Structural Attention-Based Recurrent Variational Autoencoder for Highway Vehicle Anomaly Detection

Neeloy Chakraborty
University of Illinois
Urbana, IL, United States
neeloyc2@illinois.edu

Aamir Hasan*
University of Illinois
Urbana, IL, United States
aamirh2@illinois.edu

Shuijing Liu*
University of Illinois
Urbana, IL, United States
sliu105@illinois.edu

Tianchen Ji*
University of Illinois
Urbana, IL, United States
tj12@illinois.edu

Weihang Liang
University of Illinois
Urbana, IL, United States
weihang2@illinois.edu

D. Livingston McPherson
University of Illinois
Urbana, IL, United States
dlivm@illinois.edu

Katherine Driggs-Campbell
University of Illinois
Urbana, IL, United States
krdc@illinois.edu

ABSTRACT

In autonomous driving, detection of abnormal driving behaviors is essential to ensure the safety of vehicle controllers. Prior works in vehicle anomaly detection have shown that modeling interactions between agents improves detection accuracy, but certain abnormal behaviors where structured road information is paramount are poorly identified, such as wrong-way and off-road driving. We propose a novel unsupervised framework for highway anomaly detection named Structural Attention-Based Recurrent VAE (SABeR-VAE), which explicitly uses the structure of the environment to aid anomaly identification. Specifically, we use a vehicle self-attention module to learn the relations among vehicles on a road, and a separate lane-vehicle attention module to model the importance of permissible lanes to aid in trajectory prediction. Conditioned on the attention modules' outputs, a recurrent encoder-decoder architecture with a stochastic Koopman operator-propagated latent space predicts the next states of vehicles. Our model is trained end-to-end to minimize prediction loss on normal vehicle behaviors, and is deployed to detect anomalies in (ab)normal scenarios. By combining the heterogeneous vehicle and lane information, SABeR-VAE and its deterministic variant, SABeR-AE, improve abnormal AUPR by 18% and 25% respectively on the simulated MAAD highway dataset over STGAE-KDE. Furthermore, we show that the learned Koopman operator in SABeR-VAE enforces interpretable structure in the variational latent space. The results of our method indeed show that modeling environmental factors is essential to detecting a diverse set of anomalies in deployment. For code implementation, please visit https://sites.google.com/illinois.edu/saber-vae.

KEYWORDS

Anomaly Detection; Autonomous Vehicles; Unsupervised Learning; Human Behavior Modeling

1 INTRODUCTION

Autonomous vehicles have the potential to realize a fast, safe, and labor-free transportation system. A trustworthy self-driving vehicle should have the ability to operate reliably in normal situations and, more importantly, to perceive and react to anomalous driving scenarios (e.g., skidding and wrong-way driving of surrounding human vehicles) promptly and robustly. The detection of such abnormal situations can help identify traffic accidents and dangerous driving behaviors of road participants, and thus provide high-level guidance for vehicle controllers to act safely.

Deep-learning based Anomaly Detection (AD) algorithms have shown great promise in intelligent vehicle applications [7]. Many previous works utilize vehicle trajectories as an anomaly signal [2, 12, 43]. However, only a few vehicle trajectory datasets with sufficient anomaly labels exist for supervised learning methods [18, 43, 47]. To leverage the larger store of unlabeled driving data, researchers like Yao and Wiederer have employed unsupervised learning methods [42, 44, 45]. Specifically, a neural network, which generally follows an encoder-decoder architecture for trajectory reconstruction or prediction, learns an underlying distribution of normal vehicle trajectories in the latent space. An anomaly is then detected whenever the trajectory is out of distribution and produces a large reconstruction or prediction error. In interactive driving scenarios, Wiederer *et al.* [42] showed that modeling interactions between agents can largely improve the reconstruction accuracy and subsequently the AD performance. However, such interaction-aware methods still ignore the effect of road structures on vehicle behaviors, and thus can miss abnormal scenarios like wrong-way

---

driving trajectories that appear normal when environmental context is overlooked.

Alongside performance accuracy, the decisions made by AD algorithms need to be interpretable to stakeholders. Deep neural networks are black boxes by nature. However, the decisions of deep networks impact various stakeholders such as policy makers and end users. Designing methods with *interpretable* features for stakeholders is a key challenge in AD, and the field of machine learning overall [5, 16, 37, 39]. In vehicle AD more specifically, interpretable algorithms need to account for the wide distribution of human drivers who act according to their own policies [6]. For example, different drivers may choose to overtake other vehicles at different times and speeds. To ensure interpretability, we use variational autoencoder (VAE) to cluster useful features from similar behaviors together in a continuous and stochastic latent space [26]. Our results indicate that vehicle trajectories transitioning to an abnormal state are explicitly encoded by interpretable transformations in the learned latent space.

In this paper, we present our novel unsupervised Structural Attention-Based Recurrent Variational Autoencoder (SABeR-VAE) for highway vehicle anomaly detection. Since contemporary vehicles have map information available to them regarding their nearby environment and lanes, we make use of the environmental information that prior works [29, 36, 42] have ignored to explicitly model the effect of lane structure on normal vehicle behaviors. Specifically, we treat a highway scenario as a structured interaction graph where nodes represent vehicles and lane positions, and edges connect nearby vehicles, and permissible lanes. Two separate attention modules learn relations between vehicles (vehicle-vehicle self-attention) and legal permissible route trajectories (lane-vehicle attention) respectively. A sequence of embeddings from the vehicle-vehicle attention module are encoded into a Gaussian latent space to capture the randomness of vehicle trajectories with a recurrent network, and cluster similar behaviors close together in an interpretable fashion. Our work is more computationally efficient than STGAE-KDE [42], which has a deterministic latent space and requires the expensive process of fitting a Kernel Density Estimator (KDE) to learn a meaningful distribution of normal behaviors. We then use a learned Koopman operator to propagate the current latent distributions forward in time conditioned on the useful lane embeddings. We show that the Koopman operator explicitly enforces interpretable transformations in the latent space that standard autoencoders like STGAE are unable to incorporate, and is able to model the complex, non-linear dynamics of drivers. Finally, we decode a sampled point from the propagated distribution to predict next states of vehicles. We train our method to predict trajectories from normal scenarios in the Multi-Agent Anomaly Detection (MAAD) dataset [42], and compare accuracy metrics against linear, recurrent, and graph convolutional approaches on anomalous trajectories [34, 36, 42]. Our SABeR-VAE improves AUPR-Abnormal and wrong-way driving detection over the STGAE-KDE by 18% and 35% respectively, and has an interpretable latent space.

