TAE: A Semi-supervised Controllable Behavior-aware Trajectory Generator and Predictor

Ruochen Jiao[1], Xiangguo Liu[1], Bowen Zheng[2], Dave Liang[2], and Qi Zhu[1]

***Abstract*— Trajectory generation and prediction are two interwoven tasks that play important roles in planner evaluation and decision making for intelligent vehicles. Most existing methods focus on one of the two and are optimized to directly output the final generated/predicted trajectories, which only contain limited information for critical scenario augmentation and safe planning. In this work, we propose a novel behavior-aware Trajectory Autoencoder (TAE) that explicitly models drivers' behavior such as aggressiveness and intention in the latent space, using semi-supervised adversarial autoencoder and domain knowledge in transportation. Our model addresses trajectory generation and prediction in a unified architecture and benefits both tasks: the model can generate diverse, controllable and realistic trajectories to enhance planner optimization in safety-critical and long-tailed scenarios, and it can provide prediction of critical behavior in addition to the final trajectories for decision making. Experimental results demonstrate that our method achieves promising performance on both trajectory generation and prediction.**

I. Introduction

Tremendous progress has been made for enabling autonomous driving in recent years. The autonomous driving pipeline typically consists of several modules such as sensing, perception, prediction [1], [2], planning [3], [4], [5], and control, which can be roughly divided as two parts – environment perception and decision making. Between these two parts, the prediction and generation of surrounding vehicles' trajectories can be viewed as a two-way bridge. In the forward direction, the prediction module encodes the environment information and translates it into potential future trajectories of surrounding vehicles to facilitate the planning module. Reversely, to train and evaluate the planning module, we will need to discover critical traffic scenarios and vehicle behaviors, and generate more realistic and diverse trajectories of surrounding vehicles – this is particularly important for evaluating the safety of vehicle planning as some of the "long-tail" scenarios could be quite challenging and lead to the violation of safety requirements [6], [7], [8].

Most existing works of trajectory generation or augmentation indeed try to identify risky scenarios and then extract corresponding features or styles in order to generate more safety-critical scenarios. For instance, in [9] and [10], the authors extract features and variables that lead to safety-critical scenarios and then feed them into generative models. However, the definition of the critical styles is vague, and the controllability and interpretability of the models are limited. [10] also points out that most existing works focus on generating the entire scenarios but lack control over individual agents (vehicles) and their detailed behaviors.

On the prediction side, recent works on motion forecasting [1], [11], [12], [13], [2], [14] have been focusing on the displacement error between the predicted trajectories and the ground truth, with great performance achieved. However, as mentioned in [15], the performance on current datasets has begun to plateau. Moreover, other than the trajectories themselves, those works pay little attention to other information that could also be very important for understanding surrounding vehicle's behaviors and making safe decisions. For instance, behavior prediction such as whether other vehicles may change lanes or whether they may yield to the ego vehicle, is critical information for safe planning [16]. In fact, human drivers make decisions generally relying on high-level predictions instead of exact future trajectories of surrounding vehicles. Thus, in our work, we consider *intention and aggressiveness* as such high-level behaviors of surrounding vehicles – their detailed definitions are explained in Section III but Fig. 1 shows a simple illustration – and leverage them in both predicting trajectories and in generating more diverse trajectories and behaviors. To the best of our knowledge, this is the first work that explicitly models and utilizes aggressiveness in trajectory prediction and generation.

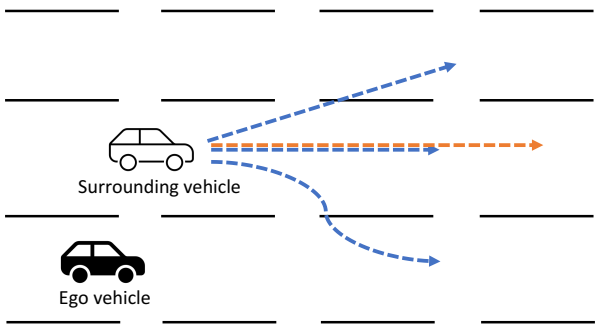


Fig. 1. Trajectories with different intentions and aggressiveness levels. Blue dashed lines demonstrate different potential intentions in changing lanes and the orange one shows a more aggressive trajectory in the current lane.

