Detecting Socially Abnormal Highway Driving Behaviors via Recurrent Graph Attention Networks

Yue Hu
Vanderbilt University
Nashville, US
yue.hu@vanderbilt.edu

Yuhang Zhang
Vanderbilt University
Nashville, US
yuhang.zhang.1@vanderbilt.edu

Yanbing Wang
Vanderbilt University
Nashville, US
yanbing.wang@vanderbilt.edu

Daniel Work
Vanderbilt University
Nashville, US
dan.work@vanderbilt.edu

ABSTRACT

With the rapid development of Internet of Things technologies, the next generation traffic monitoring infrastructures are connected via the web, to aid traffic data collection and intelligent traffic management. One of the most important tasks in traffic is anomaly detection, since abnormal drivers can reduce traffic efficiency and cause safety issues. This work focuses on detecting abnormal driving behaviors from trajectories produced by highway video surveillance systems. Most of the current abnormal driving behavior detection methods focus on a limited category of abnormal behaviors that deal with a single vehicle without considering vehicular interactions. In this work, we consider the problem of detecting a variety of socially abnormal driving behaviors, i.e., behaviors that do not conform to the behavior of other nearby drivers. This task is complicated by the variety of vehicular interactions and the spatial-temporal varying nature of highway traffic. To solve this problem, we propose an autoencoder with a Recurrent Graph Attention Network that can capture the highway driving behaviors contextualized on the surrounding cars, and detect anomalies that deviate from learned patterns. Our model is scalable to large freeways with thousands of cars. Experiments on data generated from traffic simulation software show that our model is the only one that can spot the exact vehicle conducting socially abnormal behaviors, among the state-of-the-art anomaly detection models. We further show the performance on real world HighD traffic dataset, where our model detects vehicles that violate the local driving norms.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; **Data mining**.

KEYWORDS

Intelligent transportation; Spatial-temporal learning; Graph neural networks; Anomaly detection

1 INTRODUCTION

1.1 Motivation and challenges

Nowadays, the development of Internet of Things (IoT) technologies has greatly advanced intelligent monitoring and management of urban transportation systems. Sensor networks including highway surveillance cameras and radar detectors, combined with Web of Things (WoT) technologies, have enabled transportation authorities to intelligently monitor traffic systems at scales and resolutions previously out of reach [17, 43, 44]. One of the emerging tasks in intelligent traffic management is anomaly detection, since abnormal drivers could have adversarial impact on the smoothness of traffic stream, or even pose safety concerns. Yet it is infeasible for human operators to manually inspect and analyze all of this data, given the now massive amount of data these systems can generate. Consequently, there is need to spot the anomalies from terabytes of data and highlight the scenes that need further human inspection. In this work, we tackle the task of detecting abnormal driving behaviors, from trajectories produced by IoT highway video surveillance systems.

The existing approaches to vehicular anomaly detection mainly fall into two categories. The first set of approaches [4, 9, 26, 41, 52] focus on detecting severe events that cause vehicles to stop, and turn the problem into detecting stalled cars via computer vision from surveillance videos. The second set of approaches [3, 16, 31, 34] focus on single-car abnormal driving behaviors, such as speeding and abrupt braking. Methods ranging from thresholding [8, 15] to machine learning [31, 32, 34] are applied on data obtained from a single car, e.g. from on-board sensors and GPS devices.

However, the above approaches only cover a subset of abnormal driving behaviors and treat cars in isolation from each other. Vehicles constantly interact with their surroundings, and traffic context

is needed for anomaly detection. For example, a car at a constant speed of 50 mph might be perfectly normal, yet a car at 50 mph on the inner most lane of highway is blocking all the other cars driving at 65 mph or above, and should be considered an anomaly on highway. As another example, abrupt braking might be considered abnormal, but if the vehicle is braking because its front car is stalled, then we should detect the front car as abnormal, whereas the braking car is doing what is expected. Our task is to detect such *socially abnormal behaviors* that do not conform to the commonly accepted and observed social norms, by developing a contextual understanding of vehicle interactions.

Moreover, most of the existing methods are rule-based [8, 15] or supervised-learning method [16, 34], and can only detect predefined types of anomalies such as stalled and speeding cars. Yet vehicles can behave anomalously in unexpected ways, and building a comprehensive set of rules that includes every possible occasion could be hard. Our goal is to build a model that can identify a variety of anomalies with unsupervised learning.

Graph neural networks (GNN) have seen rapid development in recent years [23, 46, 49, 50], showing great advantage in modeling the complex relationships in graph data. By representing vehicles as nodes and their relationships as edges, interactions with neighbors can be modeled via a GNN. Two challenges exist in developing a GNN for detecting anomalous driving behaviors. First, given the spatial-temporally varying nature of trajectories, we need to deal with dynamic graphs. This is a nontrivial problem as compared to anomaly detection on only a static graph, or only considering time-varying signals. Second, we need to take stochasticity into consideration, which is intrinsic in driving behaviors. Under a particular context, there could be a range of acceptable behaviors that should all be considered normal. For example, a car can conduct lane changing from time to time, or have some variation in speed. A deterministic model that fails to capture such normal stochasticity would mislabel every lane-changing car as an anomaly.

1.2 Our approach

To detect socially abnormal behaviors on large scale trajectory data while addressing the aforementioned challenges, we develop a model for *Detecting Socially Abnormal Behaviors* **(DSAB)** in highway driving via Recurrent Graph Attention Autoencoder. Graph attention networks combined with recurrent neural networks is used to capture the spatial-temporal pattern of vehicle trajectories, while dynamically taking each vehicle's neighbors into consideration based on vehicle states. To facilitate scalability while capturing anomalous driving behaviors occurring over longer periods of time, we sample the trajectories over a relatively long-horizon and coarse-grid time window. We further use a sparse graph where the vehicles are only connected to close neighbors to reduce computation. An autoencoder structure is used for anomaly detection, which can encode and decode normal data well. To address the stochasticity in driving behaviors, we reconstruct the probabilistic distribution of the trajectories in decoding process. Samples with small reconstruction probabilities are marked as anomalies.

With these designs, our model can detect the socialy abnormal driving behaviors, and is scalable to thousands of cars over 5 miles of highway. We show the effectiveness of our model on both simulation and real-world data. First, we generate large-scale trajectory

data with ground-truth anomaly labels, via a microscopic traffic simulator, to quantitatively evaluate the performance. Compared with the existing state-of-the-art methods, our model is the only one that can detect the exact vehicle with abnormal behavior, whereas the rest models can only detect anomalous scenes We also apply our method on the real-world highD trajectory dataset [24], where our model detects vehicles that violate the local driving norms.

To summarize, the contribution of our works is as follows:

- We propose a new problem of detecting the exact anomalous vehicles that violate the social interaction norms in highway driving, and develop a model to solve it, achieving state of the art performance.
- We develop a DSAB model based on Recurrent Graph Attention Networks. It well captures the spatial-temporal trajectory dynamics, while considering both the vehicular interactions and the stocasticity in driving behaviors.
- We conduct extensive experiments on both simulation and real-world data sets, and show the ability of our model to scale to large highway monitoring systems with thousands of vehicles, and detect a variety of abnormal behaviors.

2 RELATED WORK

Trajectory Modeling. The majority of trajectory modeling work focuses on the prediction of future trajectories for humans or vehicles, and in this line, modeling the interaction between the agents is gaining interest. To aggregate information across agents, a pooling mechanism is used in the Social LSTM [2] and the Social GAN [18], while an attention mechanism is used in SoPhie [36], and scene context fusion via convolutional neural network is used in Desire [25]. With recent development of graph convolutional networks, works including [21, 33, 47] model the agents as nodes and their relationships as edges, and develop spatio-temporal GNN to learn the dynamics.