Our contributions can be summarized as follows: (1) We present a novel unsupervised variational approach for anomaly detection conditioned on structured lane information; (2) We quantitatively show that incorporating the structured information increases anomaly detection accuracy, compared with state-of-the-art baselines and
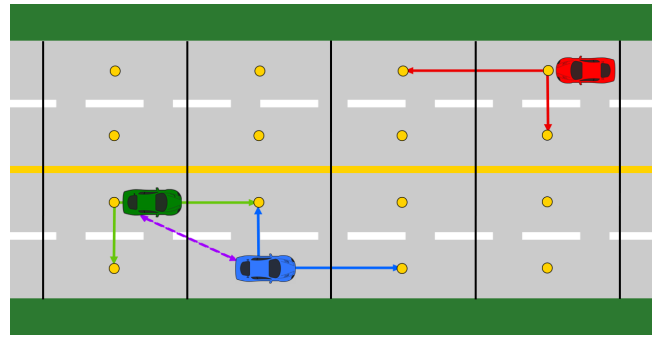
Figure 1: Map discretization and interaction edges. We model the vehicle AD problem as an interaction graph with vehicle and lane nodes. A continuous map of the road is discretized into blocks. Directed lane edges between lane nodes encode permissible routes for vehicles. The red vehicle has a directed edge toward the lane nodes in front and to its left because the driver can legally continue forward or merge left. Conversely, the green vehicle has no edge connecting to a left lane node since it cannot cross the road divider. Vehicle edges, shown in purple, exist for vehicles that are close enough to interact with each other.

ablations using the MAAD dataset; (3) We show that the stochastic Koopman operator learns interpretable features of (ab)normal behaviors in the latent space.

Our paper is organized as follows: Section 2 discusses relevant works in the areas of structured modeling and anomaly detection. Our problem formulation and methods are presented in Section 3. We discuss results in Section 4. Finally, we conclude the paper and discuss future directions in Section 5.

2 RELATED WORKS

2.1 Exploiting Map Information

The quality of information about an environment provided by High Definition maps (HD-maps) has dramatically increased and led to their ubiquitous use due to recent advancements in autonomous driving [30, 46]. Currently, most state-of-the-art methods for vehicle trajectory prediction, motion forecasting, and anomaly detection, do not make effective use of the rich information provided in these HD-maps, and only rely on modeling the interactions between vehicles on the road [9, 27, 38]. Hence, these methods ignore vital information such as the plausible movement of vehicles in the environment, which can be paramount in identifying anomalies such as wrong-way driving.

However, trajectory prediction methods such as those proposed by Deo *et al.* and Liang *et al.* do exploit the information in these HD-maps and significantly outperform their counterparts [17, 28]. In proposing LaneGCN, Liang *et al.* encode different types of interactions between agents on the road with lane information extracted from maps. [28]. They show that attention-based models can be used to encode interactions between vehicles and lanes, which are learned by constructing a graph representation of the road. PGP, proposed by Deo *et al.* , further produces scene-compliant trajectories by sampling from a distribution of driving profiles conditioned on environment and vehicle interactions [17]. We corroborate the

usefulness of these vehicle and lane attention-based representations and show that such embeddings do in fact provide meaningful insights in detecting highway vehicle anomalies in SABeR-VAE.

2.2 Variational Autoencoders for Sequences

Variational autoencoders (VAE) have been applied to sequential data combined with recurrent neural networks (RNN) in fields such as speech and image synthesis and autonomous driving [8, 13–15, 19, 29, 34]. Liu *et al.* attempt to infer the traits of drivers from trajectories encoded in a variational latent space [29]. However, only two classes of traits and a restricted set of defined trajectories were considered, while real drivers have a much wider range of behaviors on the road. Furthermore, they do not utilize map information in their learning process, which provide relevant context for traits. Conditional VAE formulations have also been found to be able to generate trajectories with different driving styles, but fail to consistently produce feasible trajectories without necessary environment context [22, 35, 40]. Recurrent VAEs have also been applied to robot anomaly detection, but are limited by the simplicity of the single agent problem statement [34]. These sequential generative modeling approaches perform reasonably on their simple tasks, but fail to generate realistic samples from points in the latent space in more complex areas, due to the limitations of their RNN components [13, 14, 20].

To bridge the gap between complex human behaviors and the structured environment, and overcome the hurdles of the temporal propagation in simplistic RNNs, we propose the use of a lane-conditioned Koopman Operator to model the temporal relations in the latent space. We were specifically inspired to use the Koopman operator to propagate the latent space due to its capability to model the dynamics of complex, non-linear data, including fluid dynamics, battery properties, and control tasks [1, 3, 4, 31].

2.3 Anomaly Detection

Anomaly detection is well studied in diverse research areas and application domains [11, 32]. In robotics and automated vehicles, AD has been used to detect abnormal patterns such as robot failures [24, 33] and dangerous driving scenarios [42, 45].

Park *et al.* propose a long short-term memory based variational autoencoder (LSTM-VAE) to reconstruct the expected distribution of robot sensor signals. A reconstruction-based anomaly score is then used for anomaly detection [34]. Furthermore, Ji *et al.* adopt an attention mechanism to fuse multi-sensor signals for robust anomaly detection in uncertain environments [23]. While these approaches focus on AD for single agent problem statements, our highway scenarios consist of complex multi-agent social interactions among vehicles, and need to be modeled as such.

In the domain of traffic anomaly detection using multi-agent trajectories, the most similar work to ours is the spatio-temporal graph autoencoder (STGAE) proposed with the MAAD dataset [42]. The architecture follows an encoder-decoder structure to reconstruct vehicle trajectories, where vehicle interactions and motions are considered using spatial graph convolution and temporal convolution layers, respectively. The method has been shown to be effective by modeling interactions among vehicles to detect anomalous maneuvers in traffic. However, such a network ignores the constraints imposed by road structures on vehicle trajectories and

the variability of human driver behaviors. In this work, we explicitly model both vehicle-to-vehicle interactions and lane-to-vehicle interactions to boost performance, and use an interpretable variational architecture to learn a continuous distribution over behaviors.

3 METHODOLOGY

In this section, we first introduce our problem formulation of anomaly detection from vehicle trajectories, and then explain our proposed SABeR-VAE framework.

3.1 Problem Formulation

Suppose $n_t \in [1, N]$ vehicles are on a road segment at any time t, and each vehicle takes an acceleration and steering action every timestep according to unknown policies. Let $c_t^{(i)} = \left(x_t^{(i)}, y_t^{(i)}\right)$ be the 2D coordinates of the ith vehicle at time t, where $i \in [1, ..., n_t]$. Each vehicle also has a set of corresponding permissible lane positions in front, to the left, and to the right of the vehicle, provided in the form of a discretized map representation shown in Fig. 1. At every timestep, each vehicle's position within the map is used to identify their corresponding front, left, and right lane nodes. We define a tuple $l_t^{(i)} = (\text{front}, \text{left}, \text{right})_t^{(i)}$ of three 2D coordinates containing the lane information for vehicle i at time t. The discretization step only impacts $l_t^{(i)}$. Altogether, the observed information of each vehicle at any time is the relative displacement of coordinates $o_t^{(i)} = \left(c_t^{(i)} - c_{t-1}^{(i)}, l_t^{(i)} - c_t^{(i)}\right) = \left(X_t^{(i)}, L_t^{(i)}\right)$. A trajectory of length T for any vehicle is represented as $\left(o_0^{(i)}, o_1^{(i)}, ..., o_{T-1}^{(i)}\right)$. We assume that any vehicle A that is within a distance d to another vehicle B at time t can accurately detect and track the relative coordinates $c_t^{(B)} - c_t^{(A)} = R_t^{(A,B)}$. The purple arrow between the green and blue vehicle in Fig. 1 represents this vehicle interaction type. For the i-th car, the number of observable cars is $m_t^i \in [0, n_t - 1]$. Given all vehicle trajectories in a scene, our goal is to provide an anomaly score $\text{AS}_t \in \mathbb{R}_{\geq 0}$ for each time t.
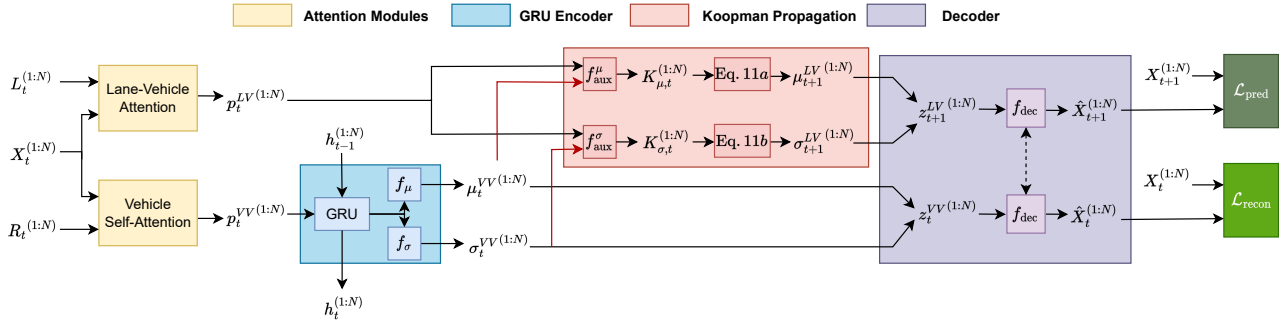
3.2 Architecture