Unlike previous methods that are designed solely for either

[1]Ruochen Jiao, Xiangguo Liu, and Qi Zhu are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, 60201, USA. {ruochen.jiao, xg.liu}@u.northwestern.edu, qzhu@northwestern.edu.

[2]Bowen Zheng and Dave Liang are with Pony.ai, Fremont, CA 94538, USA. {bowen.zheng, dave.liang}@pony.ai.

We gratefully acknowledge the support from NSF grants 1834701, 1839511, 1724341, 2038853, and ONR grant N00014-19-1-2496.

trajectory prediction or generation, we propose a unified framework for both tasks using behavior-aware adversarial autoencoder architecture combined with domain knowledge in the transportation. Our goal is to design a *hierarchical and behavior-aware* predictor and a generator that can augment *realistic, diverse, explainable, and controllable* trajectories. We believe that this will facilitate both prediction and planning modules to address the critical (and potentially unsafe) tail events on the road. More specifically, our contribution can be summarized as follows:

- We propose Trajectory Autoencoder (TAE), a novel and unified architecture based on adversarial autoencoder for trajectory generation and prediction. It facilitates both tasks with behavior-level awareness and control.
- Ours is the first work to explicitly consider aggressiveness in trajectory generation/prediction. We utilize semi-supervised training along with adversarial generation and domain knowledge to model the behaviors with limited data. The method is extensible for other driving behaviors.
- We conduct experiments in a commonly-used dataset for trajectory generation and prediction. We evaluate five metrics to demonstrate the advantages of our methods in generating diverse and controllable vehicle trajectories and safety-critical scenarios, and in predicting surrounding vehicles' behaviors.

The rest of the paper is organized as follows. Section II reviews the related works. Section III presents the methodology and major components of our proposed semi-supervised behavior-aware TAE. Section IV presents the experimental results and discussions. Section V concludes the paper.

II. Background

A. Trajectory Generation and Prediction

1) Trajectory Generation: Trajectory generation or augmentation is of great significance to optimize and evaluate decision making module in autonomous driving. [17] proposes a flow-based generative model using the objective function of weighted likelihood to generate multimodal safety-critical scenarios. Their following work [9] demonstrates a generative model conditioned on road maps to bridge safe and collision driving data. The model combines conditional variational autoencoder and style transferring techniques to generate the whole risky scenario, but it cannot control agent-level trajectories. [10] proposes a RouteGAN to generate diverse trajectories for every single agent and the trajectory is influenced by a style variable. However, the latent spaces of these generative methods are not well explained with driving or transportation knowledge, especially at the behavior level. And because of the nature of GAN and style transferring techniques, the models only have rough and limited control over the generation process, which may lead to unrealistic and uncontrolled trajectories.

2) Trajectory Prediction: Recent works have applied different methods to represent the past trajectory and contexts. CNN with rasterized images [18], graph neural networks (GNN) [11], [12], transformers [19], [1], and even 3D point cloud [2] are used to encode the map and interaction information. These works achieve good performance in terms of the displacement error between ground truth and prediction. In our work, we choose to use GNN-based method for extracting features, but with the additional consideration of behavior-level prediction for safety-critical scenarios.

B. Driving Behavior Modeling

The work in [20] proposes an intention predictor based on mixture density network (MDN) [21], which considers the semantic information on the road and predicts the insertion area (region proposals) using the MDN architecture. However, the approach has to select useful features and define the insertion area manually in different scenarios. [22] designs an online two-level framework that anticipates the high-level driving policy such as forward, yielding, turning, and then feeds such intention to an optimization-based predictor.

