

Unsupervised Anomaly Detection Improves Imitation Learning for Autonomous Racing

Yuang Geng¹, Yang Zhou¹, Yuyang Zhang¹, Zhongzheng Ren Zhang¹, Kang Yang¹,
Tyler Ruble¹, Giancarlo Vidal¹, and Ivan Ruchkin¹

Abstract—Imitation Learning (IL) has shown significant promise in autonomous driving, but its performance heavily depends on the quality of training data. Noisy or corrupted sensor inputs can degrade learned policies, leading to unsafe behavior. This paper presents an unsupervised anomaly detection approach to automatically filter out abnormal images from driving datasets, thereby enhancing IL performance. Our method leverages a Convolutional Autoencoder with a novel *latent reference loss*, which forces abnormal images to reconstruct with higher errors than normal images. This enables effective anomaly detection without requiring manually labeled data. We validate our approach on the realistic DonkeyCar autonomous racing platform, demonstrating that filtering videos significantly improves IL policies, as measured by a 25-40% reduction in cross-track error. Compared to baseline and ablation models, our method achieves superior anomaly detection across three real-world video corruptions: collision-based occlusions, transparent obstructions, and raindrop interference. The results highlight the effectiveness of unsupervised video anomaly detection in improving the robustness and performance of IL-based autonomous control.

Video: https://youtu.be/RjJ3nZR6_RQ

I. INTRODUCTION

Imitation Learning (IL) has gained significant popularity in autonomous driving due to its ability to leverage large-scale datasets of expert demonstrations, enabling efficient policy learning without explicit rewards [1]. IL approaches, such as behavior cloning [2], [3] and direct policy learning [4], have shown promise in developing self-driving policy.

IL relies on a high-quality dataset with clean videos and perfect human actions for effective control policy learning [1], [5]. However, noisy or unreliable data from sensors can lead to unsafe behavior in the agent, even if the IL model is well-trained [6]. Such corruptions can arise from various factors, such as external obstructions (e.g., debris covering the camera lens), incorrect human behavior (e.g., the camera hitting the barrier while a human is driving), and sensor malfunctions (e.g., a faulty camera capturing distorted frames). If a corrupted image is used for training,

the learned controller may fit incorrect patterns, leading to poor generalization and unsafe behavior. Therefore, ensuring a reliable and clean dataset is crucial for robust imitation learning.

A major challenge in preparing IL datasets is the *absence of labels* that indicate the quality of a video frame, especially in autonomous driving [7]. In practice, datasets often contain a mix of normal and abnormal images without predefined quality labels (e.g., a two-hour naturalistic driving video). However, most existing methods for abnormal video detection rely on semi-supervised learning approaches [8], [9], which require manually curated datasets containing only historical normal images. This process demands significant human effort to remove abnormal samples [10], limiting the practicality of imitation learning (IL), particularly for large-scale autonomous driving applications. Additionally, if the training data differs in environmental conditions from the test data, false alarms may arise. Therefore, a fully unsupervised approach for detecting quality anomalies in raw, unstructured datasets is essential for real-world applications.

This paper presents an unsupervised anomaly detection method for identifying abnormal images to enhance imitation learning performance. Unlike existing fully supervised, semi-supervised, and weakly supervised anomaly detection, our method directly detects anomalous images within the training data without requiring labeled samples. The framework is built on a *convolutional autoencoder (CAE)* with a novel loss, *reference latent loss*, which encourages the CAE to reconstruct normal images more accurately than abnormal ones during training. Leveraging this reconstruction difference, our unsupervised anomaly detection method autonomously identifies anomalous images in the training data.

Our unsupervised detection performance on abnormal images is validated through extensive experiments on the physical self-driving DonkeyCar platform [11]. This system serves as a widely used, low-cost autonomous racing testbed that replicates real-world driving challenges in a controlled environment. Experiments demonstrate that our proposed method outperforms the state-of-the-art (SOTA) unsupervised video anomaly detection techniques [12] in identifying three types of anomalies: raindrops, collisions, and plastic obstructions on the camera. The main contributions are:

- An unsupervised anomaly detection method with latent reference loss to clean unlabeled driving videos.
- A video-cleaning pipeline to improve imitation learning.
- Validation of our pipeline on improving imitation learning for the robotic platform DonkeyCar.

Provisional patent filed, contact techlicensing@research.ufl.edu for details. This work was supported in part by the NSF Grants CCF 2403616 and CNS 2513076. Any opinions, findings, or conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF) or the U.S. Government. We thank Vivaan Goomer, Mohit Kukreja, and Siyuan Pan for their exploration of autoencoder cleaning. We also thank Akshat Kothiyal for enhancing the reproducibility of this work.

¹Trustworthy Engineered Autonomy (TEA) Lab, Department of Electrical and Computer Engineering, University of Florida, yuang.geng@ufl.edu, zhouyang1@ufl.edu, yuyangzhang@ufl.edu, renzhongzh.zhang@ufl.edu, yang.kang@ufl.edu, tyler.ruble@ufl.edu, g.vidal@ufl.edu, iruchkin@ece.ufl.edu

The remainder of this paper is organized as follows. Section II reviews related work on anomaly detection and its relevance to imitation learning. Section III formulates our problem of unsupervised anomaly detection. Section IV introduces details of the reconstruction-based cleaning pipeline. Section V presents the experimental setup, including dataset descriptions and implementation details. Section VI analyzes the results, comparing the proposed method with baselines and evaluating its impact on imitation learning. Finally, Section VII concludes the paper and discusses future work.