Figure 2 contains the complete architecture diagram of SABeR-VAE, which we discuss in this section.

3.2.1 *Vehicle-Vehicle Self-Attention Network.* Our goal is to learn a representation of spatial interactions among vehicles. Rather than using convolutional methods like those in prior works [28, 42], we encode the positions of vehicles on the road at each time with scaled dot-product multi-head self-attention, which allows each head to learn different features of the data [41].

We embed the displacement of each car X_t with a multi-layer perceptron (MLP) f_Q^{VV} to obtain queries Q_t^{VV}:

$$f_Q^{VV}(X_t) = Q_t^{VV} \in \mathbb{R}^{1 \times D}, \tag{1}$$

where D is the attention size. Let $\mathbf{R}_t = \left[R_t^{(i,1)}, ..., R_t^{(i,m)}\right]^{\top}$ be the displacements of all neighboring cars for the i-th car. We use two other MLPs, f_K^{VV} and f_V^{VV}, to embed $\mathbf{R}_t$ to obtain keys K_t^{VV} and

Figure 2: SABeR-VAE architecture. The SABeR-VAE architecture attempts to predict the one-step future states of vehicles conditioned on current vehicle positions and structural lane information. Vehicle interactions are modeled by the self-attention module while permissible routes are encoded by the lane-vehicle attention module. A GRU encoder processes the self-attention embeddings through time to produce a latent distribution. Then the Koopman operator conditioned on the lane embeddings propagates the latent distributions forward, which finally get decoded to predict next states. The f_{dec} network shares parameters for reconstruction and prediction.

values V_t^{VV} respectively:

$$f_K^{VV}(\mathbf{R}_t) = K_t^{VV} \in \mathbb{R}^{m \times D}$$
$$f_V^{VV}(\mathbf{R}_t) = V_t^{VV} \in \mathbb{R}^{m \times D} \tag{2}$$

The final encoding of each vehicle position from this self-attention layer for time t is calculated as:

$$\text{softmax}\left(\frac{Q_t^{VV}\left(K_t^{VV}\right)^{\top}}{\sqrt{D}}\right) V_t^{VV} = p_t^{VV} \in \mathbb{R}^{1 \times D} \tag{3}$$

Nonexistent or unobserved vehicles further than a distance d cannot be allowed to contribute to the attention score of other vehicles. Thus, we use a mask to set the score contributed from unobserved vehicles to $-\infty$.

*3.2.2 **Lane-Vehicle Attention Network**.* We use available map and lane information in a separate lane-vehicle attention layer to model legal maneuvers in structured environments. Similar to vehicle-vehicle attention, the query Q_t^{LV} is an embedding of X_t. Lane information of each vehicle L_t is used to produce keys K_t^{LV} and values V_t^{LV}:

$$f_Q^{LV}(X_t) = Q_t^{LV} \in \mathbb{R}^{1 \times D}$$
$$f_K^{LV}(L_t) = K_t^{LV} \in \mathbb{R}^{3 \times D} \tag{4}$$
$$f_V^{LV}(L_t) = V_t^{LV} \in \mathbb{R}^{3 \times D}$$

The lane-conditioned vehicle embeddings are calculated as:

$$\text{softmax}\left(\frac{Q_t^{LV}\left(K_t^{LV}\right)^{\top}}{\sqrt{D}}\right) V_t^{LV} = p_t^{LV} \in \mathbb{R}^{1 \times D} \tag{5}$$

Note that all three lane nodes may not always be permissible to a vehicle. For example, a car in the left-most lane of a road is unable to legally turn left. As such, we mask out impermissible lane nodes like in the self-attention layer.

*3.2.3 **Recurrent Encoder**.* A gated recurrent unit (GRU) network encodes the sequence of self-attention features for each vehicle $\left(p_0^{VV}, p_1^{VV}, ..., p_{T-1}^{VV}\right)$ into a sequence of Gaussian distributions in the latent space with temporal correlation. Thus, the latent space

captures the stochastic nature of human behaviors. Specifically, after embedding the vehicle-vehicle attention feature with a network f_e, we pass the embedding through the GRU to get the hidden state of each vehicle for the current timestep:

$$h_t = \text{GRU}\left(h_{t-1}, f_e\left(p_t^{VV}\right)\right) \tag{6}$$

Mean and variance neural networks f_μ and f_σ produce parameters for a latent normal distribution of dimension j conditioned on a vehicle's hidden state at any time:

$$\mu_t^{VV} = f_\mu(h_t), \quad \sigma_t^{VV} = f_\sigma(h_t). \tag{7}$$

*3.2.4 **Latent Propagation with Koopman Operator**.* While the GRU encoder encodes vehicle behaviors into the latent space solely conditioned on past and current vehicle interactions, we need a method to propagate the latent distributions in time to predict the future states of vehicles. To this end, we learn a stochastic Koopman operator conditioned on the lane-vehicle embeddings to perform this task, like Balakrishnan and Upadhyay [4]. The Koopman operator is responsible for temporal reasoning (modeling vehicle state dynamics), while the preceding attention modules take charge of spatial reasoning.

In Koopman operator theory, a discrete time system evolves according to potentially nonlinear dynamics $x_{t+1} = F(x_t)$. However, a function g maps the state x_t into a space where dynamics evolve linearly with the Koopman operator $\mathcal{K}$ [4]:

$$\mathcal{K}g(x_t) = g(F(x_t)) = g(x_{t+1}) \tag{8}$$

Similarly, the inverse of function g translates an observable of x back into the original dynamics space [4]:

$$g^{-1}(\mathcal{K}g(x_t)) = x_{t+1} \tag{9}$$

In our case, function g is represented by the GRU encoder and neural networks f_μ and f_σ, which altogether, produce a latent distribution $\mathcal{N}(\mu_t, \sigma_t)$ conditioned on inter-vehicle embeddings p_t^{VV}.

