-ROC, AUC-PR, F1@0.99, F1@0.95, F1@0.90, F1@0.80, and F1@0.70 across seven OOD conditions (three weather transformations and four adversarial attacks).

RQ2. Robustness of OOD Detectors Across OOD conditions: This research question evaluates the robustness of each OOD detector under different OOD conditions. OOD conditions have been defined into two main scenarios: weather-transformed and adversarial-attacked. By comparing results from the same detector under different OOD conditions, we can summarize the behavior pattern of each detector when facing different OOD scenarios, as well as the detector's cross-OOD robustness.

RQ3. Time Efficiency: This research question focuses on the inference time for each OOD model. It aims to identify the most time-efficient model that balances accuracy and computational cost, making it suitable for real-world deployment in autonomous driving systems. We also want to analyze whether there is a constant pattern between the precision of the OOD detectors and the inference speed. We test each detector on an NVIDIA A100 with our test dataset and measure the FPS. Results that measure the inference speed are combined with previous precision criteria to make a convincing OOD detector selection recommendation.

4.2 Datasets and Autonomous Driving Models

We used the the Dave dataset [37] and Udacity dataset [42] in our experiments. The Dave dataset consists of real-world driving images recorded around Rancho Palos Verdes and San Pedro, California, and includes approximately 45,000 images under InD conditions with steering angles as labels. We used 60% of the total images for training and 40% for evaluation. Since this dataset only contains

images captured in noraml driving conditions, we implemented the OOD dataset generation process to automatically generate OOD samples for evaluation purpose. In this process, We used Albumentations [7] to apply weather transformations—fog, rain, and snow—to the InD evaluation images, generating 18,228 images for each weather condition.

The Udacity dataset contains approximately 10,000 images under in-distribution (InD) conditions for training and another 40,000 images under various weather conditions for evaluation, all generated using the Udacity Driving Simulator. The images were captured from the vehicle's center camera, with steering angles and weather conditions provided as labels. Since Udacity dataset provides weather labels, so we simply involved images under InD, fog, rain, and snow conditions (10,421 InD, 10,562 fog, 12,725 rain, and 13,286 snow) as weather-transformed OOD samples in the evaluation dataset.

The evaluation also included testing detectors on adversarial-attacked images. We applied different adversarial attack methods to the InD images in the OOD dataset generation process to automatically generate adversarial samples as another part of the evaluation set. Black-box attacks (SPSA, SP) were directly applied to the InD images, whereas white-box attacks (FGSM, PGD) accessed the gradients of our trained ADS models to generate adversarial samples. Each adversarial attack technique is described in detail in Section 3. For OOD detection, we labeled InD images as 0, and weather-transformed or adversarially attacked images as 1.

For the ADS models used to generate white-box adversarial samples and measure attack strength, we trained Dave2 [6] and Epoch [41]. During training, we split the data into 80% for training and 20% for validation. All images were resized to 320×180 and normalized to the range $[-1, 1]$. We adopted the mean squared error (MSE) as the training loss for both models, defined as

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left(s^{(i)} - \hat{s}^{(i)} \right)^2, \tag{1}$$

where N denotes the batch size, $s^{(i)}$ is the ground-truth steering angle for sample i, and $\hat{s}^{(i)}$ is the predicted steering angle.

The Dave2 model trained on the Dave dataset achieved a training loss of 451.74 and a validation loss of 929.13. The Epoch model trained on the Dave dataset achieved a training loss of 18.68 and a validation loss of 49.04. When trained on the Udacity dataset, the Dave2 model obtained a training loss of 0.0058 and a validation loss of 0.023, while the Epoch model achieved a training loss of 0.0147 and a validation loss of 0.0241.

We further evaluated the performance of our trained ADS models on adversarially attacked images. We calculated the mean and standard deviation of the MSE loss and summarized the results in Table 3. These results illustrate the strength of the adversarial attacks on the ADS models, which should also be considered as an important factor when evaluating OOD detectors.

4.3 OOD Detectors

We trained OOD detectors on the training dataset in an unsupervised learning manner. The image preprocessing stage involves resizing images into 160x320 and normalizing them to the range [-1, 1]. Density-based methods and part of distance-based methods (DeepKNN, DeepRoad) extract features with a pretrained Resnet or VGG model and compute the corresponding indicator to identify OOD images. Detectors are trained to converge.

Model	FGSM	PGD	SPSA	SP
Dave2[1]	0.303 ± 0.216	0.454 ± 0.271	0.153 ± 0.138	0.164 ± 0.147
Epoch[1]	0.140 ± 0.146	0.224 ± 0.175	0.103 ± 0.098	0.116 ± 0.110
Dave2[2]	0.963 ± 0.979	0.969 ± 0.986	0.960 ± 0.970	0.970 ± 0.970
Epoch[2]	19.43 ± 27.96	19.28 ± 27.86	22.74 ± 32.36	17.06 ± 25.63

Table 3: Prediction difference (Mean ± Std) of ADS under various adversarial attacks for each dataset ([1] Udacity dataset, [2] Dave dataset).