II. RELATED WORK

Imitation learning. Humans often learn by imitation, observing others perform tasks and inferring the appropriate actions [1]. IL applies this principle to train agents by mimicking expert demonstrations. IL becomes increasingly important in complex scenarios, as solving tasks with a higher number of possible actions (e.g., continuous action spaces) or intricate dynamics requires significantly more training samples [13]. We study IL in the context of a vision-based autonomous racing task. Specifically, we apply behavior cloning to learn direct mappings from images to steering and throttle actions.

Data Quality in Imitation Learning. Data quality is crucial for imitation learning: poor-quality demonstrations can significantly degrade model performance. A recent paper [5] highlights the importance of curating high-quality datasets for imitation learning and formalizes the concept of quality with two properties: action divergence and transition diversity. Another work [14] reweighs bad demonstrations during training a controller with a confidence predictor. This reweighing policy is useful when data is limited and redundant in the self-driving domain, when large-scale driving data is available. Therefore, we propose distinguishing high-quality from low-quality data to enhance the controller's performance, which still remains a challenge.

Weakly Supervised Anomaly Detection. Anomaly detection has the potential to improve the quality of IL datasets by identifying and removing corrupted demonstrations. Unfortunately, nearly all methods for anomaly detection in videos [15]–[17] address the *weakly supervised* setting. That is, they are given access to a clean/nominal video dataset and then exposed to a video with anomalies. For instance, the MVTecAD [18] benchmark ensures the training data is completely clean, whereas the test data may contain anomalous contents such as differences in size, color, and structure. Therefore, its evaluation protocol is one-class classification (OCC, also known as zero-positive setting [15]), and the methods that inherit this assumption are also essentially one-class classifiers [19]. This setting has limited applicability to large-scale autonomous datasets because manual cleaning of anomalies is labor-intensive. Attempts to relax the requirement for clean data, like pseudo-labeling mixed videos with a pretrained model [20], still rely on external supervision.

Unsupervised Anomaly Detection. Anomaly detection with unlabeled training data (i.e., combining normal and anomalous samples) is under-explored in the literature. Most

anomaly detection methods in the literature fall in the category of OCC [7], [21]. To our knowledge, only one method has been proposed to address our setting — the *Generative Cooperative Learning (GCL)* [12]. It uses a generator-discriminator framework to iteratively refine anomaly detection in unlabeled videos. The generator reconstructs normal representations and distorts anomalous ones using negative learning, while the discriminator estimates anomaly probabilities, with both models improving through alternating pseudo-labeling. However, as our results show in Section VI, GCL struggles to detect anomalies when the differences between normal and corrupted videos are subtle. In contrast, **our approach detects low-quality driving data more robustly (e.g., raindrops on the camera)**, even in cases where anomalies are difficult to distinguish from normal data visually.

III. PROBLEM: UNSUPERVISED ANOMALY DETECTION

Given an unlabeled video training dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where the \mathbf{x} is an image frame and \mathbf{y} is the corresponding expert action from set \mathcal{A} , imitation learning — and in particular behavior cloning — is used to train a neural controller $h : \mathcal{X} \rightarrow \mathcal{A}$. To enhance the performance of IL, we aim to improve the quality of \mathcal{D} by detecting and removing corrupted data.

We assume that the raw dataset \mathcal{D} consists of two parts: high-quality clean data $\mathcal{D}_{\text{clean}}$ and low-quality dirty data $\mathcal{D}_{\text{dirty}}$. The partition between the two is not known ahead of time. We assume that dirty data constitutes significantly less than 50%; otherwise, the definitions of clean and dirty would need to be reversed.

$$\mathcal{D} = \mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{dirty}} \quad (1)$$

We aim to automatically identify as many dirty images as possible with a learning-based function f without any supervision on dataset \mathcal{D} :

$$f(\mathbf{x}_i, \mathbf{y}_i) = \Pr[(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{\text{clean}}] \quad (2)$$

Having learned such function f , we can use a confidence threshold τ to construct the *filtered dataset* as:

$$\hat{\mathcal{D}}_{\text{clean}} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D} \mid f(\mathbf{x}_i, \mathbf{y}_i) \geq \tau\}, \quad (3)$$

where the estimated cleaned dataset $\hat{\mathcal{D}}_{\text{clean}}$ closely approximates the exact clean dataset $\mathcal{D}_{\text{clean}}$, by maximizing shared clean images and minimizing corrupted ones.

The impact of dataset filtering is determined based on three trained controllers:

- 1) h is trained on the full dataset \mathcal{D} .
- 2) h_{clean} is trained on the true clean dataset $\mathcal{D}_{\text{clean}}$.
- 3) \hat{h}_{clean} is trained on the filtered dataset $\hat{\mathcal{D}}_{\text{clean}}$.

After the controllers are trained, we use cross-track error to evaluate each performance. The final objective is to minimize the performance gap between \hat{h}_{clean} and h_{clean} , ensuring the \hat{h}_{clean} covers the expert behavior in imitation learning.