Like the Stochastic Adversarial Koopman (SAK) model [4], we use auxiliary neural networks f_{aux}^μ and f_{aux}^σ to predict tridiagonal Koopman matrices $K_{\mu,t}$ and $K_{\sigma,t}$, rather than solving for their closed form solution. The outputs of f_{aux}^μ and f_{aux}^σ are conditioned on the current latent distributions $\mathcal{N}\left(\mu_t^{VV}, \sigma_t^{VV}\right)$ and the lane features

p_t^{LV}, so that the Koopman operators capture legal route maneuvers in the latent space propagation:

$$K_{\mu,t} = f_{\text{aux}}^{\mu}\left(\mu_t^{VV}, p_t^{LV}\right)$$
$$K_{\sigma,t} = f_{\text{aux}}^{\sigma}\left(\sigma_t^{VV}, p_t^{LV}\right) \tag{10}$$

The predicted Koopman matrices are applied to the inter-vehicle distributions to linearly propagate the mean and variance of the latent distributions forward in time:

$$\mu_{t+1}^{LV} = K_{\mu,t}\,\mu_t^{VV} + \mu_t^{VV} \tag{11a}$$
$$\sigma_{t+1}^{LV} = K_{\sigma,t}\,\sigma_t^{VV} + \sigma_t^{VV} \tag{11b}$$

Intuitively, we can interpret the GRU encoder as predicting a distribution of vehicle behaviors from their current trajectories, and the Koopman operator propagates to a one-step future distribution of behaviors based on lane information.

3.2.5 ***The Decoder Network***. At this point, we have two sets of distributions in the j-dimensional latent space for the current states and future predictions of vehicles at each time: $\mathcal{N}\left(\mu_t^{VV}, \sigma_t^{VV}\right)$ and $\mathcal{N}\left(\mu_{t+1}^{LV}, \sigma_{t+1}^{LV}\right)$. We utilize the reparameterization trick to sample a point from each of the distributions:

$$\epsilon_t^{VV} \sim \mathcal{N}(0,1) \quad z_t^{VV} = \mu_t^{VV} + \epsilon_t^{VV}\sigma_t^{VV}$$
$$\epsilon_{t+1}^{LV} \sim \mathcal{N}(0,1) \quad z_{t+1}^{LV} = \mu_{t+1}^{LV} + \epsilon_{t+1}^{LV}\sigma_{t+1}^{LV} \tag{12}$$

A multi-layer perceptron f_{dec} is used as a decoder network, similar to g^{-1} in Eq. 9, to predict a vehicle coordinate change from the sampled latent points:

$$\hat{X}_t = f_{\text{dec}}\left(z_t^{VV}\right), \qquad \hat{X}_{t+1} = f_{\text{dec}}\left(z_{t+1}^{LV}\right). \tag{13}$$

3.3 Training and Evaluation

3.3.1 ***End-to-End Training***. To fairly compare our method with prior convolutional approaches, we utilize a similar sliding window training approach performed by Wiederer *et al.* [42]. Specifically, whole trajectories of length T are divided into small overlapping segments, or windows, of constant length T'.

In our training objective, we minimize the current reconstruction loss and one-step future prediction loss of the model by splitting our input ground truth trajectories into current states $X^- = X_{0:T'-2}$ and one-step future states $X^+ = X_{1:T'-1}$. We also regularize the current distributions $\mathcal{N}\left(\mu_t^{VV}, \sigma_t^{VV}\right)$ and propagated distributions $\mathcal{N}\left(\mu_{t+1}^{LV}, \sigma_{t+1}^{LV}\right)$ to follow a standard normal distribution. Let $D_{\text{KL}}(\mu, \sigma)$ be the KL divergence between any Gaussian distribution $\mathcal{N}(\mu, \sigma)$ and the standard normal distribution $\mathcal{N}(0,1)$. Then the regularized prediction and reconstruction losses are:

$$\mathcal{L}_{\text{pred}} = \beta_1 \cdot D_{\text{KL}}\left(\mu^{LV}, \sigma^{LV}\right) + \|X^+ - \hat{X}^+\|_2$$
$$\mathcal{L}_{\text{recon}} = \beta_2 \cdot D_{\text{KL}}\left(\mu^{VV}, \sigma^{VV}\right) + \|X^- - \hat{X}^-\|_2, \tag{14}$$

where β_1 and β_2 are tunable weights applied to the regularization of the latent distributions similar to beta-VAE [21].

The overall objective we optimize is:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{recon}} \tag{15}$$
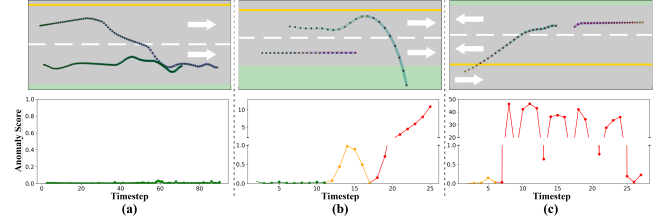
Figure 3: Trajectories and SABeR-VAE anomaly scores. (top row) Examples of a normal overtaking (a,) abnormal off-road driving (b,) and wrong-way driving (c) scenarios in the MAAD dataset. White arrows point toward direction of normal traffic flow. **(bottom row)** Predicted anomaly score curves for each scenario above. Colors of lines within the curves show the ground-truth labels of normal (green,) ignored (yellow,) & abnormal (red) timesteps.

We again mask out coordinates of unobserved vehicles so they do not contribute to the loss.

While SAK [4] applies maximum mean discrepancy (MMD) to synchronize the current and propagated distributions of their Koopman model to any general distribution, we explicitly encourage the latent space distributions to follow the standard gaussian. We leave experimentation of various Koopman synchronization methods for the anomaly detection task as a future direction of research.

3.3.2 ***Anomaly Detection Evaluation***. At test time, we follow the same sliding window practice as performed in training. First, we calculate the one-step future prediction loss $\mathcal{L}_{\text{pred},t+1} = \|X_{t+1}^{(i)} - \hat{X}_{t+1}^{(i)}\|_2$ for every vehicle at each timestep, within every window of a complete trajectory.

Then, we average the prediction loss of overlapping timesteps among all windows in the sequence, for each vehicle separately. Suppose $\mathcal{W}^{(t,i)}$ is the set of all windows in the complete trajectory containing time t where vehicle i is observed. The averaged prediction error for car i at t is:

$$\bar{\mathcal{L}}_{\text{pred},t}^{(i)} = \frac{\sum_{w \in \mathcal{W}^{(t,i)}} \mathcal{L}_{\text{pred},w_t^{(i)}}}{|\mathcal{W}^{(t,i)}|}, \tag{16}$$

where $\mathcal{L}_{\text{pred},w_t^{(i)}}$ is the prediction error of time t for vehicle i in window w of the set $\mathcal{W}^{(t,i)}$.

Finally, we choose the anomaly score AS to be the maximum averaged prediction loss over all vehicles at a given timestep t:

$$\text{AS}_t = \max_{i=1,\ldots,n_t} \bar{\mathcal{L}}_{\text{pred},t}^{(i)} \tag{17}$$

4 EXPERIMENTAL SETUP AND RESULTS

In this section, we first describe the MAAD dataset on which we performed experiments and detail baselines and ablations. We also present our quantitative results and latent space interpretations.