4.4 Hyperparameters

We listed all the hyperparameters that were used for ADS, adversarial attacks, and OOD detectors. Since these hyperparameters can affect our final result, we list them to help reproduce our experiment results. The dataset preprocessing step remains identical for all stages and detectors, which resizes images to 160x320 and normalizes them to the interval of [-1, 1].

4.4.1 Autonomous Driving Systems. For both our implemented ADS, we set 1000 as the train epoch, 32 as the batch size, learning rate as 1e-4, MSE loss, and Adam as the optimizer. We shuffled only the train set.

4.4.2 Adversarial Attack Techniques. We set 8/255 as the γ for both FGSM and PGD, which controls the attack strength. We also set α as 0.1 and steps t as 40 for the PGD. We set the number of iterations as 300, the c as 0.001, and the α as 0.01 for the SPSA. We set the p as 0.02, the maximum as 1, and the minimum as -1 for the SP.

4.4.3 OOD Detectors. This section presents the unique parameters of each OOD detector, as shown in Table 4. For detectors using PCA, we set the PCA dimension to 50.

Detector	Hyperparameters (per model)	PCA
SAE	Epochs = 50; Adam; lr = 1×10^{-4}	No
VAE	Epochs = 50; Adam; lr = 1×10^{-4}; z-dim = 128	No
KDE	$d = 1$	Yes
GMM	$N = 10$ Gaussian components	Yes
DeepKNN	$k = 50$ nearest neighbours	Yes
DeepRoad	$m = 1$ past frame (time window)	Yes
DeepSVDD	Epochs = 50; Adam; lr = 1×10^{-4}; z-dim = 128; weight-decay = 5×10^{-7}	No
FAnoGAN	Epochs = 30; Adam; lr = 2×10^{-4}	No
Ganomaly	Epochs = 50; Adam; lr = 2×10^{-4}; $\lambda_{\text{con}} = 50$; $\lambda_{\text{lat}} = 1$	No

Table 4: Key training configuration settings for each detector.

5 Results and Analysis

This section demonstrates our experiment results and answers the research questions proposed in Section 4.1. We listed our results in tables.

Table 7 calculates the average performance among families to demonstrate the family effects on OOD detectors. Table 8 measures the inference time of each detector that could be used to

summarize the tradeoff between precision and inference time, and it can determine whether they are suitable for real-time deployment. Table 5 shows the performance of each OOD detector on the weather-transformed dataset. This table can be used to evaluate the ability of detectors to identify OOD under different weather conditions. Table 6 represents OOD detectors' performances when facing adversarial-attacked images, and their ability to separate these OOD images from the InD images. In the experiment stage, we generated white-box attack samples with the help of different ADS (Dave2 and Epoch) and test detectors separately. These results under the same attack strategy were concatenated in the result table by averaging them. For example, the AUC-ROC for SAE on FGSM is computed by averaging the AUC-ROC obtained on samples generated using FGSM-Dave2 and FGSM-Epoch. Table 7 reports family-level means (± standard deviation) aggregated across all detectors and perturbation types.

In Table 5 and 6, we group detectors by family (reconstruction, density, distance, GAN) and evaluate them under weather-transformed (fog, rain, snow) and adversarial-attacked (FGSM, PGD, SPSA, SP) conditions on Udacity and DAVE. Metrics include AUC-ROC, AUC-PR, and thresholded F1 at 0.70/0.80/0.90/0.95/0.99. Each cell reports the mean over five runs. To reduce randomness and enable rank-aware comparisons, we apply the Scott-Knott ESD test ($\alpha = 0.05$) on each metric. We color the top three statistically significant groups (group 1 darkest purple → group 3 lightest purple). If there are at least five groups, we bold the fourth group value.

5.1 RQ1. Effectiveness of OOD Detectors Among and Across Families

Across weather-transformed OOD inputs, distance-based family is consistently the most effective. DeepKNN produces superior performances across fog, rain, and snow conditions. It retains the top three statistical groups as the decision threshold tightens, which indicates that nearest-neighbor structure in latent space yields robust separation at high confidence. Density-based methods rank second overall, with GMM and KDE frequently entering the top 2 groups, particularly for high threshold F1 (e.g., F1@0.95 and F1@0.99), where KDE shows large discrimination on two evaluation datasets. GAN-based detectors achieve moderate means but display visible rank volatility, which rarely enters the top 3 groups under rain and snow conditions. Reconstruction-based detectors are the least reliable family. Even though SAE and VAE frequently stay in the top 2 groups on the Udacity dataset, their evaluation metrics were remarkably low when it came to the real-world Dave dataset, which contains a more complicated environment.

Under adversarial conditions, the pattern persists but becomes sharper. Distance-based and density-based detectors dominate the top groups across all evaluation metrics. There is a clear performance gap between distance-based/density-based detectors and reconstruction-based/GAN-based detectors in terms of F1 score under higher thresholds.