Anomaly detection. Anomaly detection has been an important task in transportation. Unexpected autonomous driving condition detection is studied in [39, 40]. Extreme event detection in urban traffic is studied in [19, 20]. Our work detects anomalies on graphs, and a comprehensive survey on graph anomaly detection can be found in [30]. Common detection methods on graphs include autoencoder-based methods [5, 14], generative adversarial learning [13], and contrastive learning [53]. However, most of the existing works are on static graphs, whereas we need to deal with dynamic graphs, adding the temporal dimension. The existing methods for dynamic graph anomaly detection have different problem settings compared to ours. Specifically, NetWalk [54] and TADDY [29] deal with unattributed graphs with no node or edge features; AddGraph [55] and StrGNN [7] detect anomalous edges.

The most relevant work with ours is STGAE [51], where a spatio-temporal graph autoencoder is combined with kernel density estimation (KDE) to detect abnormal driving behaviors. Yet while STGAE works well in experiments with only two vehicles, the time complexity of KDE is too high for detection in large numbers of vehicles. Moreover, STGAE tackles a different task of detecting the existence of anomalies among several vehicles over a stretch of road over a specific time, which we term as *abnormal scene detection*.

Compared with our task of detecting the specific abnormal vehicles, abnormal scene detection is an easier task, since an abnormal vehicle can have subsequent influence on its neighbor vehicles. For example, a slow or stalled car can cause its following cars to break abruptly or change lanes, which could be detected as abnormal, but should have been considered normal when looking for the root cause. We demonstrate in our experiments that while most baseline methods can work well on abnormal scene detection, they achieve poor performance on abnormal vehicle detection.

3 METHOD

In this section, we formulate the problem mathematically, then describe the proposed DSAB model for anomaly detection on highways. In an overview, we first construct the vehicle trajectories as a spatial-temporal dynamic graph. Then we build an autoencoder with an encoder to compress the vehicle time series in low-dimensional vectors, and a decoder to reconstruct the probabilistic distribution of original input trajectories. At test time, we use the reconstruction probability as a measurement of anomalous behavior. The overview of the model is shown in Fig 1.

We further make the following considerations to best accommodate large scale anomaly detection on highway driving. Unlike most works [2, 25, 33] that works on short-term fine-grained trajectories, e.g., 3-10 Hz over 1-5s, we choose coarser-grained sampling with larger time window. In addition to reducing the amount of data needed to be processed, it also captures the anomalous driving behaviors that are expected to occur over longer periods of time. E.g., a vehicle that is speeding or tailgating will likely persist for more than a few seconds. While computationally advantageous, sampling at a more coarse timescale requires a different model to be used for lane changing. Specifically, the continuous bi-variate Gaussian distribution, which is commonly used in trajectory modeling [2, 33, 47], is no longer suitable to capture a discrete lane-changing motion. Instead, we model longitudinal motions as Gaussian distribution, and lateral lane location as categorical distribution.

3.1 Problem formulation

Our model input is the observations of a set of N vehicles on a highway over time window $\mathcal{T}$, where N can vary for different time windows. For each vehicle i at time $t \in \mathcal{T}$, the observation $\mathbf{o}_t^i = [x_t^i, y_t^i, l_t^i, v_t^i, a_t^i]$ includes longitudinal position x_t^i, lateral position y_t^i, driving lane id l_t^i, longitudinal speed v_t^i, and longitudinal acceleration a_t^i.

Given the vehicle observations $\mathbf{o}_t^i, \forall t \in \mathcal{T}$, our major goal is to detect the *vehicles* that have abnormal behavior during the time window $\mathcal{T}$. Additionally, we also report the performance of detecting abnormal *scenes* - dividing the entire highway into short stretches of length δ_s, a scene contains all cars on a stretch $\mathcal{S}$ during $\mathcal{T}$, and is labeled as abnormal if it contains any abnormal cars.

3.2 Graph construction for vehicle trajectories

We first construct a dynamic graph for the vehicle trajectories during time window $\mathcal{T}$, denoted as $G(\mathcal{T}) = \{(\mathcal{V}, \mathcal{E}_1), (\mathcal{V}, \mathcal{E}_2), \ldots, (\mathcal{V}, \mathcal{E}_T)\}$ for discrete time steps $t \in \mathcal{T} = \{1, 2, \ldots, T\}$. The node set $\mathcal{V}$ includes all the vehicles on highway during the time window $\mathcal{T}$. The number of nodes $N = |\mathcal{V}|$ is fixed for the graph of a specific
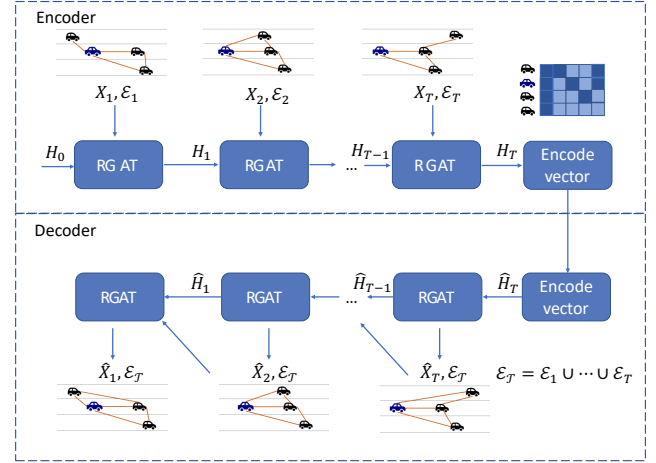

Figure 1: DSAB model overview. We construct a spatial-temporal dynamic graph to represent vehicles. An encoder compresses the dynamic graph into a low-dimensional encode vector, and a decoder reconstructs the input vehicle states, based on Recurrent Graph Attention Network (RGAT). In the encoder, $X_t, \mathcal{E}_t$ denotes the node feature matrix and edge set respectively. In the decoder, $\hat{X}_t$ is the node reconstruction matrix, $\mathcal{E}_{\mathcal{T}}$ is the union of all input edge sets.

time window, but can vary for graphs of different time windows[1]. At each time step t, the vehicle observations are summarized into a raw observation matrix $\mathbf{O}_t \in \mathbb{R}^{N \times d}$. $\mathbf{O}_t$ includes the observation $\mathbf{o}_t^i = [x_t^i, y_t^i, l_t^i, v_t^i, a_t^i]$ for all vehicles $V_i \in \mathcal{V}$. There is an edge $e_t^{ij} \in \mathcal{E}_t$ if vehicle i and j are close at time t, that is, the two vehicles are less than δ_x feet apart longitudinally, and less than δ_l lanes apart laterally. This is based on the consideration that in reality, the vehicles are most influenced by their close neighborhood vehicles. Thus we limit the neighbor range to reduce storage and computational requirements.

3.3 Encoder

In this subsection, we explain how we use *Recurrent Graph Attention Network* (RGAT) to encode the spatial-temporal vehicle trajectories. We adopt an RNN which has proven to work well on time series data. Furthermore, to consider the dynamic influence of each vehicle's neighbors, we integrate the graph attention network with RNN.