4.1 MAAD Dataset and Augmentation

The MAAD dataset [42] consists of $2D$ trajectories of two vehicles on a straight two-lane highway with a divider separating the two possible directions, as visualized in the top row of Fig. 3. There are 80 training and 66 test-split trajectories ranging from a length of 25

Table 1: Accuracy results of baselines, ablations, and SABeR methods over ten runs.

Method	Detection Type	AUROC ↑	AUPR-Abnormal ↑	AUPR-Normal ↑	FPR @ 95%-TPR ↓
CVM	Reconstruction Loss	83.1 ± 0.0	54.5 ± 0.0	96.0 ± 0.0	74.6 ± 0.0
RAE-Recon[*]	Reconstruction Loss	56.2 ± 0.7	16.9 ± 1.0	89.5 ± 0.1	84.6 ± 0.3
STGAE[*]	Reconstruction Loss	74.8 ± 5.1	37.8 ± 7.2	94.1 ± 1.3	77.8 ± 9.8
STGAE-KDE[*]	One Class	86.3 ± 1.7	55.2 ± 7.7	**97.2 ± 0.5**	**50.0 ± 7.9**
RAE-Pred	Prediction Loss	72.5 ± 15.3	43.5 ± 17.4	92.9 ± 4.4	75.8 ± 10.6
VV-RAE	Prediction Loss	54.2 ± 4.9	14.8 ± 0.9	89.5 ± 2.4	77.1 ± 7.3
Att-LSTM-VAE	Prediction Loss	85.8 ± 0.7	64.9 ± 0.9	96.5 ± 0.3	66.6 ± 5.9
SABeR-AE	Prediction Loss	**87.2 ± 0.4**	**69.0 ± 0.5**	96.9 ± 0.2	64.1 ± 5.0
SABeR-VAE	Prediction Loss	87.0 ± 1.5	65.5 ± 2.9	96.9 ± 0.5	57.7 ± 7.6

[*] These results are presented in [42].

to 127 timesteps. To compare fairly with baselines, these dynamic length trajectories are subsampled to produce approximately 6.3K training and 3.1K testing windows of constant length $T' = 15$. As the original dataset sequences did not come with map or lane details, we augmented the data to include this information. Specifically, we discretized the highway in the x-coordinate direction into blocks of length five meters as shown in Fig. 1, and stored the $2D$ coordinates of the front, left, and right blocks for each vehicle at every timestep in all trajectories. We chose a discretization factor of five meters because vehicles traveled on average five meters or less every timestep. All the training sequences consist of normal vehicle behaviors like driving side-by-side, overtaking, following, and driving in opposite directions. In contrast, the test-split contains both normal and 11 anomalous behavior classes like aggressive overtaking, pushing aside, tailgating, off-road, and wrong-way driving.

4.2 Baseline Methods

We compare against baselines implemented by Wiederer *et al.* that depend on reconstruction loss rather than future prediction error [42]. **(1)** The Constant Velocity Model (**CVM**) is a standard baseline that predicts the next states of vehicles assuming each vehicle travels at the same velocity as the last timestep, without modeling any inter-vehicle relations. **(2)** Recurrent Autoencoder (**RAE-Recon**) uses an LSTM network to encode and decode a sequence of coordinates from an unregularized latent space, attempting to minimize reconstruction loss. **(3)** Spatio-Temporal Graph Autoencoder (**STGAE**) is a convolutional method that models inter-vehicle behaviors, and outputs parameters for a bi-variate distribution describing the estimated state of the reconstructed pose of vehicles, and is trained to maximize the log-likelihood of the estimated probability distribution. Finally, **(4)** the **STGAE-KDE** baseline fits a Kernel Density Estimator (KDE) to the unregularized latent space of a trained STGAE model to predict the one-class probability of a set of points originating from a normal behavior window. Unlike the STGAE-KDE, our SABeR-VAE does not require an expensive KDE fitting procedure since our anomaly score solely relies on prediction error, and we still model inter-vehicle relations unlike CVM and RAE-Recon.

We additionally train ablation models with future prediction loss to identify the impact of different components in our method. We train **(5)** an unregularized Recurrent Autoencoder (**RAE-Pred**)

using a standard deterministic MLP to propagate latent points forward in time, without explicitly modeling any inter-vehicle behaviors, like the RAE-Recon. **(6)** A Recurrent Autoencoder with a vehicle-vehicle Self-Attention module (**VV-RAE**) minimizes prediction error while modeling inter-vehicle relations. We also train **(7)** a deterministic variant of SABeR-VAE without a regularized latent space, **SABeR-AE**. SABeR-AE utilizes both vehicle self-attention and lane-vehicle attention like SABeR-VAE, but encodes trajectories into an unregularized (uninterpretable) latent space. **(8)** To test the effectiveness of the Koopman operator in SABeR-VAE, we train an ablation model (**Att-LSTM-VAE**) that replaces the Koopman propagation module with a recurrent decoder like Park *et al.* [34].

4.3 Quantitative Evaluation Metrics

We quantitatively evaluate the effectiveness of models on the MAAD dataset using four metrics. **(1)** Area Under Receiver-Operating Characteristic curve (**AUROC**) is calculated by plotting the False-Positive Rate (FPR) and True-Positive Rate (TPR) of a model over several decision thresholds, and computing the area under the curve. A model with greater AUROC performs better, and a perfect classifier has an AUROC of 100%. Though, AUROC is skewed in datasets where there are very few positive labels, like in the field of outlier identification. As such, FPR may be misleadingly low, producing an optimistic AUROC value. We compute **(2)** the Area Under Precision-Recall Curve (AUPR) with the anomalous points being the positive class (**AUPR-Abnormal**) and **(3)** with normal points being positive (**AUPR-Normal**). The AUPR metric adjusts for skewed dataset distributions, and we evaluate model effectiveness of classifying anomalies, and not mis-classifying normal points with AUPR-Abnormal and AUPR-Normal respectively. Finally, we use **(4) FPR @ 95%-TPR** to check the rate of mis-labeling normal points when TPR is high.

4.4 Accuracy Results

Every model was trained for 500 epochs on the training split with a Tesla V100 GPU [25]. Like Wiederer *et al.* , we calculate metrics for each hyperparameter choice on a 20% validation split of the whole test data, and choose to evaluate the best set of hyperparameters for each respective method on the complete test split [42]. More details on training and hyperparameter choices are provided in the supplementary material [10]. Table 1 holds accuracy results of
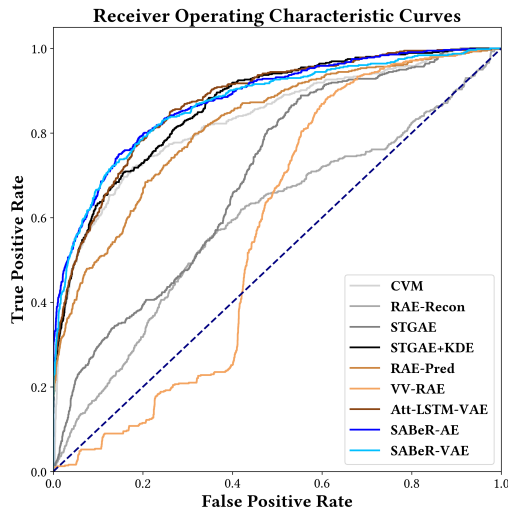
Receiver Operating Characteristic Curves

Figure 4: ROC curves of tested methods.