Difference from Other Settings. Our problem formulation calls for an unsupervised anomaly detection approach trained directly on the test data, which includes *both* normal and

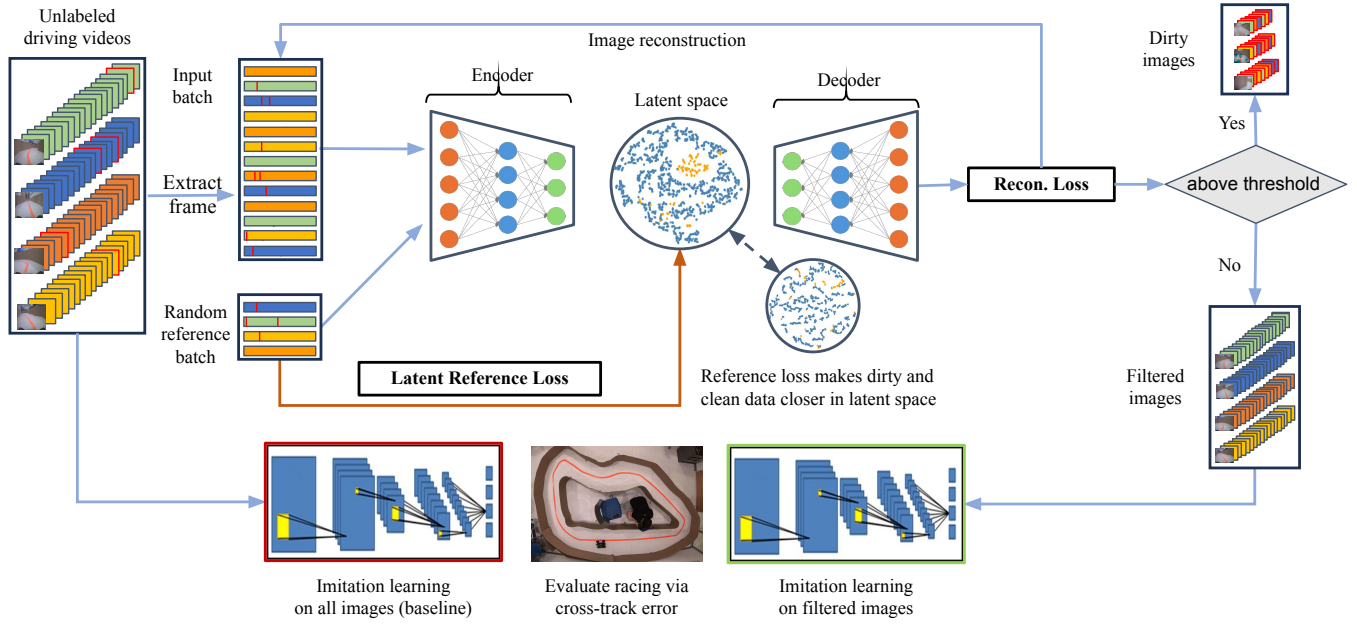


Fig. 1: Pipeline for improving imitation learning trained on dirty images via unsupervised video anomaly detection.

abnormal images. In other words, the test data also serves as the training data, eliminating the need for additional (cleaned) training datasets. In contrast, typical anomaly detection methods are *semi-supervised* from our perspective: they require historical normal images for training. If we use them with the training data that contains abnormal images or differs in environmental conditions from the test data, false alarms are likely at test time.

Furthermore, our setting does not set aside any validation data to prevent overfitting. Instead, it is specifically designed to require the detector to overfit normal (i.e., more frequent) signals while avoiding overfitting abnormal signals and hence responding to them differently.

IV. RECONSTRUCTION-BASED CLEANING PIPELINE

Our data-cleaning pipeline for anomaly image detection is illustrated in Figure 1. Anomalous images are identified based on the reconstruction differences between normal and abnormal images during training. To preserve these differences, a latent reference loss is introduced. The method is detailed in the following subsections.

A. Reconstruction-Based Unsupervised Anomaly Detection

Our anomaly detection method leverages a neural network designed to reconstruct normal images more accurately than anomalous ones. To achieve this, we utilize a convolutional autoencoder (CAE) for image reconstruction. The CAE consists of an encoder, $\mathbf{e}_\psi(\mathbf{x}_i) = \mathbf{h}_i$, and a decoder $\mathbf{d}_\phi(\mathbf{h}_i) = \mathbf{x}'_i$. The hyperparameters of the CAE are detailed in Table I.

Loss function design. Our loss function \mathcal{L} for the CAE consists of two terms: reconstruction loss \mathcal{L}_{rec} and the novel *latent reference loss* $\mathcal{L}_{\text{refer}}$ weighted with a hyperparameter λ :

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{refer}} \quad (4)$$

The reconstruction loss \mathcal{L}_{rec} aims to minimize the pixel-level mean squared error (MSE) between an original image \mathbf{x}_i and its reconstructed counterpart $\mathbf{x}'_i = \mathbf{d}_\phi(\mathbf{e}_\psi(\mathbf{x}_i))$, denoted $\epsilon(\mathbf{x}_i, \mathbf{x}'_i) = \|\mathbf{x}_i - \mathbf{x}'_i\|_2$. This loss is calculated across all N input signals in the raw, unlabeled dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, as follows:

$$\mathcal{L}_{\text{rec}}(\psi, \phi) = \frac{1}{N} \sum_{i=1}^N \epsilon(\mathbf{x}_i, \mathbf{d}_\phi(\mathbf{e}_\psi(\mathbf{x}_i))) \quad (5)$$