Specifically, we adopt the GRU [10, 11] variant of RNN, which is capable of learning long range dependencies in time via a gating mechanism. Furthermore, similar to the practice of [27, 38], we substitute the matrix multiplications in the original GRU with graph convolutions denoted as $*_{\mathcal{G}}$, to operate on the input and hidden states and capture neighborhood interactions:

$$
\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_{xz} *_{\mathcal{G}} \mathbf{X}_t + \mathbf{W}_{hz} *_{\mathcal{G}} \mathbf{H}_{t-1} + \mathbf{b}_z), \\
\mathbf{r}_t &= \sigma(\mathbf{W}_{xr} *_{\mathcal{G}} \mathbf{X}_t + \mathbf{W}_{hr} *_{\mathcal{G}} \mathbf{H}_{t-1} + \mathbf{b}_r), \\
\tilde{\mathbf{H}}_t &= \tanh(\mathbf{W}_{xh} *_{\mathcal{G}} \mathbf{X}_t + \mathbf{W}_{hh} *_{\mathcal{G}} (\mathbf{r}_t \odot \mathbf{H}_{t-1} + \mathbf{b}_h)), \\
\mathbf{H}_t &= \mathbf{z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{H}}_t,
\end{aligned}
\tag{1}
$$

---

[1]We explain in implementation details in Section 4.3 how we deal with vehicles that are not present in the highway during the entire time window.

where $\mathbf{H}_t \in \mathbb{R}^{N \times d_h}$ is the hidden state, $\mathbf{z}_t \in \mathbb{R}^{N \times d_h}$ and $\mathbf{r}_t \in \mathbb{R}^{N \times d_h}$ are the update gate and the reset gate respectively, with hidden dimension d_h. The weights $\mathbf{W}_{xz}, \mathbf{W}_{hz}, \mathbf{W}_{xr}, \mathbf{W}_{hr}, \mathbf{W}_{xh}, \mathbf{W}_{hh}$ and biases $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h$ are trainable parameters. σ is the sigmoid function, and $\odot$ is element-wise multiplication. $\mathbf{X}_t \in \mathbb{R}^{N \times d_f}$ is the input at time step t, derived from the raw observation input $\mathbf{O}_t$, and consists of numerical observations of position, speed, and acceleration $[x_t, y_t, v_t, a_t] \in \mathbb{R}^{N \times 4}$, concatenated with lane embeddings $\mathbf{h}_{\text{lane}_t} \in \mathbb{R}^{N \times d_l}$ for categorical observations of driving lane IDs $\mathbf{l}_t$. Thus, the input feature dimension $d_f = 4 + d_l$. Each driving lane ID is mapped to a corresponding embedding vector with dimension d_l. The entries of the lane embedding vectors are initialized at random and learned during training.

In terms of the graph convolution operator $*_{\mathcal{G}}$, instead of the Chebyshev spectral graph convolutional operator [12] adopted by [38], which uses pre-defined edge weights, we adopt graph convolutions based on graph attention mechanism [6, 46], which determines the relevance of the vehicle's neighbors dynamically based on the vehicle states. Next we introduce the graph attention based convolution $\mathbf{W} *_{\mathcal{G}} \mathbf{X}_t$ operated on the input observation matrix $\mathbf{X}_t$, whereas $\mathbf{W} *_{\mathcal{G}} \mathbf{H}_t$ follows the same process on hidden states $\mathbf{H}_t$ with corresponding weights. Denoting $\mathbf{x}_t^i \in \mathbb{R}^{d_f}$ as the entry for vehicle i in $\mathbf{X}_t$, $\mathcal{N}_t^i$ as the set of neighbors of node i at t, $\mathbf{W} *_{\mathcal{G}} \mathbf{X}_t$ works as follows:

$$\mathbf{x}_t'^i = \mathbf{W} *_{\mathcal{G}} \mathbf{X}_t := \alpha_{i,i} \mathbf{W} \mathbf{x}_t^i + \sum_{j \in \mathcal{N}_t^i} \alpha_{i,j} \mathbf{W} \mathbf{x}_t^j, \qquad (2)$$

where $\mathbf{x}_t'^i \in \mathbb{R}^{d_h}$ is the output node embedding, $\mathbf{W} \in \mathbb{R}^{d_h \times d_f}$ is the weight matrix. The term $\alpha_{i,j}$ is the attention score calculated as:

$$e_{i,j} = \left(\mathbf{a}^\top \text{LeakyReLU} \left([\mathbf{W} \mathbf{x}_t^i \parallel \mathbf{W} \mathbf{x}_t^j] \right) \right),$$
$$\alpha_{i,j} = \text{softmax}(e_{i,j}) = \frac{\exp(e_{i,j})}{\sum_{k \in \mathcal{N}_t^i} \exp(e_{i,k})}, \qquad (3)$$

where $\parallel$ is the concatenate operator, and $\mathbf{a} \in \mathbb{R}^{2d_h}$ is a trainable weight vector. In (2) and (3), first every node input feature goes through a linear transformation parameterized by $\mathbf{W}$. Then, the attention coefficients $e_{i,j}$ are calculated by concatenating the transformed node features, followed by a non-linear activation (LeakyReLU), and a linear transformation with parameter $\mathbf{a}$. Then, for each node, the attention coefficients of all its neighbors are normalized by a softmax operator to reach the attention scores $\alpha_{i,j}$. Finally, the output node embedding is calculated as the linear combination of its neighbor transformed feature vectors, weighted by the attention scores.

Compared with works [33, 51] that pre-define edge weights as a function of physical distances, the attention mechanism we use has more expressive power, and can comprehensively determine the neighbor relevance based on observation information. Furthermore, the real influence of vehicles in front of and behind an ego car is asymmetric (e.g., you must slow down immediately for slow-moving cars in front of you, but not for slow cars behind you), and the attention formulation in (3) can achieve such asymmetry. In comparison, when using a function of distance as edge weights [33, 51], the same importance is assigned to cars that are close to a vehicle, regardless if they are in front or behind the ego vehicle.

Moreover, *multi-head attention* is used to attend to different aspects of the neighborhood information, similar to [45, 46]. Specifically, K independent heads is used following (2), and the final results of the heads are averaged:

$$\mathbf{x}_t'^i = \frac{1}{K} \sum_{k=1}^{K} \left(\alpha_{i,i}^k \mathbf{W}^k \mathbf{x}_t^i + \sum_{j \in \mathcal{N}_t^i} \alpha_{i,j}^k \mathbf{W}^k \mathbf{x}_t^j \right). \qquad (4)$$

The hidden states for all vehicles are updated at every time step by (1), (2) and (3). The final hidden state $\mathbf{H}_T$ is used as the encoded vector embedding.

3.4 Decoder

The decoder works in the same way as the encoder, using the RGAT structure. To avoid the computation burden of re-calculating the edge set thresholded by the reconstructed vehicle position at every step, we use the union of all edge sets at encoding time steps, $\mathcal{E}_{\mathcal{T}} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_T$. The edge set only limits the range of neighbors each node attends to, while the importance of neighbors are calculated dynamically with attention mechanism. The decoded hidden state $\hat{\mathbf{H}}_t \in \mathbb{R}^{N \times d_h}$ further goes through a fully connected layer to produce the output vehicle state reconstruction $\hat{\mathbf{X}}_t$. The details of $\hat{\mathbf{X}}_t$ will be explained in Section 3.5. Then, the output $\hat{\mathbf{X}}_t$ is used as input for the next recurrent step. We decode the time series from time T backwards, since the encoder vector is most relevant to the states at time T, which is most recently encoded. The initial input into the decoder GAT-RNN is the vehicle states at time T.

3.5 Loss function

For each vehicle i at time t, we assume its longitudinal position x_t^i, speed v_t^i, and acceleration a_t^i each follows an univariate Gaussian distribution. That is, $x_t^i \sim \mathcal{N}\left(\mu_{x_t^i}, \sigma_{x_t^i}\right)$, $v_t^i \sim \mathcal{N}\left(\mu_{v_t^i}, \sigma_{v_t^i}\right)$, $a_t^i \sim \mathcal{N}\left(\mu_{a_t^i}, \sigma_{a_t^i}\right)$. Further, we assume the lateral lane position l_t^i is a discrete choice among L lanes, with an underlying categorical distribution $\left\{p_{1_t^i}, \ldots, p_{L_t^i}\right\}$. The Gaussian distribution and categorical distribution parameters are estimated by decoder output $\hat{\mathbf{X}}$. That is, $\hat{\mathbf{x}}_t^i = \left[\hat{\mu}_{x_t^i}, \hat{\sigma}_{x_t^i}, \hat{\mu}_{v_t^i}, \hat{\sigma}_{v_t^i}, \hat{\mu}_{a_t^i}, \hat{\sigma}_{a_t^i}, \hat{p}_{1_t^i}, \ldots, \hat{p}_{L_t^i} \right]$ estimates the mean and variance of position, speed, and acceleration, as well as the probability of being in each lane.