These observations are corroborated by the family-level means in Table 7, where the distance-based methods achieve the highest AUC-ROC and AUC-PR and maintain the strongest F1 sequence from 0.70 through 0.99, while the density-based family remains competitive with lower variability than GAN-based and reconstruction-based families.

Inside each family, Detectors shared the same strength and weakness, like SAE and VAE produce strength results on weather-transformed samples in the Udacity datset and near-zero metrics in handling adversarial attack images. There is no clear pattern or constant ranking for detectors inside each family, since the in-family difference on evaluation metrics is not obvious.

> **Finding 1:** We find that distance-based detectors achieve the highest overall OOD detection effectiveness, always staying in the top three groups in statistical test results, combining strong separability (high AUC) with stability across datasets and OOD conditions. Density-based detectors provide competitive performance and maintain robustness across both weather transformation and adversarial attacks. GAN-based detectors show moderate results but with high variance. Reconstruction-based detectors consistently underperform, highlighting their limited utility for reliable OOD detection in complex visual domains.

5.2 RQ2. Robustness of OOD Detectors Across OOD Conditions

Across all OOD settings, detector families exhibit distinct robustness characteristics. Distance-based detectors are the most resilient and stable under most OOD conditions. Density-based detectors show the second-highest robustness, though they are more sensitive to adversarial attack conditions than distance-based methods. GAN-based detectors demonstrate moderate robustness, which perform consistently among most OOD conditions but show an extremely low F1@99 score(around 0.02) in some particular cases, demonstrating their obscure border between InD and OOD samples. Reconstruction-based detectors are the least robust, as their reliance on pixel-level reconstruction makes them highly sensitive to both weather transformation, adversarial perturbations, and the evaluation datasets.

Among distance-based detectors, DeepKNN achieves top statistical rankings in most OOD conditions, maintaining high AUC and stable F1 scores even at stricter thresholds such as F1@0.95 and F1@0.99, which implies nearest-neighbor relations in latent space remain unaffected by most OOD conditions. DeepRoad also reveals a similar pattern, while DeepSVDD produces weaker results. In the density-based family, GMM and KDE demonstrate comparable robustness under OOD conditions and show strong performances under white-box adversarial attack conditions, which suggests that adversarial attack type may cause fluctuation in their detection ability.

GAN-based detectors show acceptable stability in weather-transformed and adversarial-attacked datasets produced by Udacity Simulator, yet their performance becomes highly degraded and inconsistent in real-world complex conditions (Dave dataset), where F1@0.99 can fall close to zero. This indicates sample collection domain can largely affect the performance of GAN-based detectors. Finally, reconstruction-based detectors are affected the earliest and most severely across both weather and adversarial conditions. SAE and VAE show acceptable results only on weather-transformed samples from the Udacity dataset, while lagging in other conditions.

Family	Model	Fog							Rain							Snow						
		AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99
Reconstruction	SAE	0.56	0.57	0.46	0.43	0.34	0.24	0.06	0.99	0.99	0.86	0.90	0.93	0.96	0.90	0.99	0.99	0.87	0.90	0.93	0.97	0.96
Reconstruction	VAE	0.97	0.97	0.83	0.85	0.90	0.89	0.83	0.93	0.94	0.86	0.86	**0.85**	0.84	0.69	0.98	0.98	0.86	0.88	0.93	0.94	0.86
Density	GMM	0.92	**0.81**	0.76	0.80	0.85	0.89	0.92	**0.93**	**0.87**	**0.79**	**0.83**	0.87	0.91	0.94	**0.94**	0.90	**0.80**	0.83	0.88	0.91	0.95
Density	KDE	0.68	0.68	0.67	0.67	0.66	0.58	0.60	0.67	0.72	0.71	0.71	0.67	0.58	0.62	0.70	0.75	0.72	0.72	0.71	0.63	0.63
Distance	DeepKNN	**0.90**	0.78	0.82	0.86	0.89	0.89	0.76	0.92	0.82	0.85	0.88	0.91	0.93	0.90	**0.94**	**0.87**	0.85	0.88	0.92	0.94	0.96
Distance	DeepRoad	0.86	0.74	**0.74**	**0.76**	**0.80**	**0.83**	0.83	0.86	0.78	0.78	0.80	0.83	0.85	0.84	0.92	0.83	0.79	**0.81**	**0.85**	**0.88**	**0.93**
Distance	DeepSVDD	0.89	0.76	0.71	0.74	**0.80**	0.84	0.87	0.99	0.99	0.72	0.75	0.79	0.84	0.92	0.98	0.98	0.74	0.75	0.80	0.84	**0.93**
GAN	FAnoGAN	0.96	0.96	0.69	0.73	0.78	**0.83**	0.87	0.85	0.77	0.73	0.77	0.80	**0.85**	0.86	0.90	0.84	0.74	0.77	0.81	0.86	0.91
GAN	Ganomaly	0.96	0.94	0.67	0.69	0.73	0.79	**0.85**	0.90	0.83	0.72	0.73	0.76	0.82	**0.87**	0.92	0.84	0.72	0.74	0.77	0.83	0.88

(a) Evaluation result on Udacity Dataset.