Since abnormal images are present in the data, our CAE will also learn to reconstruct them during training. To mitigate this, we introduce a *latent reference loss* $\mathcal{L}_{\text{refer}}$ designed to hinder the CAE from effectively reconstructing abnormal images. Before training, we randomly sample $M < N$ datapoints to create a *reference dataset* $\mathcal{D}_{\text{refer}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^M$. In each training epoch, this reference dataset $\mathcal{D}_{\text{refer}}$ is encoded into the latent space by the encode $\mathbf{e}_\psi(\mathbf{x}_i)$, forming the reference latent set $\mathcal{H}_{\text{refer}} = \{\mathbf{h}_1^r, \dots, \mathbf{h}_M^r\}$.

During training, each input image \mathbf{x}_i is encoded into its latent representation $\mathbf{h}_i = \mathbf{e}_\psi(\mathbf{x}_i)$. The nearest neighbor $\mathbf{h}_i^{\text{near}}$ for the i -th image \mathbf{x}_i in the latent space is then selected from $\mathcal{H}_{\text{refer}}$ based on Euclidean distance,

$$\mathbf{h}_i^{\text{near}} = \arg \min_{\mathbf{h}_j^r \in \mathcal{H}_{\text{refer}}} \|\mathbf{h}_i - \mathbf{h}_j^r\|_2 \quad (6)$$

Then, the reference latent loss $\mathcal{L}_{\text{refer}}$ is defined as the MSE between the latent representation \mathbf{h}_i of the i -th image \mathbf{x}_i and its nearest latent representation $\mathbf{h}_i^{\text{near}}$ selected from the reference latent set $\mathcal{H}_{\text{refer}}$:

$$\mathcal{L}_{\text{refer}}(\psi, \phi) = \frac{1}{N} \sum_{i=1}^N \epsilon(\mathbf{h}_i, \mathbf{h}_i^{\text{near}}) \quad (7)$$

Loss function intuition. Our loss function \mathcal{L} comprises two components: the reconstruction loss \mathcal{L}_{rec} and the proposed

latent reference loss $\mathcal{L}_{\text{refer}}$. Their goals are at odds: the reconstruction loss ensures that the CAE-generated image \mathbf{x}' closely matches the original input \mathbf{x} , while the latent reference loss encourages the reconstructed image \mathbf{x}' to resemble its nearest reference image \mathbf{x}_{near} from the randomly selected dataset $\mathcal{D}_{\text{refer}}$. Since normal images are more prevalent than abnormal ones in the data, and their latent-space representations are closely clustered due to their similarity, both normal and abnormal images are more likely to find normal images as their nearest references in $\mathcal{D}_{\text{refer}}$. While minimizing the latent reference loss, we effectively “pull” abnormal images closer to normal ones in the latent space, thereby degrading their reconstruction quality compared to normal images. As a result, normal images are reconstructed more accurately than anomalies.

B. Detection of Abnormal Images

We quantify the reconstruction ability of the autoencoder for normal and abnormal images using *Pearson correlation coefficient (PCC)* [22], [23]. In practice, both MSE and PCC are widely used to measure reconstruction quality. Unlike reconstruction MSE measuring absolute pixel-wise differences, PCC focuses on their structural similarity between original and reconstructed images, making it more suitable for anomaly detection. For example, a raindrop anomaly would spike the reconstruction MSE less than PCC. Additionally, PCC’s well-defined range (-1 to 1) facilitates threshold selection, whereas MSE lacks clear bounds. PCC r_i of the i_{th} image is defined as:

$$r_i = \frac{(\mathbf{x}'_i - \bar{\mathbf{x}}'_i)^T (\mathbf{x}_i - \bar{\mathbf{x}}_i)}{\|\mathbf{x}'_i - \bar{\mathbf{x}}'_i\|_2 \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|_2}, \quad (8)$$

where \mathbf{x}_i and $\bar{\mathbf{x}}_i$ are, respectively, the i -th image and the mean of the i -th image, while \mathbf{x}'_i and $\bar{\mathbf{x}}'_i$ represent the reconstructed image and its mean.

We anticipate that normal images will be reconstructed more accurately than abnormal images, resulting in higher PCCs for normal images. However, there is an additional challenge: autonomous racing videos have significant variation among normal images due to different driving circumstances. For instance, normal images captured from straight-line driving are reconstructed more accurately than those from turning scenarios. Consequently, even normal images exhibit varying reconstruction quality with the CAE.