Denoting the estimated probability density function of position, speed, and acceleration as $q\left(x_t^i | \hat{\mu}_{x_t^i}, \hat{\sigma}_{x_t^i}\right)$, $q\left(v_t^i | \hat{\mu}_{v_t^i}, \hat{\sigma}_{v_t^i}\right)$, and $q\left(a_t^i | \hat{\mu}_{a_t^i}, \hat{\sigma}_{a_t^i}\right)$ respectively, we aim to minimize the negative log-likelihoods as follows:

$$\begin{aligned}
\mathcal{L}_{x_t^i} &= -\log\left(q\left(x_t^i | \hat{\mu}_{x_t^i}, \hat{\sigma}_{x_t^i}\right) \right), \\
\mathcal{L}_{v_t^i} &= -\log\left(q\left(v_t^i | \hat{\mu}_{v_t^i}, \hat{\sigma}_{v_t^i}\right) \right), \\
\mathcal{L}_{a_t^i} &= -\log\left(q\left(a_t^i | \hat{\mu}_{a_t^i}, \hat{\sigma}_{a_t^i}\right) \right).
\end{aligned} \qquad (5)$$

As for lane classification, we aim to minimize the cross entropy loss as follows:

$$\mathcal{L}_{l_t^i} = -\sum_{l=1}^{L} \mathbb{1}_{l_t^i} \log\left(\hat{p}_{l_t^i}\right), \qquad (6)$$

where $\mathbb{1}_{l_t^i} = 1$ if vehicle i is in lane l at time t and 0 otherwise. The final loss is a weighted sum of the negative log-likelihood losses and the cross entropy loss across all agents and all times:

$$\mathcal{L}_t^i = \lambda_x \mathcal{L}_{x_t^i} + \lambda_v \mathcal{L}_{v_t^i} + \lambda_a \mathcal{L}_{a_t^i} + \lambda_l \mathcal{L}_{l_t^i},$$

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=1}^{T} \mathcal{L}_t^i, \qquad (7)$$

The weights are set as $\lambda_x = 1$, $\lambda_v = 1$, $\lambda_a = 2$ and $\lambda_l = 2$ empirically.

3.6 Anomaly detection

For abnormal vehicle detection, the anomaly score for vehicle i over time window $\mathcal{T}$ is calculated by averaging the loss over all time steps:

$$\alpha_{\mathcal{T}}^i = \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_t^i, \qquad (8)$$

where T is the length of time window $\mathcal{T}$. For abnormal scene detection, we aggregate the loss of all the vehicles that appear in the stretch $\mathcal{S}$ during $\mathcal{T}$ as the stretch anomaly score:

$$\alpha_{\mathcal{T}}^{\mathcal{S}} = \max(\mathcal{L}_t^i), \forall (i, t) \text{ that } x_t^i \in \mathcal{S}. \qquad (9)$$

The maximum aggregation is used instead of the mean for abnormal scene detection, so that the score is more sensitive to the existence of anomalies, and not averaged out by normal vehicles. Appendix C conducts a detailed comparison between maximizing and averaging.

4 EXPERIMENTS

In this section, we use two data sources to evaluate the performance of our method. First, simulation data is used to quantitatively compare the performance of our method with the baselines, where we have the ground truth anomaly labels. Second, highD dataset [24] is used to qualitatively show our method works on real-world trajectories. The code is publicly available on GitHub[2].

4.1 Datasets

The detailed information of the two datasets is as follows.

4.1.1 Simulation data. TransModeler is a microscopic traffic simulator that generates vehicle trajectories mimicking human driving behavior and interactions. In this work, we generate a set of recordings at 1Hz on a 5-mile stretch of a 4-lane highway. The recordings have different traffic flows and vehicle type distribution to include different traffic conditions and abnormal scenarios. Specifically, we include the following scenarios:

- Normal traffic. A standard car following model, *Modified General Motors* [1] is used, which has been demonstrated to correlate well with field traffic data. The desired speed of the vehicles follows a typical distribution found in real-world traffic, with around 5% speeding vehicles and 5% slow vehicles. We include both free flow and congested conditions with varying traffic demands.

- Speeding. The abnormal speeding cars drive at least 15 mph faster than the other vehicles when it is possible to do so.
- Slow. The abnormal slow cars drive at least 15 mph slower than the other vehicles when it is possible to do so.
- Tailgating. The tailgating vehicle's headway is less than 0.5s from the lead car. We simulate tailgating cars via a *Constant Time Gap* car-following model, where drivers can keep a constant desired headway from the leading vehicle [48].
- Stalled car. A vehicle randomly stops on the road for a period.
- Comprehensive scenario. We include all the above abnormal scenarios, i.e., speeding, slow, tailgating and stalled vehicles are all present in a single scenario, to comprehensively evaluate model detection performance.

We include both common anomalies (e.g., speeding, tailgating and stalled vehicles [3, 4, 31, 34]) as well as those that are not commonly studied (e.g., slow driving). Slow driving can create moving bottlenecks with adverse impacts on traffic, but are difficult to identify using existing single-vehicle approaches, demonstrating the importance of developing an interaction-based detector.

The detailed experimental settings and distributions can be found in the Appendix. We adjust the percentages in each abnormal scenario, so that the abnormal behaving cars consist of around 3%-5% of total cars, randomly distributed on the road. The actual anomaly rate has some variation across different recordings because of simulation randomness. A total of 180 min of normal traffic is used for training. The recordings of speeding, slow, tailgating, stalled car and comprehensive scenarios are used for testing, each lasting 10 min. The training data has 7,886 cars in total, producing 920,311 trajectories when segmented into 15s windows at 1s stride. The testing sets together have a total of 5,067 cars and 630,718 trajectories. Detailed statistics can be found in the Appendix.

We note that our training dataset is not perfectly clean, but contains a small portion of abnormal data (e.g., speeding and slow cars). This is intentionally done to simulate the real-world situation where we may not have a clean labeled training set that is known to be free of anomalous vehicles. Experiments excluding training anomalies can be found in Appendix D, which shows similar results.

4.1.2 Real-world data. The HighD dataset [24] contains high accuracy vehicle trajectories extracted from video recordings captured via unmanned aerial vehicles over various stretches of German highways, each stretching approximately 1300 feet long. We select a total of approximately 350 minutes of recordings on three-lane highways as our training data, which covers both light and heavy traffic conditions, with traffic flow varying from 1200 to 3600 vehicles/lane/hour. Then, we test on an unseen 15-min recording with a flow of 2300 vehicles/lane/hour. The training data has 42,106 cars in total, producing 459,187 trajectories when segmented into 10s windows at 1s stride. The testing sets together has 1,795 cars and 21,840 trajectories. For the HighD data, we do not have the ground truth anomaly labels. Thus, we qualitatively examine the top anomalies detected by our model.

4.2 Baselines and Metrics

4.2.1 Baselines. We compare our method with simple heuristic methods as well as state-of-the-art methods for anomaly detection

---

on trajectory data. The baselines include: *i*) *Linear temporal interpolation* (LTI) implemented by [51] that uses a linear interpolation between the first and last position of the vehicle to reconstruct the trajectory; *ii*) *Constant Velocity Model* (CVM) [37] that assumes constant speed as recorded at the first observation time step to reconstruct the trajectory; *iii*) *Robust tensor Recovery* (RTR) [20] model that captures spatial-temporal correlations via low-rank tensor decomposition, and detects sparse outliers that deviates from the normal patterns; *iv*) *Seq2Seq model* [42] that encodes and decodes the time series with two LSTM networks, and uses *Mean Square Error* (MSE) as the reconstruction loss; *v*) *Spatio-temporal graph auto-encoder* (STGAE) [51] that uses convolutional networks temporally and graph convolutional networks spatially, and uses bi-variate Gaussian reconstruction error to build autoencoder to derive anomaly score[3]; *vi*) DSAB-biv, which is the variant of our model, that uses the same RGAT structure, but uses the bi-variate Gaussian loss as in [33, 51]. Each of the baseline methods produces an anomaly score for the vehicles in each time step within a time window, and the same process is used following Eq (8) and (9) to calculate the vehicle and scene anomaly scores.