Family	Model	Fog							Rain							Snow						
		AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99
Reconstruction	SAE	0.50	0.50	0.35	0.26	0.16	0.10	**0.03**	0.35	0.40	0.14	0.09	0.04	0.02	0.00	0.17	0.34	0.09	0.07	0.04	0.02	0.01
Reconstruction	VAE	0.50	0.50	0.35	0.26	0.18	**0.12**	0.05	0.15	0.33	0.06	0.04	0.02	0.00	0.00	0.25	0.37	0.14	0.10	0.06	0.04	0.01
Density	GMM	0.50	0.50	0.39	0.31	**0.19**	0.11	0.03	1.00	1.00	0.86	0.90	0.95	0.97	0.98	0.78	0.79	0.72	0.69	0.61	0.50	0.27
Density	KDE	0.52	0.51	0.67	0.66	0.52	0.34	0.10	**0.89**	**0.90**	0.67	**0.67**	**0.80**	**0.77**	0.70	0.85	0.84	0.67	0.67	0.79	0.71	0.61
Distance	DeepKNN	0.50	0.50	**0.37**	**0.28**	0.17	0.09	0.02	1.00	1.00	0.87	0.91	0.95	0.97	0.99	0.67	**0.63**	0.59	0.50	0.32	0.17	**0.04**
Distance	DeepRoad	0.52	0.52	0.67	0.66	0.52	0.34	0.10	**0.89**	**0.90**	0.67	**0.67**	**0.80**	**0.77**	0.70	0.85	0.85	0.67	0.67	0.79	0.71	0.60
Distance	DeepSVDD	0.50	0.50	0.39	0.31	**0.19**	0.11	0.02	0.96	0.93	0.84	0.88	0.90	0.88	**0.52**	0.77	0.73	0.70	0.67	0.57	0.43	0.06
GAN	FAnoGAN	0.50	0.50	0.39	0.32	0.21	0.13	0.04	0.77	0.65	**0.76**	0.61	0.34	0.14	0.01	**0.68**	0.60	0.58	0.45	0.29	0.16	**0.04**
GAN	Ganomaly	0.50	0.50	0.47	0.39	0.26	0.14	0.03	0.57	0.50	0.52	0.35	0.17	0.07	0.00	0.66	0.60	**0.65**	**0.54**	**0.37**	**0.22**	0.02

(b) Evaluation result on Dave Dataset.

Table 5: The prediction on OOD images for each OOD detector under the weather-transformed (fog, rain, snow) datasets.

Family	Model	FGSM							PGD							SPSA							SP						
		AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99
Reconstruction	SAE	0.52	0.52	0.46	0.37	0.27	0.12	0.03	0.45	0.46	0.38	0.31	0.16	0.08	0.02	0.69	0.66	0.59	0.58	0.44	0.34	0.07	0.78	0.82	0.69	0.70	0.71	**0.72**	0.56
Reconstruction	VAE	0.52	0.51	0.46	0.41	0.27	0.21	0.05	0.44	0.46	0.40	0.29	0.22	0.17	0.03	0.63	0.61	0.56	0.49	0.45	0.29	0.16	0.73	0.77	0.66	0.67	0.68	0.61	0.48
Density	GMM	0.78	0.70	0.70	0.72	0.75	0.76	0.72	0.83	0.72	0.75	0.78	0.81	0.80	0.66	0.93	0.85	0.76	0.80	0.85	0.89	0.94	0.79	0.78	0.69	0.71	0.73	0.76	0.79
Density	KDE	0.61	**0.62**	0.67	**0.67**	**0.62**	0.53	0.53	**0.61**	**0.60**	**0.67**	**0.66**	**0.61**	**0.54**	0.52	**0.70**	0.70	0.67	0.67	0.67	0.60	0.60	0.70	0.73	0.66	0.67	0.68	0.68	0.54
Distance	DeepKNN	0.77	0.71	0.73	0.75	0.75	0.72	**0.48**	0.81	0.70	0.79	0.79	0.73	0.63	**0.34**	0.93	0.85	0.82	0.86	0.89	0.92	0.95	0.81	0.86	0.71	0.74	0.76	0.78	0.80
Distance	DeepRoad	0.73	0.64	0.70	0.71	0.72	0.71	0.60	0.69	0.61	0.70	0.70	0.69	0.65	0.51	0.93	**0.82**	0.74	0.77	0.81	0.86	**0.91**	**0.78**	0.71	0.69	0.69	0.71	0.73	0.76
Distance	DeepSVDD	**0.63**	0.58	**0.67**	0.66	0.63	**0.58**	0.39	0.54	0.51	0.63	0.60	0.55	0.49	0.24	0.96	0.95	**0.71**	**0.74**	**0.80**	**0.85**	0.93	0.81	0.87	**0.67**	**0.67**	**0.69**	**0.72**	0.76
GAN	FAnoGAN	0.56	0.54	**0.67**	0.66	0.60	0.53	0.34	0.52	0.51	0.65	0.61	0.56	0.48	**0.33**	0.69	0.63	0.69	0.72	0.74	0.69	0.38	0.78	0.85	0.66	0.67	0.69	0.70	**0.70**
GAN	Ganomaly	0.53	0.52	0.66	0.65	0.62	0.56	0.43	0.51	0.50	0.66	0.64	**0.61**	0.53	0.41	0.64	0.59	0.67	0.68	0.69	0.69	0.54	**0.77**	**0.84**	0.66	0.66	0.65	0.68	0.70

(a) Evaluation result on Udacity Dataset.