Table 2: AUROC (↑) of methods by anomaly type.

Anomaly Type	CVM	STGAE-KDE[*]	SABeR-AE	SABeR-VAE
Reeving	**96.7**	94.6	87.9	89.5
Pushing Aside	**91.5**	90.4	88.6	91.3
Right Spreading	87.7	**96.2**	86.7	95.2
Left Spreading	90.9	**96.6**	96.2	95.9
Off-Road	88.7	98.2	98.2	**99.7**
Skidding	96.9	99.7	∼**100.0**	99.8
Wrong-way	63.2	73.2	∼**100.0**	99.3

[*] These results are presented in [42].

baselines, ablations, and the SABeR methods on the test split of the MAAD dataset. Each method, besides CVM, was trained ten separate times with the same hyperparameters, and we report the average and standard deviation of each methods' results over the ten runs. Figure 4 plots the ROC curves for each method.

Amongst baselines, the simple CVM model already performs well as an anomaly detector since its AUROC is only 3% less than that of the STGAE-KDE method. CVM also has no variation of results since it is a deterministic model that is not trained. The LSTM-based RAE-Recon model is unable to effectively distinguish between anomalies and normal scenarios using reconstruction loss, since it does not model vehicle or lane information. While recurrent models encode current timestep features based solely on previous timesteps, temporal convolution methods extract information from the whole trajectory, which helps to predict a more accurate reconstruction. Thus, the convolutional STGAE method drastically improves AUROC and AUPR scores over RAE-Recon.

However, once we incorporate a latent propagation network and predict future timesteps, the RAE-Pred ablation increases AUROC over RAE-Recon by 29% and even outperforms STGAE in the AUPR-Abnormal metric, without even modeling inter-vehicle behaviors. This result hints to the idea that recurrent networks learn to model normal behaviors more accurately with future prediction error, than reconstruction error of observed timesteps, which assists in AD performance. Furthermore, recurrent methods are capable of reaching the same performance as convolutional methods, while

relying only on past data points. Still, RAE-Pred is shown to be unstable as it produces a high variance in results over the ten trained models. This variance was caused by two of the ten runs achieving only 45% AUROC. STGAE also has the highest variance in results among baselines since it is a stochastic method reconstructing a distribution over states, rather than the deterministic CVM and RAE-Recon approaches, but is more stable than RAE-Pred.

We see that adding a vehicle-vehicle self-attention layer in VV-RAE model actually hinders performance, and gives results similar to RAE-Recon. Effectively, the vehicle-vehicle self-attention layer did not learn useful features for the future prediction task, and confused the model generations. This outcome could be a result of a low complexity neural network or a potentially poor choice for masking distance d.

The one-class prediction model STGAE-KDE fits a KDE to the latent space of the STGAE to learn a distribution of normal latent behaviors. As such, this one-class classification approach improves detection rates and training stability over the STGAE such that it outperforms other baselines. However, the fitting process of a KDE to a large dimensional space is a computationally complex and constrictive part the method. With gaussian regularization of the latent space, our SABeR-VAE clusters similar behaviors together and learns a latent distribution without fitting a KDE, which we discuss in 4.5.

Finally, SABeR-AE and SABeR-VAE incorporate a lane-vehicle attention module to capture the effect of the structure of the environment on normal behaviors. We see that SABeR-AE outperforms all methods in AUROC and AUPR-Abnormal with low variance, showcasing the importance of modeling environment structure in this field. SABeR-VAE performs slightly better than STGAE-KDE in AUROC, and significantly increases the AUPR-Abnormal score by 18%. However, the stochasticity of the SABeR-VAE method hinders its reproducibility, and AUROC scores ranged from 84% to 89% over the ten training runs. SABeR-VAE further decreases the average FPR @ 95%-TPR of SABeR-AE by 10%. STGAE-KDE and the two SABeR approaches have similar AUPR-Normal. SABeR-VAE also outperforms Att-LSTM-VAE meaning a recurrent decoder is unnecessary when using the Koopman operator.

We present examples of SABeR-VAE scoring anomalous timesteps in Fig. 3. There, a normal overtaking maneuver was scored very low during the whole trajectory, whereas going off-road or driving in the wrong direction were scored high. We can also see in Fig. 3.b that timesteps where vehicles are acting normally prior to erratic behavior are still correctly scored low. Table 2 holds AUROC by anomaly type for CVM, STGAE-KDE, and SABeR methods. SABeR-VAE improves wrong-way driving AD by 35% over STGAE-KDE, while performing comparably in other metrics. The complete version of Table 2 is provided in the supplementary material [10].

4.5 Latent Space Interpretation

SABeR-VAE is a variational model with a continuous latent space such that observations with similar learned characteristics are clustered closer together in the latent space. In Fig. 5.a, we plot the test-split latent space of one of the SABeR-VAE models evaluated in Table 1. Points are clustered into three distinct regions in the latent space, which we will refer to as "bottom," "middle," and "top" clusters respectively. We see from sampled trajectories that the
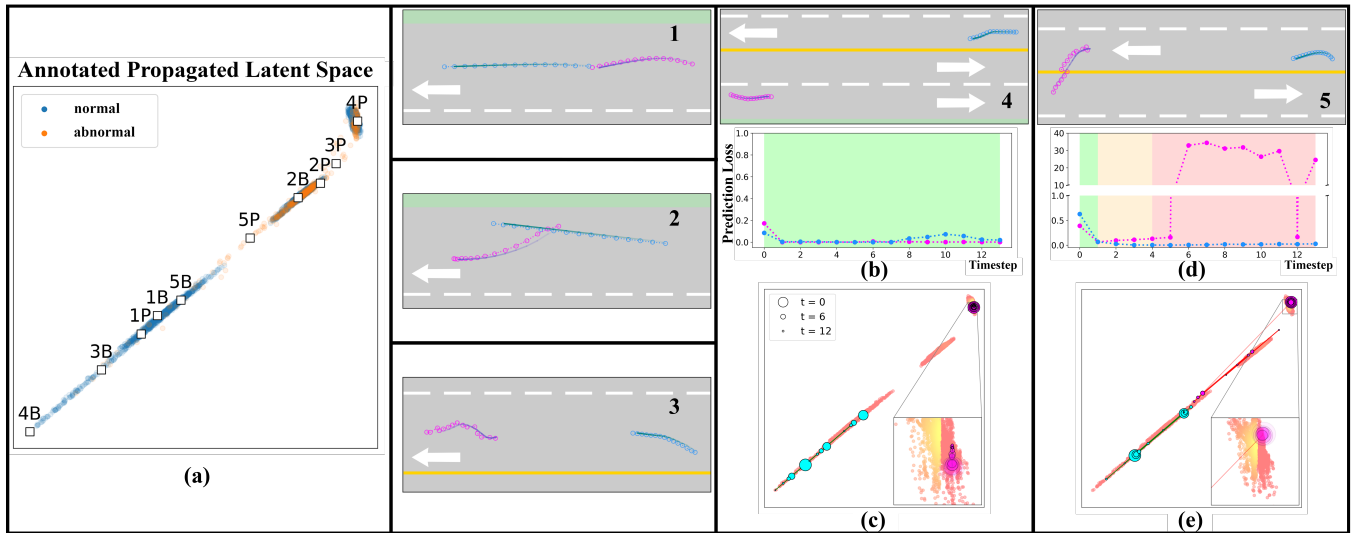
Figure 5: Koopman propagated latent space and corresponding trajectories. (a) We encode every window trajectory in the test-split of the MAAD dataset and plot the 2*D* sampled latent positions of the final timestep of the windows. Blue points correspond to ground truth normal windows while orange are abnormal. **(1-5)** Five scenario windows are encoded into the latent space, and are explicitly annotated in (a.) Each of the five windows has two latent points for the pink and blue vehicles respectively. (e.g., annotations "1B" & "1P" are the latent points of the blue and pink vehicles in road trajectory window (1.) Within the five trajectory windows, solid lines are the ground truth trajectory of the vehicle while open circles are predicted by SABeR-VAE. White arrows denote direction of traffic flow. **(b & d)** The prediction error curves for trajectories 4 and 5 respectively. **(c & e)** The trajectory of latent points for vehicle windows 4 and 5 respectively. A heat map of the original latent space is plotted in orange in the background. Blue and pink circles are the latent trajectories of the blue and pink car through time. The largest circles encode the initial timestep of the window, and they decrease in size as the window progresses.