Out of all the baselines, Seq2Seq, RTR, CVM and LTI consider only each vehicle's own trajectory, and STGAE considers the relationships between vehicles. The heuristic models CVM and LTI are able to detect non-free-flow scenarios where the vehicle speed changes dramatically. The tensor PCA can capture linear correlations among trajectories, and the neural-network methods Seq2seq and STGAE can further capture non-linear patterns.

4.2.2 Metrics. We follow the practice used in the anomaly detection works [13, 28, 35], and include the following standard metrics: *i*) ROC-AUC score, which is widely used for anomaly detection. *ii*) Average precision, which summarizes the precision-recall curve into a single value. *iii*) Precision@k. In settings where one is only able to e.g., manually verify a fixed number of anomalies, Precision@k measures the relevance of the samples we check.

4.3 Implementation details

The model setting for DSAB is as follows. For simulated data: For graph construction, the distance threshold δ_x is 0.1 miles, and the lane threshold $\delta_l = 1$, i.e., vehicles only attend to its own and immediate neighboring lanes within 0.1 miles upstream and downstream. The window size T is 15s, and sampling interval is 1s. The model hidden size $d_h = 5$, the number of attention heads $K = 3$. In training, the model is trained for 500 epochs, with a batch size of 64. The initial learning rate is 0.05, and decreases by half every 50 epochs. Gradient clipping is used to avoid exploding gradients, with max norm of 1. For abnormal scene detection, the stretch length δ_s is set to 0.15 miles. For the highD data: The window size T is 10s. The distance threshold δ_x is 0.2 mile. The batch size is set to 256. All other configurations are the same as the simulation data above.

Our model requires a constant number of vehicles in a single time window. For vehicles with incomplete trajectories, due to vehicles entering or leaving the observed stretch, two approaches

---

[3]We use the STGAE-biv variant from the paper, since the version with KDE is computationally too expensive. With n trajectories in the training set, and m trajectories in the testing set, the KDE complexity is $O(mn)$. The KDE did not complete given two days of computation time.

Table 1: Abnormal *vehicle* detection performance on the test set of comprehensive scenario, where all abnormal behaviors exist on the highway. Our model is the only one that can identify anomalous cars in traffic.

	Pre@100	Pre@200	Pre@500	Avg Pre	AUC
LTI	0.21	0.205	0.210	0.093	0.623
CVM	0.15	0.140	0.142	0.089	0.617
RTR	0.16	0.190	0.228	0.109	0.598
Seq2seq	0.56	0.435	0.310	0.132	0.770
STGAE	0.18	0.180	0.178	0.093	0.641
DSAB-biv	0.15	0.150	0.114	0.062	0.655
DSAB (Ours)	**0.82**	**0.775**	**0.726**	**0.381**	**0.900**

Table 2: Abnormal *scene* detection performance. While identifying abnormal scene is an easier task than identifying specific abnormal cars, and most method can perform well, ours is still the best in performance, with an increase of 0.1 in ROC-AUC score and average precision.

	Pre@100	Pre@200	Pre@500	Avg Pre	AUC
LTI	0.89	0.860	0.834	0.690	0.726
CVM	0.67	0.625	0.640	0.662	0.714
RTR	0.89	0.895	0.864	0.769	0.744
Seq2seq	0.87	0.870	0.860	0.742	0.745
STGAE	0.82	0.740	0.776	0.666	0.678
DSAB-biv	0.73	0.665	0.610	0.470	0.551
DSAB (Ours)	**0.93**	**0.950**	**0.912**	**0.859**	**0.841**

can be used: *i*) discard the vehicles with incomplete trajectories; or *ii*) linearly extrapolate the trajectories assuming constant velocity dynamics, and then mask the extrapolated part when calculating the loss. The first approach is used for simulation data on long stretches with a small portion of incomplete trajectories. the second approach is used for real-world highD data with short stretches, and a large fraction of incomplete trajectories.
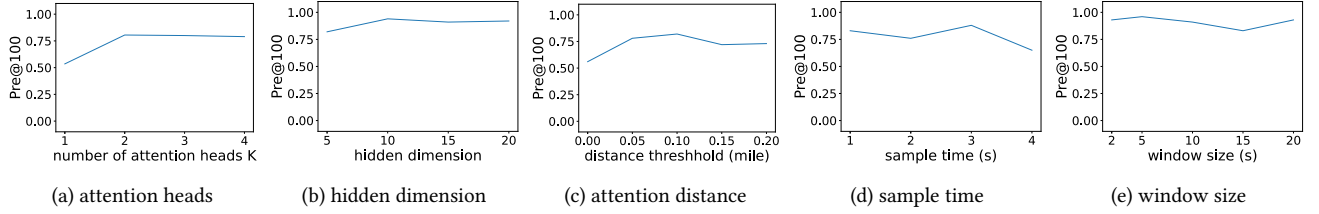
4.4 Results and analysis

In this section, we quantitatively compare our method with the baselines on simulation data. We also conduct ablation studies to see the influence of each model component on performance.

4.4.1 Model comparison. We first benchmark the methods on the test set of the comprehensive scenario, where all abnormal behaviors exist on the highway. The result for abnormal *vehicle* detection is shown in Table 1. We can see that our model is the only one that can identify the specific car that behaves abnormally in the traffic. The precision scores at different k vales are constantly above 0.7 in our method. For the other methods, the precision score are around 0.2 most of the time, and are never higher than 0.6. Our average precision is also around 0.3 higher than the second best method. The ROC-AUC score for our method is 0.9, which is more than 0.1 higher than the second best method. This shows that without consideration of neighboring vehicles, Seq2Seq, RTR, CVM and LTI are unable to detect the social anomalies. Meanwhile, although STGAE

Table 3: Abnormal car detection performance on each individual anomaly scenario. Our method is the only one that can detect slow and stalled car in traffic, and is also the best at detecting speeding cars.

	slow			speeding			tailgating			stalled		
	Pre@100	Avg Pre	AUC	Pre@100	Avg Pre	AUC	Pre@100	Avg Pre	AUC	Pre@100	Avg Pre	AUC
LTI	0.00	0.023	0.443	0.82	0.512	0.877	0.02	0.161	0.835	0.00	0.030	0.010
CVM	0.01	0.023	0.432	0.67	0.472	0.871	0.06	**0.177**	**0.840**	0.00	0.030	0.010
RTR	0.00	0.030	0.466	0.61	0.365	0.655	0.08	0.102	0.400	0.00	0.046	0.410
Seq2seq	0.02	0.046	0.668	0.85	0.549	0.923	0.18	0.091	0.709	0.03	0.056	0.676
STGAE	0.01	0.048	0.573	0.83	0.335	0.712	0.01	0.055	0.491	0.00	0.025	0.454
DSAB-biv	0.00	0.025	0.465	0.29	0.092	0.600	0.05	0.043	0.436	0.13	0.066	0.696
DSAB (Ours)	**0.97**	**0.246**	**0.841**	**1.00**	**0.907**	**0.995**	**0.33**	0.148	0.747	**0.98**	**0.148**	**0.762**

(a) attention heads   (b) hidden dimension   (c) attention distance   (d) sample time   (e) window size

Figure 2: Influence of different parameters. The model benefits from having more than one attention head, and is not sensitive to the number of heads and hidden dimensions otherwise. On spatio-temporal graph construction, when attention distance threshhold equals zero, a car does not attend to any neighbor, and the performance significantly drops. Otherwise, the model is not sensitive to the window size and sample frequency, allowing us flexibility to choose parameters.

and DSAB-biv both model the neighboring vehicle interactions via graph neural networks, they use bi-viariate Gaussian trajectory loss and cannot well capture the discrete lane-changing behaviors, and have similarly poor results.