Family	Model	FGSM							PGD							SPSA							SP						
		AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99	AUC_R	AUC_P	F1@70	F1@80	F1@90	F1@95	F1@99
Reconstruction	SAE	0.51	0.51	0.37	0.27	0.17	0.11	0.04	0.49	0.49	0.35	0.25	0.16	0.10	0.03	0.61	0.56	0.45	0.35	0.21	0.13	0.05	0.85	0.75	0.83	0.73	0.48	0.25	0.04
Reconstruction	VAE	0.51	0.50	0.36	0.27	0.18	0.13	**0.05**	0.48	0.49	0.33	0.24	0.16	0.11	**0.04**	0.58	0.54	0.41	0.32	0.21	0.15	0.06	0.76	0.64	0.63	0.47	0.25	0.14	0.03
Density	GMM	0.71	0.69	0.64	0.58	0.47	0.33	0.12	0.67	0.65	0.59	0.52	0.40	0.28	0.10	0.98	0.98	0.86	0.90	0.93	0.92	0.79	**0.98**	**0.98**	0.86	0.90	**0.94**	**0.93**	0.81
Density	KDE	0.81	0.80	0.67	0.67	0.78	0.66	0.52	0.75	0.74	0.67	0.67	0.73	0.59	0.42	**0.88**	**0.89**	0.67	0.67	**0.80**	**0.73**	**0.69**	0.87	0.88	0.67	0.67	0.80	0.69	0.69
Distance	DeepKNN	0.65	0.62	0.57	**0.48**	**0.32**	**0.18**	0.04	**0.62**	0.60	0.52	0.44	0.28	**0.16**	0.03	0.96	0.96	0.87	0.89	0.91	0.87	0.56	0.99	0.99	0.87	0.91	0.95	0.96	**0.87**
Distance	DeepRoad	0.81	0.80	0.67	0.67	0.77	0.66	0.52	0.76	0.74	0.67	0.67	0.73	0.59	0.42	**0.88**	**0.89**	0.67	0.67	**0.80**	**0.73**	0.70	0.87	0.88	0.67	0.67	0.80	0.70	0.69
Distance	DeepSVDD	0.65	**0.62**	0.56	0.48	0.34	0.22	0.04	0.60	**0.57**	0.51	0.42	0.28	0.17	0.03	1.00	0.99	0.86	0.90	0.94	0.97	0.98	1.00	1.00	0.86	0.90	0.94	0.97	0.99
GAN	FAnoGAN	**0.61**	0.57	0.51	0.41	0.29	0.18	0.06	0.56	0.53	0.44	0.36	**0.24**	0.14	0.05	0.73	0.67	**0.65**	**0.57**	0.42	0.28	0.12	1.00	1.00	0.86	0.89	0.94	0.96	0.99
GAN	Ganomaly	0.58	0.55	**0.55**	0.46	**0.32**	**0.18**	0.04	0.55	0.53	**0.52**	**0.43**	0.29	**0.16**	0.03	0.63	0.60	0.61	0.52	0.39	0.26	0.06	1.00	1.00	**0.82**	**0.86**	0.92	0.96	0.99

(b) Evaluation result on Dave Dataset.

Table 6: The prediction on OOD images for each OOD detector under the weather-transformed (FGSM, PGD, SPSA, and SP) datasets. Cell color represents the statistical significance among data for each evaluation metric, computed using ScottKnottESD tests. We colored the cells in the top three groups and bolded the value in the fourth group. Darker cells show superior performance.

This may imply that reconstruction-based detectors face challenges in distinguishing reconstruction loss of InD and OOD samples.

Overall, the dataset type does not result in a remarkable effect on detectors. Real-world samples may obscure the border between InD and OOD since the F1@99 is more frequently dropped to around zero in the Dave dataset. On the other hand, adversarial attacks can make OOD samples harder to identify than weather transformation. Black-box attacks seem to produce fewer effects on OOD detectors than white-box attacks. Under the weather transformation domain, fog in the Dave dataset is the most challenging case due to the fact that fog samples might be very similar to InD samples.