To mitigate reconstruction variations in normal images and reduce false alarms, we apply a *median filter* with a window size of 100 (5 seconds during driving) to smooth all PCCs before anomaly detection. If the median-filtered PCC of an image falls below the threshold δ , we classify it as abnormal. The threshold δ is determined as the median of all reconstruction coefficients minus 0.05,

$$\delta = \text{median}(r_{t-50}, r_{t-49}, \dots, r_t, \dots, r_{t+50}) - 0.05 \quad (9)$$

C. Imitation Learning for Control

Imitation learning aims to train a controller $h_{\kappa} : \mathcal{X} \rightarrow \mathcal{A}$ by mimicking expert demonstrations from a dataset $\mathcal{D} =$

$\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$. We use a common IL approach, *behavior cloning (BC)* [13], which learns a supervised (sensitive to dataset quality) mapping from states to actions. The controller training loss is to minimize the difference between the model’s action and expert action:

$$\mathcal{L}_{BC}(\kappa) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i - h_{\kappa}(\mathbf{x}_i)\|_2^2 \quad (10)$$

V. EXPERIMENTAL SETUP

This section first provides details of dataset, then introduces the structure of our method, and eventually, we explain how to calculate cross-track error to measure the controller.

Datasets. We collected all of our datasets on the physical autonomous racing platform DonkeyCar [11]. The human driver’s objective is to keep the car as close to the centerline as possible. We collected a normal (clean) dataset and three types of realistically abnormal (corrupted/dirty) datasets:

- 1) *Collision-based corruption*: The car hits the track wall, which partially obscures the camera view.
- 2) *Transparent obstruction*: The camera is wrapped in a transparent plastic, distorting the image.
- 3) *Environmental interference*: The camera is affected by raindrops, reducing visibility.

These corrupted datasets closely resemble real-world driving data challenges, where environmental factors and physical obstructions degrade perception. Additionally, the differences in *contrast and brightness* between normal and abnormal images are *subtle*, making it particularly difficult for models to distinguish between them. This provides a *challenging yet practical benchmark* for anomaly detection in autonomous driving systems.

For each, we collected 10 minutes of driving video. For each corrupted dataset, the ratio of normal to abnormal frames is **10:1**, consisting of 16,000 normal frames and 1,600 abnormal frames. The raw dataset and filtered dataset are demonstrated below in Fig 2.

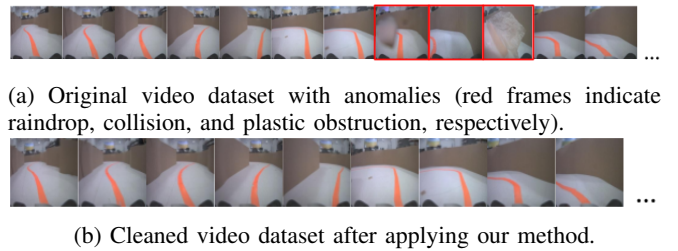


Fig. 2: Comparison of original and cleaned video.

IL and CAE implementation. Our *imitation learning (IL) model* is a CNN-based controller with three convolutional layers for hierarchical feature extraction. The first layer applies 24 filters, followed by batch normalization and max pooling. The second and third-layer extract features are flattened and passed through fully connected layers. The final tanh activation outputs continuous steering values $[-1, 1]$ for control. For **CAE**, the encoder consists of three convolutional

layers (64 filters), followed by a fully connected layer. The decoder is symmetric, with a final sigmoid to normalize pixel values. More details are provided in Table I.

The CNN controllers and the CAE are trained using the Adam optimizer. The CNN uses a learning rate of 5×10^{-5} , batch size of 512, and is trained for 100 epochs. The CAE uses a learning rate of 0.0005, batch size of 256, $\lambda = 1$, and is trained for 100 epochs on raindrop and plastic data, and for 10 epochs on wall-hit data. All the experiments are conducted on an NVIDIA A100 and 2080Ti GPU. For deployment, we use a Jetson Nano to autonomously run the DonkeyCar.

TABLE I: Architectures of the CAE and CNN Controller.

Component	Layer	Channels/Neurons	Kern.	Act. Fn.
Enc. (CAE)	Conv1	16	3×3	ReLU
	Conv2	32	3×3	ReLU
	Conv3	64	3×3	ReLU
	FC	4096→256	-	ReLU
Dec. (CAE)	FC	256→4096	-	ReLU
	Deconv1	64	3×3	ReLU
	Deconv2	32	3×3	ReLU
	Deconv3	16	3×3	Sigmoid
CNN Ctrl.	Conv1	24	5×5	ReLU
	BN+Pool	-	-	-
	Conv2	32	3×3	ReLU
	Conv3	64	3×3	ReLU
	FC	128→64	-	ReLU
	Dropout Output	- 1	- -	- Tanh

Racing evaluation. For each cleaning approach, the ultimate success metric is the *cross-track error* (CTE), which measures the distance between the car and the track’s centerline. To gather data, a ceiling-mounted camera is positioned to capture a top-down view of the entire racing track and the car’s movement. A clear piece of green tape fixed on the top of the car is used for detection. The car, controlled by the trained IL controller, runs on the track for multiple laps while being recorded by the ceiling camera.

We extract the contour of the track centerline (orange tape) and determine the N midpoints. Each midpoint ($u_{\text{center}}, v_{\text{center}}$) in image coordinate system can be converted to real world coordinates ($X_{\text{center}}, Y_{\text{center}}, Z_{\text{center}}$). Then, for each frame in the video, we perform the following steps:

- Identify the position of the green tape ($u_{\text{tape}}, v_{\text{tape}}$) in the image plane and convert its centroid to world coordinates ($X_{\text{tape}}, Y_{\text{tape}}, Z_{\text{tape}}$).
- Project the tape world coordinates onto the floor as the 3-dimensional world coordinates of the car ($X_{\text{car}}, Y_{\text{car}}, Z_{\text{car}}$). As shown in Fig. 3, the nearest midpoint on the track ($X_{\text{center}}, Y_{\text{center}}, Z_{\text{center}}$) is then located.
- Compute the distance between the car position ($X_{\text{car}}, Y_{\text{car}}, Z_{\text{car}}$) and the nearest track midpoint ($X_{\text{center}}, Y_{\text{center}}, Z_{\text{center}}$).