We also report the performance of abnormal *scene* detection for all method, shown in Table 2. We can see that the performance of all methods have a significant improvement. This is because detecting abnormal scenes is easier task than detecting the specific abnormal vehicles. For example, a slow or stalled car can cause its following cars to brake abruptly, which baseline models like CVM are able to detect. Moreover, when aggregating a group of vehicles over space and time, the abnormal rate is higher, and the methods work much better on a balanced dataset. Nonetheless, our method is still the best, with an increase of 0.1 in AUC score and average precision.

4.4.2 Performance on individual anomaly scenarios. Next, we investigate the model performance on each individual anomaly scenario. The result is shown in Table 3. We can see that our method is the only one that can detect slow and stalled cars in traffic, with Precision@100 score above 0.9, compared with scores constantly less than 0.1 for other methods. While all methods perform relatively well at detecting speeding cars, our method is the best, with an improvement of 0.1 in Precision@100, and a ROC-AUC score near 1. Our method is relatively less effective in detecting tailgating cars, yet the Precision@100 is still the highest among all methods, and no method is constantly better than ours in every metric. We note that baseline methods can detect tailgating cars, because they have different car-following dynamics as described in section 4.1.

4.4.3 Ablation study. In this section, we examine the influence of different components of the model, as well as model sensitivity to the configurations of the spatio-temporal graph data.

First, we study the influence of the model parameters, with results shown in Fig 2a and 2b. The influence of number of attention heads is shown in Fig 2a. We can see that the model benefits from having more than one attention head, while being not sensitive when the number of heads is larger than one. Meanwhile, Fig 2b indicates that the model is not sensitive to the hidden dimension. Thus we choose 5 as hidden dimension for a smaller model size.

Next, we study the influence of spatio-temporal graph parameters, with results shown in Fig 2c, 2e and 2d. Spatially, the threshold of attention distance determines how far away in distance one vehicle attends to as its neighboring car. A threshold of zero means a car do not attend to any neighbor, and consequently the spatial graph is not used. We can see in Fig 2c that when the threshold equals zero, the performance significantly drops by 0.3 in Precision@100. When threshold is larger than 0.05 miles, the performance is not sensitive to the threshold value. The result indicates that the neighboring vehicles within 0.05 miles around the ego vehicle is the most important. Temporally, we study the influence of the time window size and the sample frequency. We first fix the sampling interval at 1s and vary the window size from 2s to 20s, then fix the window size at 15s and vary the sampling interval from 1s to 4s. Fig 2e and Fig 2d show the results. We can see that the model performs best when interval is less or equal to 3s, and is not sensitive to the window size and sample frequency otherwise. This allows us flexibility to choose a larger window size and coarser sampling frequency to aid computational efficiency.

4.5 Qualitative Results

In this section, we qualitatively show the performance of our model on real-world HighD data. The traffic in this data has unique characteristics, and our model is able to learn the norms and capture the anomalies that deviates from the norm.

Specifically, the speed limit for German highways is very loose, at least 75 mph, or no speed limits in some parts. And it is not uncommon to observe speeds larger than 80 mph. Thus, speeding is not ranked among the largest anomalies. On the other hand, the typical traffic speed varies significantly by lane, as shown in Fig 3a. From leftmost to rightmost lane, the average speed are 76, 69 and 56 mph respectively, calculated from training data. That is to say, the faster the car, the more to the left the car tends to be. Accordingly, the vehicle with much higher speed than its corresponding lane is detected as abnormal, even though the speed is absolutely normal when lanes are not considered. We manually inspect the vehicles with top anomaly scores, and found them to be abnormal either because of aggressive driving with dramatic acceleration/deceleration, or because of speed range violation with respect to the lane.

We show some of the top anomalies in Fig 3. Each line denotes the trajectory of a single car, with a dot denoting the starting point and triangle the end point. We add small perturbations laterally to aid visualization. The speed, acceleration and vehicle anomaly score at the corresponding time are shown. The reconstruction shows the trajectories with the largest probability. Fig 3b shows the vehicle with largest anomaly score. The abnormal vehicle 1 is cutting in front of vehicle 2, while having dramatic deceleration, forcing the other car to change lane as well. Fig 3c shows the vehicle with second largest anomaly score. The abnormal vehicle 1 is driving too fast with respect to the lane it is in – 20 mph larger than the typical speed in rightmost lane. The reconstruction actually puts vehicle 1 in the middle lane which has larger typical speed.

While we presented the top two anomalies as identified by DSAB, with more examples in Appendix E, the result illustrates that the method is applicable to real world datasets.

5 CONCLUSION

Advanced connectivity and sensing will continue to transform intelligent traffic management. In this work, we tackle an important problem of abnormal driving behavior detection, using trajectories produced by IoT highway video surveillance systems. Specifically we detect the exact anomalous vehicles considering the vehicular interaction with DSAB, an autoencoder based on Recurrent Graph Attention Networks. The results demonstrate the method captures the spatial-temporal trajectory dynamics, while considering both the neighbor interactions and the stocasticity in driving behaviors. Extensive experiments on both simulation and real-world data sets show the ability of our model to scale to large highway monitoring systems with thousands of vehicles, and detect a variety of abnormal behaviors. The performance on identifying single vehicle anomalies is state of the art, indicating potential to pinpoint specific problematic vehicles in a large traffic stream.
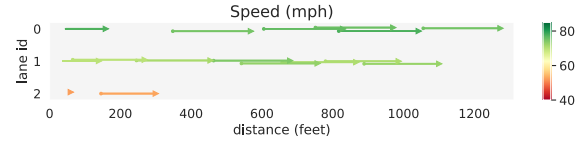
(a) Normal traffic condition. The typical speed varies by lane. From leftmost to rightmost lane, the average speed are 76, 69 and 56 mph respectively, calculated from training data
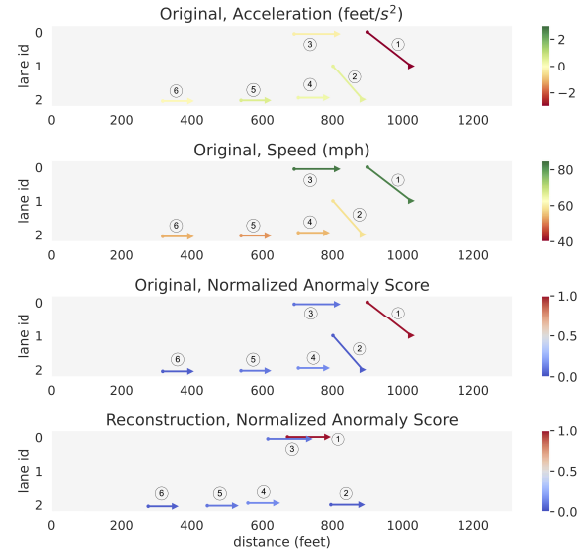
(b) Vehicle with largest anomaly score. The abnormal vehicle 1 is cutting in front of vehicle 2, while having dramatic deceleration, forcing vehicle 2 to change lane as well.
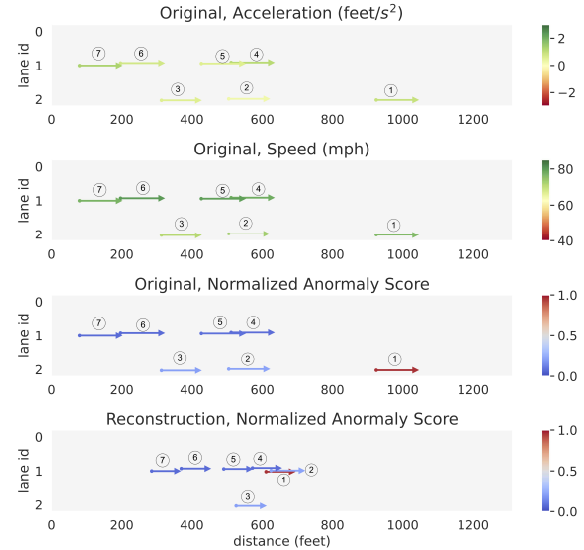
(c) Vehicle with second largest anomaly score. The abnormal vehicle 1 is driving too fast with respect to the lane it is in. The reconstruction actually puts vehicle 1 in the middle lane which has larger typical speed.