Family	AUC-ROC	AUC-PR	F1@0.70	F1@0.80	F1@0.90	F1@0.95	F1@0.99
Reconstruction	0.612 ± 0.341	0.656 ± 0.289	0.489 ± 0.344	0.472 ± 0.365	0.454 ± 0.389	0.441 ± 0.407	0.416 ± 0.419
Density	0.782 ± 0.166	0.772 ± 0.152	0.703 ± 0.114	0.700 ± 0.143	0.693 ± 0.179	0.683 ± 0.206	0.674 ± 0.234
Distance	**0.840 ± 0.174**	**0.794 ± 0.169**	**0.717 ± 0.166**	**0.711 ± 0.201**	**0.700 ± 0.247**	**0.691 ± 0.280**	0.667 ± 0.311
GAN	0.759 ± 0.178	0.716 ± 0.170	0.634 ± 0.121	0.605 ± 0.153	0.569 ± 0.201	0.540 ± 0.252	0.520 ± 0.296

Table 7: Family-wise mean and standard deviation of metrics across weather OOD images, averaging Udacity and Dave datasets.

Finding 2: We observe that adversarial attacks can produce more serious negative effects on detectors than weather transformation. Most detectors retain their standard performance and maintain their statistical group ranking, follow the family pattern across OOD conditions and datasets, while samples collected from complex scenes (Dave dataset) can seriously harm the reconstruction-based family.

5.3 RQ3. Time Efficiency

According to the Table 8, total inference times of detectors vary from 11.5s to 91.7s, due to the steps of models in the OOD detection process. We convert this to FPS, it can vary from 1,814 fps to 228 fps, which represents OOD detection process is not a time-consuming process. Hence, every model comfortably clears the 30 fps real-time threshold on the test workstation, but their computational footprints differ sharply:

Fastest tier (<16s): VAE (11.5s), GMM (13.8s), and Ganomaly (15.1s) dominate on raw speed, making them attractive for latency-critical modules such as perception pre-filters or embedded edge deployments.

Balanced tier (15–30s): DeepKNN (18.3s) and FAnoGAN (28.4s) offer moderate runtimes while, in earlier accuracy experiments, delivering competitive AUROC under both weather shifts and adversarial noise, suggesting a favourable accuracy-latency trade-off.

Slow tier (>30s): KDE (52.9s) and DeepSVDD (32.2s) remain feasible on high-end GPUs but risk violating real-time budgets once model ensembles or additional perception modules are co-scheduled.

Outlier: DeepRoad (91.7s) is an order of magnitude slower than the fastest contenders, primarily due to its sequential patch-level processing; optimization (e.g., patch batching or TensorRT fusion) would be required for field deployment.

Model	SAE	VAE	KDE	GMM	DeepKNN	DeepRoad	DeepSVDD	Ganomaly	F–AnoGAN
Time (s)	29.1	11.5	52.9	13.8	18.3	91.7	32.2	15.1	28.4
FPS	716	1814	394	1510	1140	228	648	1380	735

Table 8: Inference latency and throughput on the 20,842 image set (A100 GPU).

Finding 3: We find that OOD detection is not a time-consuming task, where most detectors can exceed 30 FPS on real-time inference. There is no obvious correlation between the precision of an OOD detector and its inference time.

6 Threats to Validity

Despite our efforts to build a rigorous and reproducible benchmark, several threats to internal and external validity exist in our study.

6.1 Internal Validity

(1) One of the threats can be attributed to the implementation details of the OOD detector, which was re-implemented or adopted from public code. For example, KDE was initially designed to identify OOD data in a multiclass dataset like MNIST or CIFAR-10 for a classification task. We implemented this detector by changing the model to make one-class classification and modifying the dataset preprocessing step to make it satisfy our dataset and the task. The designed logic of each detector shows a remarkable disparity; some detectors could directly handle our task, while others may need to adapt to the task, which may cause unfair comparison. To reduce these effects, we built a standard pipeline shared for every detector on the training and inference processes.

(2) Hyperparameter selection poses a further risk. We selected hyperparameters commonly from the previous paper setting without parameter search. Since our OOD detection task and dataset differ from the original design of detectors, the selected hyperparameter values may not achieve the detector's best performance. Proposing a hyperparameter search may cause the result to be slightly different from the current one. To minimize the effects of hyperparameters, we set the shared hyperparameters (e.g., epoch, learning rate) the same among each class and release all the hyperparameter values in Section 4.4 for further studies.

6.2 External Validity

(1) The number of detectors in each class may not be enough to represent the features of this class. Due to the time and budget, we only implemented the typical detectors of each class. But we tried to make our result representative by choosing detectors within each class that have structural diversity to cover detectors within this class as much as possible.

(2) Our selected techniques for generating the OOD dataset are limited. We only considered weather-transformed images and adversarial-attacked images to rank the selected detectors and make a general recommendation. However, factors like luminance, image blurring, and vehicle collision could cause unpredictable effects on detector ranking. Thus, we tried to include the most usual OOD scenario in our study and expect that each detector could show consistent performance in other OOD domains.

7 Conclusion

This paper presented a comprehensive empirical study on the performance and robustness of OOD detectors across multiple methodological families using two driving datasets on both weather-transformed (Fog, Rain, Snow) and adversarial-attacked conditions. Our unified experimental protocol and metric framework enable a consistent comparison among and across families, providing new insights into how OOD detection mechanisms behave under various driving conditions.