Besides the intention of changing lane and turning, the vehicle's style such as aggressiveness will also influence its motion. Generally, aggressive vehicles tend to drive at higher acceleration. Most works measure aggressiveness based on sensors'(e.g., accelerator, gyroscope) data [23], [24] or long-term statistics [25]. Some works propose online measurement in different scenarios. [26] demonstrates an aggressiveness measurement in lane changing scenario by using a combination of space utility and safety utility, where the space utility is the available space for lane changing while the safety utility is measured by the time headway of a vehicle, which can be generalized to other driving scenarios.

C. Adversarial Autoencoder

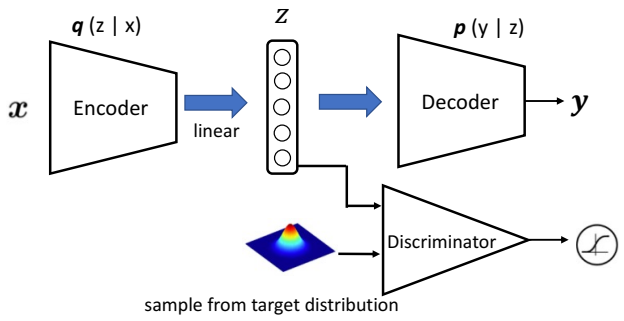


Fig. 2. Architecture of the basic adversarial autoencoder (AAE).

The variational autoencoder (VAE) [27] provides a principled method for jointly learning deep latent-variable models and corresponding inference models using stochastic gradient descent [28]. Training a VAE model consists of two steps: regularization and reconstruction. The regularization step is aimed to encode the input as certain distributions (usually Gaussian) over the latent space using Kullback-Leibler (KL) divergence, while the reconstruction step is used to decode the latent variables to the target space. In contrast to VAE that uses KL divergence and evidence lower bound, adversarial autoencoder (AAE) [29] uses adversarial learning for

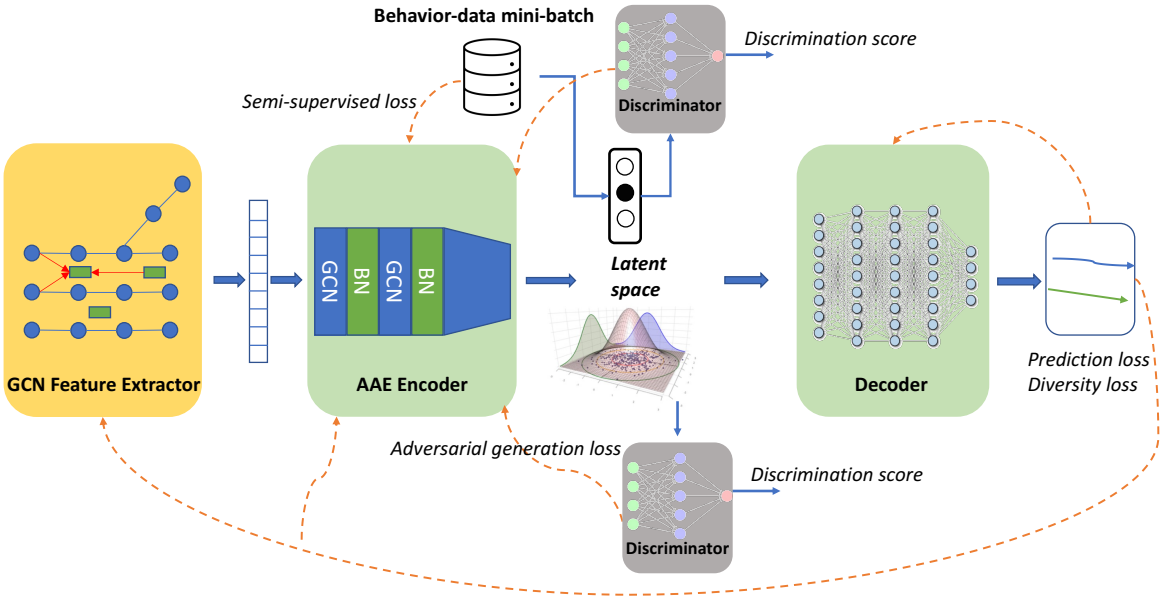


Fig. 3. Overview of our proposed TAE architecture and training pipeline.

imposing a specific distribution on the latent variables. The AAE architecture is superior to VAE in terms of imposing complicated distributions and shaping the latent space.