Figure 3: Qualitative study of real-world HighD traffic data

ACKNOWLEDGMENTS

REFERENCES

[1] Kazi Iftekhar Ahmed. 1999. *Modeling drivers' acceleration and lane changing behavior*. Ph. D. Dissertation. Massachusetts Institute of Technology.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 961–971.

[3] Monagi H Alkinani, Wazir Zada Khan, and Quratulain Arshad. 2020. Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements and open challenges. *Ieee Access* 8 (2020), 105008–105030.

[4] Shuai Bai, Zhiqun He, Yu Lei, Wei Wu, Chengkai Zhu, Ming Sun, and Junjie Yan. 2019. Traffic anomaly detection via perspective map based on spatial-temporal information matrix.. In *CVPR Workshops*. 117–124.

[5] Sambaran Bandyopadhyay, Saley Vishal Vivek, and MN Murty. 2020. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of the 13th international conference on web search and data mining*. 25–33.

[6] Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491* (2021).

[7] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. 2021. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *Proceedings of the 30th ACM international conference on Information & Knowledge Management*. 3747–3756.

[8] Manuel Ricardo Carlos, Mario Ezra Aragón, Luis C González, Hugo Jair Escalante, and Fernando Martínez. 2018. Evaluation of detection approaches for road anomalies based on accelerometer readings—Addressing who's who. *IEEE Transactions on Intelligent Transportation Systems* 19, 10 (2018), 3334–3343.

[9] 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.

[10] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation* (2014), 103.

[11] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).

[13] Kaize Ding, Jundong Li, Nitin Agarwal, and Huan Liu. 2021. Inductive anomaly detection on attributed networks. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 1288–1294.

[14] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 594–602.

[15] Mohamed Fazeen, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C González. 2012. Safe driving using mobile phones. *IEEE Transactions on Intelligent Transportation Systems* 13, 3 (2012), 1462–1468.

[16] Valentina Gatteschi, Alberto Cannavò, Fabrizio Lamberti, Lia Morra, and Paolo Montuschi. 2021. Comparing Algorithms for Aggressive Driving Event Detection Based on Vehicle Motion Data. *IEEE Transactions on Vehicular Technology* 71, 1 (2021), 53–68.

[17] Derek Gloudemans, Yanbing Wang, Junyi Ji, Gergely Zachar, Will Barbour, and Daniel B. Work. 2023. I-24 MOTION: An instrument for freeway traffic science. https://doi.org/10.48550/ARXIV.2301.11198

[18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2255–2264.

[19] Yue Hu, Ao Qu, and Dan Work. 2022. Detecting extreme traffic events via a context augmented graph autoencoder. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 6 (2022), 1–23.

[20] Yue Hu and Daniel B Work. 2020. Robust Tensor Recovery with Fiber Outliers for Traffic Events. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 1 (2020), 1–27.

[21] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. 2019. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6272–6281.

[22] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. 2013. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 245–251.

[23] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[24] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2118–2125. https://doi.org/10.1109/ITSC.2018.8569552

[25] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. 2017. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 336–345.

[26] Yingying Li, Jie Wu, Xue Bai, Xipeng Yang, Xiao Tan, Guanbin Li, Shilei Wen, Hongwu Zhang, and Errui Ding. 2020. Multi-granularity tracking with modularlized components for unsupervised vehicles anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 586–587.

[27] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[28] Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, et al. 2022. Benchmarking Node Outlier Detection on Graphs. *arXiv preprint arXiv:2206.10071* (2022).

[29] Yixin Liu, Shirui Pan, Yu Guang Wang, Fei Xiong, Liang Wang, Qingfeng Chen, and Vincent CS Lee. 2021. Anomaly detection in dynamic graphs via transformer. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[30] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[31] Matthias Matousek, EL-Zohairy Mohamed, Frank Kargl, Christoph Bösch, et al. 2019. Detecting anomalous driving behavior using neural networks. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2229–2235.

[32] Matthias Matousek, Mahmoud Yassin, Rens van der Heijden, Frank Kargl, et al. 2018. Robust detection of anomalous driving behavior. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5.

[33] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. 2020. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14424–14432.

[34] Youness Moukafih, Hakim Hafidi, and Mounir Ghogho. 2019. Aggressive driving detection using deep learning-based time series classification. In *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 1–5.

[35] Yulong Pei, Tianjin Huang, Werner van Ipenburg, and Mykola Pechenizkiy. 2022. ResGCN: attention-based deep residual modeling for anomaly detection on attributed networks. *Machine Learning* 111, 2 (2022), 519–541.

[36] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. 2019. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1349–1358.

[37] 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.

[38] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International conference on neural information processing*. Springer, 362–373.

[39] Andrea Stocco, Paulo J Nunes, Marcelo D'Amorim, and Paolo Tonella. 2022. Thirdeye: Attention maps for safe autonomous driving systems. In *37th IEEE/ACM International Conference on Automated Software Engineering*. 1–12.

[40] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. 2020. Misbehaviour prediction for autonomous driving systems. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering*. 359–371.

[41] Waqas Sultani, Chen Chen, and Mubarak Shah. 2018. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6479–6488.

[42] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27 (2014).

[43] Xianfeng Tang, Boqing Gong, Yanwei Yu, Huaxiu Yao, Yandong Li, Haiyong Xie, and Xiaoyu Wang. 2019. Joint modeling of dense and incomplete trajectories for citywide traffic volume inference. In *The World Wide Web Conference*. 1806–1817.

[44] Patara Trirat and Jae-Gil Lee. 2021. Df-tar: a deep fusion network for citywide traffic accident risk prediction with dangerous driving behavior. In *Proceedings of the Web Conference 2021*. 1146–1156.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[47] Anirudh Vemula, Katharina Muelling, and Jean Oh. 2018. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*. IEEE, 4601–4607.

[48] Junmin Wang and R. Rajamani. 2004. Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles? *IEEE Transactions on Vehicular Technology* 53, 5 (2004), 1480–1490. https://doi.org/10.1109/TVT.2004.832386

[49] Yu Wang and Tyler Derr. 2021. Tree decomposed graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2040–2049.

[50] Yu Wang, Wei Jin, and Tyler Derr. 2022. Graph neural networks: Self-supervised learning. *Graph Neural Networks: Foundations, Frontiers, and Applications* (2022), 391–420.

[51] 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.

[52] Jie Wu, Xionghui Wang, Xuefeng Xiao, and Yitong Wang. 2021. Box-level tube tracking and refinement for vehicles anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4112–4118.

[53] Zhiming Xu, Xiao Huang, Yue Zhao, Yushun Dong, and Jundong Li. 2022. Contrastive Attributed Network Anomaly Detection with Data Augmentation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 444–457.

[54] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 2018. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2672–2681.

[55] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.. In *IJCAI*. 4419–4425.

Table 4: Dataset statistics

		Car count	Trajectory count
Simulation	training	7,886	920,311
	comprehensive	1,281	147,254
	slow	636	75,809
	speeding	583	65,685
	tailgating	645	78,600
	stalled	1,922	263,370
HighD	training	42,106	459,187
	Testing	1,795	21,840

A DATA DETAILS

We describe the detailed settings for simulation data, then provide the data statistics for both simulation and highD data.

- Normal traffic. A standard car following model, *Modified General Motors* [1] is used, which has been demonstrated to correlate well with field traffic data. The desired speed of the vehicles follows a typical distribution found in real-world traffic, with the majority between 65-80 mph, only 5% above 85 mph and 5% below 60 mph. Recordings of varying traffic demands from 500 to 1600 vehicles/lane/hour are included, covering both free flow and congested conditions.
- Speeding scenario. We set 70% of the vehicles drive at a desired speed of 65 mph, and 30% above 85 mph. The traffic demand is 500 veh/l/hr.
- Slow scenario. We set 98% of the vehicles drive at a desired speed of 65 mph, and 2% below 50 mph. The traffic demand is 500 veh/l/hr.
- Tailgating scenario. We set 46.80% cars to follow a *Constant Time Gap* model and only a proportion of them could be tailgaters according to traffic conditions. The traffic demand is 500 veh/l/hr.
- Stalled car scenario. We randomly select 15 cars, each stopping for 5 minutes. The traffic demand is 1500 veh/l/hr.
- Comprehensive scenario. We set 89% of the vehicles drive at a desired speed of 65 mph, 10% at 85 mph, and 1% at 50 mph. In addition, we set 46.80% cars following *Constant Time Gap* model and only a proportion of them could be tailgaters according to traffic condition. We set 2 cars to each stall for 3 min. The traffic demand is 1000 veh/l/hr.