VI. RESULTS ANALYSIS

In this section, we compare our method with the SOTA method, Generative Cooperative Learning [12], to demonstrate the superior performance of our unsupervised anomaly video detection. Additionally, an ablation study is conducted

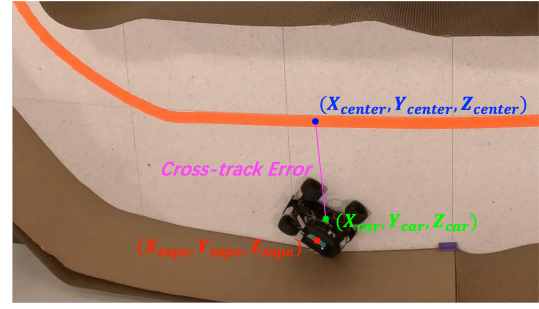


Fig. 3: Demonstration of our cross-track error calculation.

to evaluate the effectiveness of our designed latent reference loss $\mathcal{L}_{\text{refer}}$. Finally, the impact of the latent loss $\mathcal{L}_{\text{refer}}$ is analyzed and visualized using a t-SNE plot, which maps the 128-dimensional latent representations to a 2D space.

Comparative methods. We compare our pipeline with the state-of-the-art in unsupervised video anomaly detection — GCL [12] with the same structure and hyperparameters. We also compare to an ablated approach with just the reconstruction loss \mathcal{L}_{rec} . We apply all three approaches (ours, baseline, and ablation) to all three dirty datasets.

Unsupervised detection performance comparison. Unsupervised anomaly detection is achieved by reconstructing normal images more accurately than abnormal ones, measured with PCC. Normal images are expected to have higher PCC values compared to abnormal images. Figure 4 presents the PCC and median filtered PCC values for normal images and abnormal images (collected under conditions such as raindrops, wall hits, and plastic obstructions) in three column subplots. For a more accessible visualization of anomaly detection performance, we put all the abnormal images at the end of the test dataset (i.e., the last 1600 images).

As shown in the second row of Figure 4, our method leads to a clear drop in the smoothed PCC between normal and abnormal images for all three types of anomalies (raindrops, wall hits, and plastic obstructions), enabling their detection. In contrast, the SOTA method (GCL) exhibits a noticeable drop only for detecting wall hit anomalies. Meanwhile, the traditional CAE without our proposed latent reference loss fails to show any significant drop in reconstruction for any abnormal images. These results indicate that our proposed method enables the CAE to reconstruct abnormal images significantly worse than normal ones, improving anomaly detection. In comparison, the SOTA method and the conventional CAE (without our latent reference loss) tend to reconstruct both normal and abnormal images at similar levels in some or all anomaly scenarios, leading to detection failures.

To clean the dataset for imitation learning, we estimate a threshold to identify anomalous images. This threshold is determined as the median of the smoothed PCC value minus 0.05. When applying this threshold to detect anomalies using our method’s computed PCC values, nearly 100% of abnormal images are correctly identified with high precision. Our method achieves a recall of 1.000 for all three anomaly types, with precision values of 0.9639, 1.0000, and 0.9238 for Wall Hit, Raindrop, and Plastic Bag, respectively. Our