bottom and middle clusters encode vehicles that travel toward the $-x$ (left) direction in either of the top two lanes of the highway, while the top cluster encodes vehicles traveling to the right in the bottom two lanes. For example, points 1P, 2P, 3B, 4B, and 5B correspond to blue (B) and pink (P) cars that travel to the left in the top lanes. Similarly, the trajectory of the pink car driving to the right in window 4 is encoded to point 4P in the top cluster of the latent space. Vehicles that are also physically close and interacting with each other are encoded closely in the latent space, as shown with latent points corresponding to windows 1 and 2. The middle cluster embeds anomalous scenarios from the top lanes where vehicles are close enough to interact with each other, like window 2.

Furthermore, anomalous, non-interactive trajectories that are poorly predicted are encoded to the outskirts of the primary cluster distributions. For example, the pink cars in trajectories 3 and 5 are driving in the wrong direction. These trajectories have high prediction error as visualized by the little overlap between the predicted open circle positions and ground truth trajectories. As such, those poorly predicted points are encoded in the spaces between the bottom and middle, and middle and top clusters respectively. In contrast, trajectories 1 and 4 have low loss and are encoded to positions within the primary clusters. Thus the latent space has learned a correspondence between permissible lane routings and expected vehicle behavior.

Finally, we visualize the transformation of the latent space over time within one trajectory window to show the interpretable impact of the learned Koopman operator. Figures 5.c and 5.e show the transformation of the latent space as time progresses in trajectory

windows 4 and 5. We can see in Fig. 5.c that the blue and pink latent trajectories stay in the bottom and top clusters respectively, since the vehicles follow the correct direction on the road throughout window 4. Conversely, we see in Fig. 5.e that the pink latent trajectory begins in the top cluster since the pink vehicle in trajectory 5 is in one of the bottom two lanes on the road. But, at timestep 6, the pink car crosses the road divider into the wrong direction lane. Thus, we see a jump in the pink car's latent trajectory in Fig. 5.e from the top cluster to the bottom and middle clusters that correspond to the top two lanes. At the same time, Fig. 5.d has a spike in the prediction loss of the pink vehicle. For the remainder of the trajectory window, the pink car oscillates drastically in the latent space around the middle cluster, since the model expects the vehicle to be traveling to the left. Note, even though the pink vehicle in trajectory 5 is acting abnormally, this does not effect the latent trajectory of the blue vehicle in the same window, since the vehicles are not close enough to impact each other. Overall, the Koopman operator explicitly models this transition from normal to anomalous states in the latent space, in an interpretable manner.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel framework for anomaly detection with an unsupervised recurrent VAE network conditioned on structured environment information and vehicle interactions. We show that modeling this structured information is imperative to having high accuracy over a wide range of anomaly types and study the interpretability of the architecture. Future work includes using raw sensor data for detection and integrating with a vehicle controller.

ACKNOWLEDGMENTS

REFERENCES

[1] Hassan Arbabi and Igor Mezić. 2017. Study of dynamics in post-transient flows using Koopman mode decomposition. *Phys. Rev. Fluids* 2 (Dec 2017), 124402. Issue 12. https://doi.org/10.1103/PhysRevFluids.2.124402

[2] Davide Azzalini, Alberto Castellini, Matteo Luperto, Alessandro Farinelli, and Francesco Amigoni. 2020. Hmms for anomaly detection in autonomous robots. In *International Conference on Autonomous Agents and MultiAgent Systems*. ACM, 105–113.

[3] Kaushik Balakrishnan and Devesh Upadhyay. 2020. Deep adversarial koopman model for reaction-diffusion systems. *arXiv preprint arXiv:2006.05547* (2020).

[4] Kaushik Balakrishnan and Devesh Upadhyay. 2021. Stochastic Adversarial Koopman Model for Dynamical Systems. *CoRR* abs/2109.05095 (2021). arXiv:2109.05095 https://arxiv.org/abs/2109.05095

[5] Umang Bhatt, McKane Andrus, Adrian Weller, and Alice Xiang. 2020. Machine learning explainability for external stakeholders. *arXiv preprint arXiv:2007.05408* (2020).

[6] Raunak P Bhattacharyya, Derek J Phillips, Changliu Liu, Jayesh K Gupta, Katherine Driggs-Campbell, and Mykel J Kochenderfer. 2019. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 789–795.

[7] Daniel Bogdoll, Maximilian Nitsche, and J Marius Zöllner. 2022. Anomaly Detection in Autonomous Driving: A Survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4488–4499.

[8] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2012. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. In *Proceedings of the 29th International Coference on International Conference on Machine Learning* (Edinburgh, Scotland) *(ICML'12)*. Omnipress, Madison, WI, USA, 1881–1888.

[9] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. 2020. MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. In *Proceedings of the Conference on Robot Learning (Proceedings of Machine Learning Research, Vol. 100)*, Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (Eds.). PMLR, 86–99. https://proceedings.mlr.press/v100/chai20a.html

[10] Neeloy Chakraborty, Aamir Hasan, Shuijing Liu, Tianchen Ji, Weihang Liang, D. Livingston McPherson, and Katherine Driggs-Campbell. 2023. Structural Attention-Based Recurrent Variational Autoencoder for Highway Vehicle Anomaly Detection. https://doi.org/10.48550/ARXIV.2301.03634

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[12] Jingyuan Chen, Guanchen Ding, Yuchen Yang, Wenwei Han, Kangmin Xu, Tianyi Gao, Zhe Zhang, Wanping Ouyang, Hao Cai, and Zhenzhong Chen. 2021. Dual-modality vehicle anomaly detection via bilateral trajectory tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4016–4025.