The results reveal several clear trends. First, distance-based detectors consistently achieve the best overall performance, showing a strong robustness across both datasets and OOD conditions.

Their stability across thresholds suggests that feature space distance measures offer a reliable and generalizable criterion for identifying OOD samples, while density-based detectors follow closely. In contrast, reconstruction-based and GAN-based detectors exhibit higher variance and sharp performance degradation under complex or adversarial perturbations, indicating weaker generalization and calibration properties.

From a broader perspective, our findings highlight that robust OOD detection in autonomous driving depends not only on discriminative accuracy but also on stability across diverse OOD conditions. Methods grounded in embedding distance or probabilistic density estimation demonstrate a consistent advantage because they capture intrinsic structural properties of the data distribution, rather than relying solely on reconstruction fidelity or generative reconstruction error.

Future work will extend this analysis toward temporal and multimodal OOD detection, incorporating sequential consistency checks and cross-sensor fusion (e.g., camera–LiDAR–radar). Additionally, we plan to explore calibration-aware ensembling across OOD families and adaptive thresholding strategies to further enhance real-time deployability in safety-critical driving environments.

References

[1] 2022. ISO/IEC/IEEE 29119-1: Software and Systems Engineering—Software Testing—Part 1: Concepts and Definitions. International Standard.

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby Breckon. 2018. Ganomaly: Semi-Supervised Anomaly Detection via Adversarial Training. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 622–637.

[3] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. 2019. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. In *Robotics: Science and Systems*.

[4] National Transportation Safety Board. 2017. Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor–Semitrailer Truck, Williston, Florida, May 7, 2016. Accident Report NTSB/HAR-17/02. https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR1702.pdf Online; accessed 15 Jul 2025.

[5] National Transportation Safety Board. 2020. Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018. Accident Report NTSB/HAR-20/03. https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR2003.pdf Online; accessed 15 Jul 2025.

[6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316* (2016). https://arxiv.org/abs/1604.07316

[7] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (2020), 125. doi:10.3390/info11020125

[8] Holger Caesar, Varun Bankiti, and Alex H. Lang et al. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11618–11628.

[9] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. 2021. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation. arXiv:2104.14812 [cs.CV] https://arxiv.org/abs/2104.14812

[10] Ming-Fang Chang, John W. Lambert, and Patsorn Sophon et al. 2019. Argoverse: 3D Tracking and Forecasting with Rich Maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8748–8757.

[11] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. 2020. Learning by Cheating. In *Proc. Conf. Robot Learning (CoRL)*, Vol. 100. 66–79.

[12] CNBC. 2023. California Suspends Cruise Robotaxi Permit After Pedestrian Dragging Incident. https://www.cnbc.com/2023/10/24/california-suspends-cruise-robotaxi-after-pedestrian-dragging.html. Online; accessed 15 Jul 2025.

[13] Felipe Codevilla, Eder Santana, Antonio M. Lopez, and Adrien Gaidon. 2019. Exploring the Limitations of Behavior Cloning for Autonomous Driving. In *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*. 9329–9338.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[15] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142. doi:10.1109/MSP.2012.2211477

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Conference on Robot Learning (CoRL)*. PMLR, 1–16.

[17] Electrek. 2016. A Google Self-Driving Car Caused a Crash for the First Time. https://electrek.co/2016/07/01/understanding-fatal-tesla-accidentautopilot-nhtsa-probe/. Online; accessed 18 Aug. 2019.

[18] Electrek. 2016. Tesla Model S Driver Crashes into a Van While on Autopilot. https://electrek.co/2016/05/26/tesla-model-s-crash-autopilot-video/. Online; accessed 18 Aug. 2019.

[19] Ertunc Erdil, Krishna Chaitanya, Neerav Karani, and Ender Konukoglu. 2021. Task-agnostic Out-of-Distribution Detection Using Kernel Density Estimation. arXiv:2006.10712 [cs.CV] https://arxiv.org/abs/2006.10712

[20] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML] https://arxiv.org/abs/1412.6572

[21] Manzoor Hussain, Nazakat Ali, and Jang-Eui Hong. 2022. DeepGuard: A Framework for Safeguarding Autonomous Driving Systems from Inconsistent Behavior. arXiv:2111.09533 [cs.LG] https://arxiv.org/abs/2111.09533

[22] Hyeong S. Hwang and Richard A. Haddad. 1995. Adaptive Median Filters: New Algorithms and Results. *IEEE Transactions on Image Processing* 4, 4 (1995), 499–502. doi:10.1109/83.370679

[23] Rajat Koner, Poulami Sinhamahapatra, Karsten Roscher, Stephan Günnemann, and Volker Tresp. 2021. OODformer: Out-Of-Distribution Detection Transformer. arXiv:2107.08976 [cs.CV] https://arxiv.org/abs/2107.08976

[24] Alex Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report. University of Toronto. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[25] Firas Laakom, Fahad Sohrab, Jenni Raitoharju, Alexandros Iosifidis, and Moncef Gabbouj. 2023. Convolutional autoencoder-based multimodal one-class classification. arXiv:2309.14090 [cs.LG] https://arxiv.org/abs/2309.14090

[26] Todd Litman. 2021. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning. *Victoria Transport Policy Institute Working Paper* (2021).