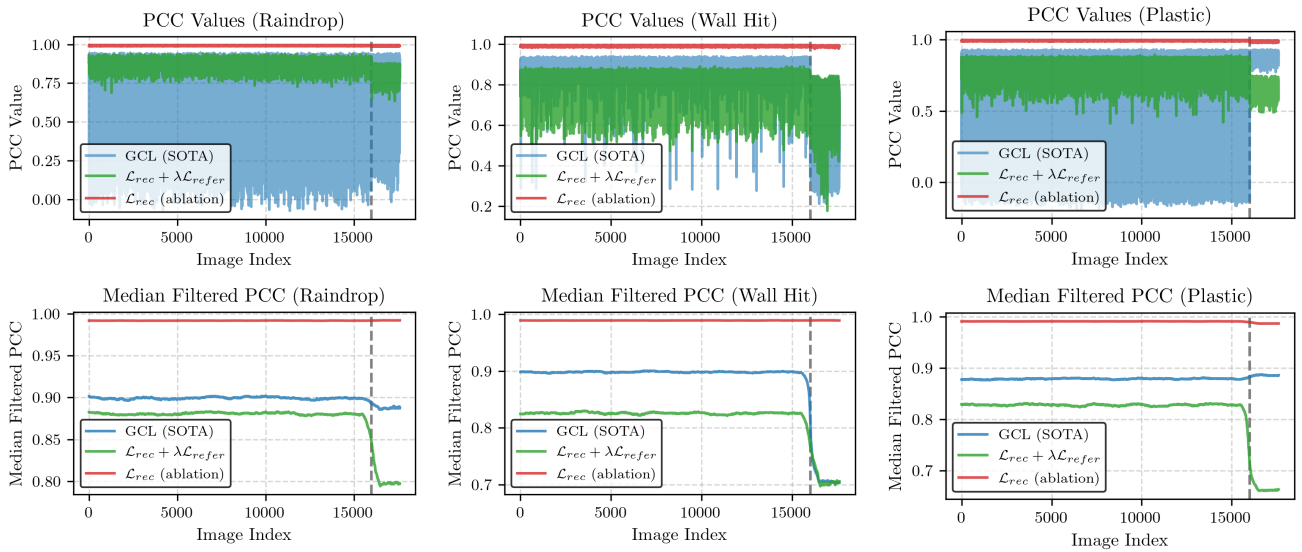


Fig. 4: Unsupervised detection performance (100 epochs for Raindrop and Plastic, 10 epochs for Wall Hit): our proposed method $\mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{refer}}$, GCL (SOTA), and ablation with only reconstruction loss \mathcal{L}_{rec} . For visualization, 1600 abnormal images are placed at the end, separated by a gray line.

method accurately detects all anomalies while preserving clean images. In contrast, the GCL (SOTA) method fails to detect anomalies caused by raindrops and plastic obstructions, yielding recall values of only 0.188 and 0.067. Furthermore, the ablation method (CAE without $\mathcal{L}_{\text{refer}}$) fails to detect any anomalies, with recall and precision values of 0.000 across all anomalies, as shown in Table II. These results highlight the effectiveness of our method in accurately identifying anomalies while maintaining high precision.

TABLE II: Comparison of Anomaly Detection Performance including Recall and Precision per Image Corruption (\uparrow).

	Hit Wall		Raindrop		Plastic Bag	
	Recall	Precision	Recall	Precision	Recall	Precision
CAE with $\mathcal{L}_{\text{refer}}$ (ours)	1.000	0.964	1.000	1.000	1.000	0.924
GCL (SOTA)	1.000	0.972	0.188	0.108	0.067	0.035
CAE without $\mathcal{L}_{\text{refer}}$ (ablation)	0.000	0.000	0.000	0.000	0.000	0.000

IL performance improvement with filtered dataset. The imitation learning performance is evaluated using CTE, measured from top-view images captured from the ceiling. As shown in Table III, training the CNN controller with a corrupted dataset containing raindrops, collisions, and plastic obstructions results in average CTE of 0.130, 0.135, and 0.117, respectively. In contrast, when using the dataset cleaned by our method, the average CTE decreases to 0.102, 0.089, and 0.072, respectively. We do not run the ablation and GCL (SOTA) for IL since they do not detect any anomalies: their controllers are equivalent to the all-data controller h . Thus, these results show that our method’s cleaning of anomalous images significantly (25-40% on average) improves imitation learning performance. A visual demonstration of IL improvement can be found in our video online: https://youtu.be/RjJ3nZR6_RQ.

TABLE III: Our method improves IL Performance with Cross-Track Error (Mean μ and Standard Deviation σ , \downarrow).

Training Data	Hit Wall		Raindrop		Plastic Bag	
	μ	σ	μ	σ	μ	σ
Controller \hat{h}_{clean} (cleaned data)	0.102	0.065	0.089	0.081	0.072	0.052
Controller h (all data)	0.130	0.110	0.135	0.128	0.117	0.093
Oracle controller h_{clean}	$\mu = 0.063$ $\sigma = 0.054$ (clean images only)					

Interpreting the effect of loss $\mathcal{L}_{\text{refer}}$ with t-SNE. To further demonstrate the effect of our designed latent reference loss $\mathcal{L}_{\text{refer}}$ in enhancing the reconstruction of normal images, we utilize *t-Distributed Stochastic Neighbor Embedding (t-SNE)* [24] to map the 128-dimensional latent representations of CAE into a 2D space while preserving local relationships, as shown in Figures 5 and 6.

- Without the latent reference loss $\mathcal{L}_{\text{refer}}$ (Ablation method), shown in Figure 5: as training epochs increase (from 5 to 95), the t-SNE representations of normal and abnormal images gradually form two distinct clusters. This occurs because normal and abnormal images differ in nature, and effective reconstruction requires their latent representations to have distinct distributions.
- With $\mathcal{L}_{\text{refer}}$ (our method) shown in Figure 6: after applying our latent reference loss, the t-SNE representations of normal and abnormal images remain mixed as training progresses. This hinders the reconstruction of abnormal images because their latent representations are positioned close to those of normal images. As a result, the decoded abnormal and normal images become more similar. Since normal and abnormal images have inherent differences, abnormal images are more likely to be reconstructed as normal ones, leading to worse reconstruction quality.

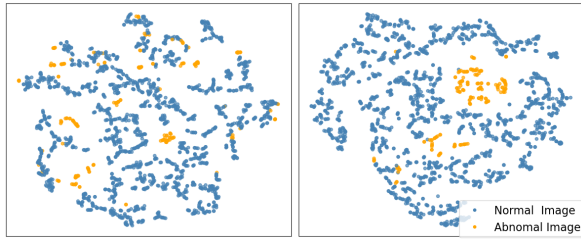


Fig. 5: Latent visualization *without* $\mathcal{L}_{\text{refer}}$ (left: 5 epochs; right: 95 epochs): abnormal images are clustered, improving their reconstruction and impeding detection.