[13] Taesung Choi, Dongkun Lee, Yuchae Jung, and Ho-Jin Choi. 2022. Multivariate Time-series Anomaly Detection using SeqVAE-CNN Hybrid Model. In *2022 International Conference on Information Networking (ICOIN)*. 250–253. https://doi.org/10.1109/ICOIN53446.2022.9687205

[14] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/b618c3210e934362ac261db280128c22-Paper.pdf

[15] Nahuel Costa, Luciano Sánchez, and Inés Couso. 2021. Semi-Supervised Recurrent Variational Autoencoder Approach for Visual Diagnosis of Atrial Fibrillation. *IEEE Access* 9 (2021), 40227–40239. https://doi.org/10.1109/ACCESS.2021.3064854

[16] Oliver De Candido, Maximilian Binder, and Wolfgang Utschick. 2021. An Interpretable Lane Change Detector Algorithm based on Deep Autoencoder Anomaly Detection. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. 516–523. https://doi.org/10.1109/IV48863.2021.9575599

[17] Nachiket Deo, Eric Wolff, and Oscar Beijbom. 2021. Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals. In *5th Annual Conference on Robot Learning*. https://openreview.net/forum?id=hu7b7MPCqiC

[18] Yongqi Dong, Kejia Chen, Yinxuan Peng, and Zhiyuan Ma. 2022. Comparative Study on Supervised versus Semi-supervised Machine Learning for Anomaly Detection of In-vehicle CAN Network. https://doi.org/10.48550/ARXIV.2207.10286

[19] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. 2015. DRAW: A Recurrent Neural Network For Image Generation. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1462–1471. https://proceedings.mlr.press/v37/gregor15.html

[20] Peihua Han, André Listou Ellefsen, Guoyuan Li, Finn Tore Holmeset, and Houxiang Zhang. 2021. Fault Detection With LSTM-Based Variational Autoencoder for Maritime Components. *IEEE Sensors Journal* 21, 19 (2021), 21903–21912. https://doi.org/10.1109/JSEN.2021.3105226

[21] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.

[22] Boris Ivanovic, Karen Leung, Edward Schmerling, and Marco Pavone. 2020. Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *IEEE Robotics and Automation Letters* 6, 2 (2020), 295–302.

[23] Tianchen Ji, Arun Narenthiran Sivakumar, Girish Chowdhary, and Katherine Driggs-Campbell. 2022. Proactive Anomaly Detection for Robot Navigation with Multi-Sensor Fusion. *IEEE Robotics and Automation Letters* 7, 2 (2022), 4975–4982.

[24] Tianchen Ji, Sri Theja Vuppala, Girish Chowdhary, and Katherine Driggs-Campbell. 2021. Multi-Modal Anomaly Detection for Unstructured and Uncertain Environments. In *Conference on Robot Learning*. PMLR, 1443–1455.

[25] Volodymyr Kindratenko, Dawei Mu, Yan Zhan, John Maloney, Sayed Hadi Hashemi, Benjamin Rabe, Ke Xu, Roy Campbell, Jian Peng, and William Gropp. 2020. HAL: Computer System for Scalable Deep Learning. In *Practice and Experience in Advanced Research Computing* (Portland, OR, USA) *(PEARC '20)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/3311790.3396649

[26] D.P. Kingma and M. Welling. 2014. Auto-encoding variational bayes.. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. Machine Learning Group, Universiteit van Amsterdam.

[27] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. 2017. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2165–2174. https://doi.org/10.1109/CVPR.2017.233

[28] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. 2020. Learning Lane Graph Representations for Motion Forecasting. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 541–556.

[29] Shuijing Liu, Peixin Chang, Haonan Chen, Neeloy Chakraborty, and Katherine Driggs-Campbell. 2022. Learning to Navigate Intersections with Unsupervised Driver Trait Inference. In *2022 International Conference on Robotics and Automation (ICRA)*. 3576–3582. https://doi.org/10.1109/ICRA46639.2022.9811635

[30] Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. 2021. HDMapGen: A Hierarchical Graph Generative Model of High Definition Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4227–4236.

[31] Jeremy Morton, Freddie D. Witherden, and Mykel J. Kochenderfer. 2019. Deep Variational Koopman Models: Inferring Koopman Observations for Uncertainty-Aware Dynamics Modeling and Control. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 3173–3179. https://doi.org/10.24963/ijcai.2019/440

[32] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.

[33] Daehyung Park, Zackory Erickson, Tapomayukh Bhattacharjee, and Charles C Kemp. 2016. Multimodal execution monitoring for anomaly detection during robot manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 407–414.

[34] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.

[35] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. 2020. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*. Springer, 683–700.

[36] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. 2020. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters* 5, 2 (2020), 1696–1703.

[37] Jonas Herskind Sejr and Anna Schneider-Kamp. 2021. Explainable outlier detection: What, for Whom and Why? *Machine Learning with Applications* 6 (2021), 100172. https://doi.org/10.1016/j.mlwa.2021.100172

[38] Meet Shah, Zhi ling Huang, Ankita Gajanan Laddha, Matthew Langford, Blake Barber, Sidney Zhang, Carlos Vallespi-Gonzalez, and Raquel Urtasun. 2020. Li-RaNet: End-to-End Trajectory Prediction using Spatio-Temporal Radar Fusion. In *CoRL*.

[39] John Sipple and Abdou Youssef. 2022. A general-purpose method for applying Explainable AI for Anomaly Detection. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 162–174.

[40] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. https://doi.org/10.48550/ARXIV.1706.03762

[42] Julian Wiederer, Arij Bouazizi, Marco Troina, Ulrich Kressel, and Vasileios Belagiannis. 2022. Anomaly Detection in Multi-Agent Trajectories for Automated Driving. In *Conference on Robot Learning*. PMLR, 1223–1233.

[43] Chule Yang, Alessandro Renzaglia, Anshul Paigwar, Christian Laugier, and Danwei Wang. 2019. Driving Behavior Assessment and Anomaly Detection for Intelligent Vehicles. In *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. 524–529. https://doi.org/10.1109/CIS-RAM47153.2019.9095790

[44] Yu Yao, Xizi Wang, Mingze Xu, Zelin Pu, Yuchen Wang, Ella Atkins, and David Crandall. 2022. DoTA: unsupervised detection of traffic anomaly in driving videos. *IEEE transactions on pattern analysis and machine intelligence* (2022).

[45] Yu Yao, Mingze Xu, Yuchen Wang, David J Crandall, and Ella M Atkins. 2019. Unsupervised traffic accident detection in first-person videos. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 273–280.

[46] Jun Yong Yoon, Jinseop Jeong, and Woosuk Sung. 2022. Design and Implementation of HD Mapping, Vehicle Control, and V2I Communication for Robo-Taxi Services. *Sensors* 22, 18 (2022). https://doi.org/10.3390/s22187049

[47] Xinhai Zhang, Jianbo Tao, Kaige Tan, Martin Törngren, José Manuel Gaspar Sánchez, Muhammad Rusyadi Ramli, Xin Tao, Magnus Gyllenhammar, Franz Wotawa, Naveen Mohan, Mihai Nica, and Hermann Felbinger. 2021. Finding Critical Scenarios for Automated Driving Systems: A Systematic Literature Review. *ArXiv* abs/2110.08664 (2021).