Cars are labeled anomaly only when it is actually behaving anomalously (e.g., when a car with desired speed of 85 mph can only drive at 65 mph because of traffic conditions, it is not an anomaly at the corresponding time).

Table 4 shows the statistics of car count, and the total trajectory count when divided into 10-15s time windows with 1s stride. We note that in simulation data, the traffic flow in stalled car and comprehensive scenarios are higher, because we want to check if we can detect the source anomaly car even if the stalled car causes upstream congestion.

B DESCRIPTION OF METRICS

In this section, we describe the evaluation metrics.

i) *area under the receiver operating characteristic curve* (ROC-AUC) score. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR), and the ROC-AUC score calculates the area under the ROC curve. An ROC-AUC score of 0.5 means the model is not able to discriminate anomalies, and an ROC-AUC score of 1 means perfect anomaly detection.

ii) Average precision, which summarizes the precision-recall curve into a single value, and is calculated as the weighted mean of previsions achieved at each threshold, the weight being the increase in recall from the previous threshold.

iii) Precision@k, which calculates the percentage of true anomaly among the top k samples scored by the models to be anomalies.

We note that for vehicle detection problem, since the anomaly rate is only 3%-5%, severe data imbalance issue exists. It is shown in [22] that in heavily imbalanced datasets, metrics like F1 score downgrades exponentially with data skewness, thus we do not include F1 metrics. Meanwhile, ROC-AUC score is less influenced by data imbalance, but with a caveat that model with distinctive performance could have very similar ROC-AUC score. Precision@k shows how well we rank anomalies over normal samples, and works well on imbalanced datasets where our focus is on the relatively rarely occurring anomalies.

C AGGREGATOR

We briefly describe the choice of loss aggregation over time and vehicles in section 3.6. Averaging the losses is more conservative than maximizing. For vehicle detection, since many abnormal behaviors persist for relatively long time periods over several seconds, we choose to average over time. Maximization results in around 0.1 decrease in average precision. For scene detection, we choose maximization to avoid abnormal vehicles being averaged out by normal vehicles. Averaging results in around 0.06 decrease in average precision.

D ROBUSTNESS ANALYSIS

We briefly explore the model robustness to anomalies in training data. As described in section 4.1, our training data contains a small portion of anomalies, to simulate the real-world situation of not having a perfectly clean normal dataset. To evaluate the influence of training anomalies, we compare the performance with model trained on clean training data. Table 5 shows the result. We can see that the results are similar, with differences in scores all less than 0.05. There is also no clear trend of one dataset better than the other. The result shows the model is robust to small number of anomalies in the training data.

E HIGHD ANOMALY DESCRIPTION

In this section, we provide more examples of the qualitative results for top ranked anomalies in the HighD real-world data. We note that during ranking, there might be samples with overlapping time windows, and we eliminate the repetitions for diversity. The results are shown in Fig 4 and Fig 5. Out of the top anomalies, most are because of drastic deceleration, as well as a significant difference in speed relative to the speed of the surrounding cars, or to the typical speed of the corresponding lane it is in.
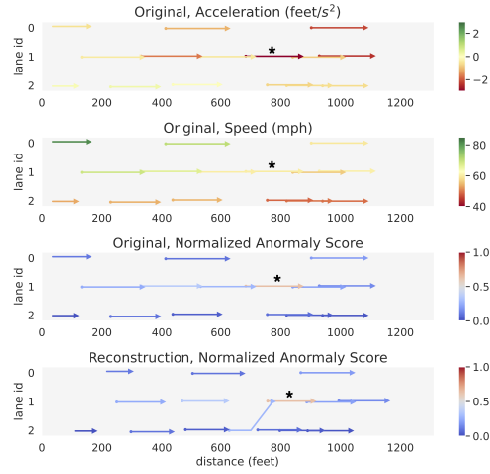
(a) Vehicle with 5th largest anomaly score. The abnormal vehicle (marked ∗) is having dramatic deceleration.
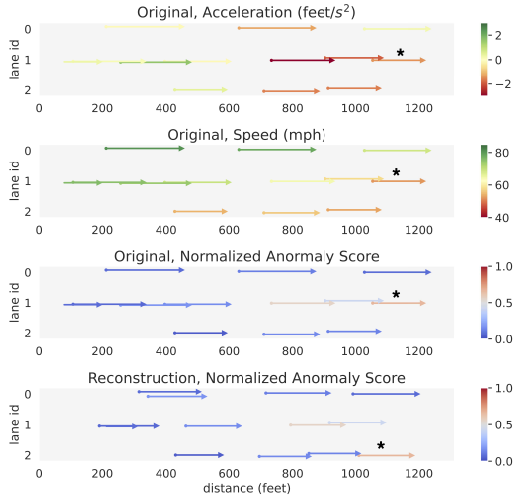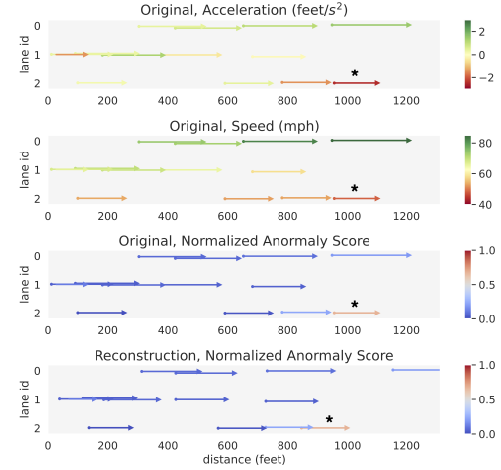

(b) Vehicle with 6th largest anomaly score. The abnormal vehicle (marked ∗) is having dramatic deceleration, while also driving too slow with respect to the lane it is in, therefore reconstructed to rightmost lane which has smaller typical speed.
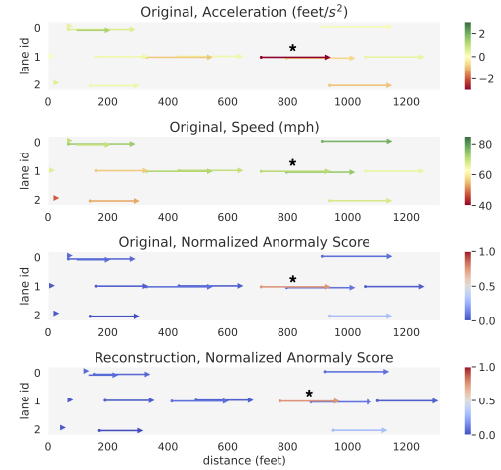
Figure 5: Additional qualitative study of real-world HighD traffic data (Part 2)

Table 5: Influence of anomalies in the training data. The performance of two training sets are comparable, with differences in scores all less than 0.05.

	Training data	Pre@100	Pre@200	Avg Pre	AUC
vehicle	Clean	0.79	0.815	0.336	0.888
detection	Polluted	0.82	0.775	0.381	0.900
scene	Clean	0.97	0.965	0.852	0.834
detection	Polluted	0.93	0.950	0.859	0.841

(a) Vehicle with 3rd largest anomaly score. The abnormal vehicle (marked ∗) is having dramatic deceleration and drives significantly slower than other cars in its lane.

(b) Vehicle with 4th largest anomaly score. The abnormal vehicle (marked ∗) is having dramatic deceleration.

Figure 4: Additional qualitative study of real-world HighD traffic data (Part1)