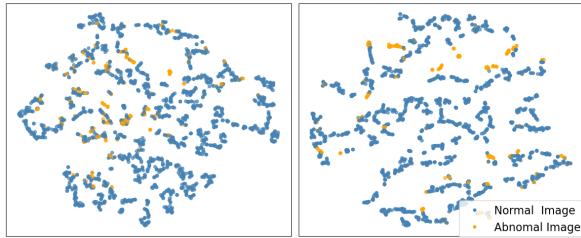


Fig. 6: Latent visualization *with* $\mathcal{L}_{\text{refer}}$ (left: 5 epochs; right: 95 epochs): abnormal images are spread out, impeding their reconstruction and improving detection.

In summary, our proposed latent representation loss enabled the CAE to reconstruct abnormal images less accurately than normal ones by making the CAE generate abnormal images closer to normal images rather than themselves. This reconstruction discrepancy enhanced the effectiveness in achieving unsupervised anomaly image detection.

VII. CONCLUSION AND FUTURE WORK

This paper introduces a novel unsupervised anomaly detection method to enhance the quality of imitation learning datasets by removing abnormal videos. Using an autoencoder with latent reference loss, the method ensures higher reconstruction errors for anomalies. Extensive experiments on the DonkeyCar platform showed that removing abnormal frames significantly enhances imitation learning performance, as measured by reduced cross-track error. Compared to state-of-the-art and ablation baselines, our method consistently outperforms in detecting anomalies across various real-world distortions. Future work will incorporate multi-modal inputs (LiDAR, radar) and improve anomaly detection with adaptive thresholds and uncertainty-aware filtering, enabling automated dataset cleaning to boost real-world imitation learning.

REFERENCES

- [1] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14128–14147, 2022.
- [2] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on robot learning*. PMLR, 2022, pp. 158–168.
- [3] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," *arXiv preprint arXiv:1805.01954*, 2018.
- [4] D. M. Jacobs and C. F. Michaels, "Direct learning," *Ecological psychology*, vol. 19, no. 4, pp. 321–349, 2007.

- [5] S. Belkhale, Y. Cui, and D. Sadigh, "Data quality in imitation learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [6] B. Zheng, S. Verma, J. Zhou, I. W. Tsang, and F. Chen, "Imitation learning: Progress, taxonomies and challenges," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 5, pp. 6322–6337, 2024.
- [7] J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin, "Deep industrial image anomaly detection: A survey," *Machine Intelligence Research*, vol. 21, no. 1, pp. 104–135, 2024.
- [8] D. Liu, T. Jiang, and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1298–1307.
- [9] Z. Liu, L. Wang, Q. Zhang, Z. Gao, Z. Niu, N. Zheng, and G. Hua, "Weakly supervised temporal action localization through contrast based evaluation networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3899–3908.
- [10] M. Abdalla, S. Javed, M. A. Radi, A. Ulhaq, and N. Werghi, "Video anomaly detection in 10 years: A survey and outlook," *ArXiv*, vol. abs/2405.19387, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:270123354>
- [11] D. Li, P. Auerbach, and O. Okhrin, "Autonomous driving small-scale cars: A survey of recent development," *arXiv preprint arXiv:2404.06229*, 2024.
- [12] M. Z. Zaheer, A. Mahmood, M. H. Khan, M. Segu, F. Yu, and S.-I. Lee, "Generative cooperative learning for unsupervised video anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14744–14754.
- [13] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," *arXiv preprint arXiv:1805.01954*, 2018.
- [14] Z. Cao, Z. Wang, and D. Sadigh, "Learning from imperfect demonstrations via adversarial confidence transfer," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 441–447.
- [15] R. Nayak, U. C. Pati, and S. K. Das, "A comprehensive review on deep learning-based methods for video anomaly detection," *Image and Vision Computing*, vol. 106, p. 104078, 2021.
- [16] S. Qiu, J. Ye, J. Zhao, L. He, L. Liu, B. E., and X. Huang, "Video anomaly detection guided by clustering learning," *Pattern Recognition*, vol. 153, p. 110550, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320324003017>
- [17] Z. Lu, I. Afridi, H. J. Kang, I. Ruchkin, and X. Zheng, "Surveying neuro-symbolic approaches for reliable artificial intelligence of things," *Journal of Reliable Intelligent Environments*, Jul. 2024. [Online]. Available: <https://doi.org/10.1007/s40860-024-00231-1>
- [18] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.
- [19] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1705–1714.
- [20] G. Pang, C. Yan, C. Shen, A. van den Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12173–12182.
- [21] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9584–9592.
- [22] H. Rahadian, S. Bandong, A. Widyotriatmo, and E. Joelianto, "Image encoding selection based on pearson correlation coefficient for time series anomaly detection," *Alexandria Engineering Journal*, vol. 82, pp. 304–322, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1110016823008657>
- [23] K. Yang, S. Kim, and J. B. Harley, "Unsupervised long-term damage detection in an uncontrolled environment through optimal autoencoder," *Mechanical Systems and Signal Processing*, vol. 199, p. 110473, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327023003813>
- [24] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. 11, 2008.