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*. arXiv:1706.06083 [stat.ML]

[28] Xiaofeng Mao, Yuefeng Chen, Yao Zhu, Da Chen, Hang Su, Rong Zhang, and Hui Xue. 2023. COCO-O: A Benchmark for Object Detectors under Natural Distribution Shifts. arXiv:2307.12730 [cs.CV] https://arxiv.org/abs/2307.12730

[29] Bo Peng, Yadan Luo, Yonggang Zhang, Yixuan Li, and Zhen Fang. 2025. ConjNorm: Tractable Density Estimation for Out-of-Distribution Detection. arXiv:2402.17888 [cs.LG] https://arxiv.org/abs/2402.17888

[30] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. 2016. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1099–1106. doi:10.1109/IROS.2016.7759164

[31] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Lukas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Vol. 80. PMLR, 4393–4402.

[32] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. 2017. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. arXiv:1703.05921 [cs.CV] https://arxiv.org/abs/1703.05921

[33] Luca Scrucca. 2023. Entropy-Based Anomaly Detection for Gaussian Mixture Modeling. *Algorithms* 16, 4 (2023). doi:10.3390/a16040195

[34] Rashmi Siddalingappa and Sekar Kanagaraj. 2021. Anomaly detection on medical images using autoencoder and convolutional neural network. *International Journal of Advanced Computer Science and Applications* 7 (2021).

[35] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs.CV] https://arxiv.org/abs/1409.1556

[36] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. 2020. Misbehaviour Prediction for Autonomous Driving Systems. In *Proceedings of 42nd International Conference on Software Engineering (ICSE '20)*. ACM, 12 pages.

[37] SullyChen. 2018. driving-datasets: A collection of labeled car driving datasets. https://github.com/SullyChen/driving-datasets. [Data set; accessed: 2025-10-18].

[38] Pei Sun, Henrik Kretzschmar, and Xerxes Dotiwalla et al. 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2446–2454.

[39] Chakkrit Tantithamthavorn, Shane McIntosh, Ahmed E. Hassan, and Kenichi Matsumoto. 2017. An Empirical Comparison of Model Validation Techniques for Defect Prediction Models. 1 (2017).

[40] David M. J. Tax and Robert P. W. Duin. 2004. Support Vector Data Description. *Machine Learning* 54, 1 (2004), 45–66. doi:10.1023/B:MACH.0000008084.60811.49

[41] Team Epoch. 2016. Steering angle model: Epoch. https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/cg23. Online; accessed 18 August 2019.

[42] Udacity. 2016. Udacity Self-Driving Car Dataset. https://github.com/udacity/self-driving-car-sim. Accessed: 2025-06-11.

[43] Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron van den Oord. 2018. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. In *Proceedings of the 35th International Conference on Machine Learning (PMLR, Vol. 80)*. 5025–5034.

[44] U.S. National Highway Traffic Safety Administration. 2020. Automated Vehicles for Safety. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety. Accessed: 8 Jul 2025.

[45] Xuanjun Wang, Jiacheng Ye, Yong Jiang, Wei Bi, and Zhifang Sui. 2023. A Survey on Out-of-Distribution Evaluation of Neural NLP Models. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*. 6789–6797. doi:10.24963/ijcai.2023/749

[46] Julia Wolleb, Robin Sandkühler, and Philippe C. Cattin. 2020. DeScarGAN: Disease-Specific Anomaly Detection with Weak Supervision. arXiv:2007.14118 [eess.IV] https://arxiv.org/abs/2007.14118

[47] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. 2022. OpenOOD: Benchmarking Generalized Out-of-Distribution Detection. arXiv:2210.07242 [cs.CV] https://arxiv.org/abs/2210.07242

[48] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2024. Generalized Out-of-Distribution Detection: A Survey. arXiv:2110.11334 [cs.CV] https://arxiv.org/abs/2110.11334

[49] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. 2019. Efficient GAN-Based Anomaly Detection. arXiv:1802.06222 [cs.LG] https://arxiv.org/abs/1802.06222

[50] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. 2018. DeepRoad: GAN-based Metamorphic Autonomous Driving System Testing. arXiv:1802.02295 [cs.SE] https://arxiv.org/abs/1802.02295

[51] Yinghua Zhang, Yangqiu Song, Jian Liang, Kun Bai, and Qiang Yang. 2020. Two Sides of the Same Coin: White-box and Black-box Attacks for Transfer Learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining (KDD '20)*. ACM, 2989–2997. doi:10.1145/3394486.3403349

[52] Wei Zhao, Qi Liu, Fengxiang He, and Ming Liu. 2024. Adversarial Perception Attacks and Defenses for Autonomous Driving Systems: A Comprehensive Survey. *IEEE Transactions on Intelligent Transportation Systems* (2024). doi:10.1109/TITS.2024.3354027