More specifically, let x be the input, y be the output, and z be the latent vector of an autoencoder with a deep encoder and decoder. Let $p(z)$, $q(z|x)$, $p(y|z)$ and $p_d(x)$ denote the prior distribution on the latent vectors, encoding distribution, decoding distribution and input data distribution, respectively. The encoding function of the autoencoder $q(z|x)$ defines an aggregated posterior distribution of $q(z)$ on the latent vector of the autoencoder as follows:

$$q(\mathbf{z}) = \int_{\mathbf{x}} q(\mathbf{z} \mid \mathbf{x}) p_d(\mathbf{x}) d\mathbf{x} \tag{1}$$

The $q(z)$ is expected to match the prior distribution $p(z)$. In the AAE, the encoder tries to fool the discriminators into thinking the generated latent vectors are from the prior target distribution $p(z)$.

As shown in Fig. 2, the AAE architecture maps the input x to the latent space of a Gaussian distribution. The real data sampled from the Gaussian distribution and the latent codes are fed into the discriminator. The discriminator tries to distinguish the real samples from the generated ones, and the discrimination scores are used to update the encoder to generate data following target distribution. Then the decoder reconstructs the output from the latent code z. In our work, we will extend the AAE architecture to model multiple and complex distributions, and encode label information in the latent space.

III. Our Proposed TAE Architecture

The design of our proposed TAE architecture for trajectory generation/prediction is shown in Fig. 3. In this section, we will explain the major modules in our framework and the methodology for modeling and optimization, i.e., the context feature extractor (III-A), the architecture of the semi-supervised behavior-aware AAE (III-B), the latent space modeling with prior knowledge of vehicle's behavior (III-C), the optimization pipeline (III-D), and several additional improvements (III-E).

A. Context Modeling

In order to capture features of the environment, we need to consider the past trajectories, the interactions between vehicles, and the map information. Similar to [11], we use the one-dimensional dilated convolutional neural network to extract features from history trajectories and utilize the graph convolutional network (GCN) to model the graphed map information and the interaction. The GCN-based method has shown good performance in modeling transportation contexts since most vehicles drive on the structured roads, especially in urban scenarios. Then, the feature extractor applies an attention mechanism to combine the information and outputs a 128-dimensional feature for each agent. The pipeline of the feature extraction is shown in Fig. 4.

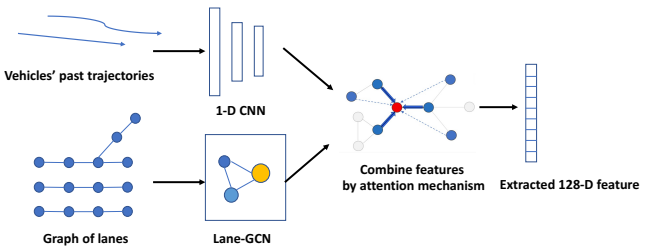


Fig. 4. Feature Extractor: extract trajectories using 1-D CNN; model interactions and structured map information by GCN [11].

B. Behavior-aware Semi-supervised AAE

The AAE architecture itself blends the autoencoder architecture with the adversarial loss concept introduced by GAN, and replaces the KL divergence in VAE with adversarial loss to regularize diverse and complex distributions of latent space. In our model, we further utilize semi-supervised learning to model the driving behavior in the latent space by incorporating the limited label information. The architecture of the proposed semi-supervised behavior-aware AAE is shown in Fig. 5. The model consists of an encoder, behavior-aware and remaining latent space, discriminators for different latent vectors, and a decoder.

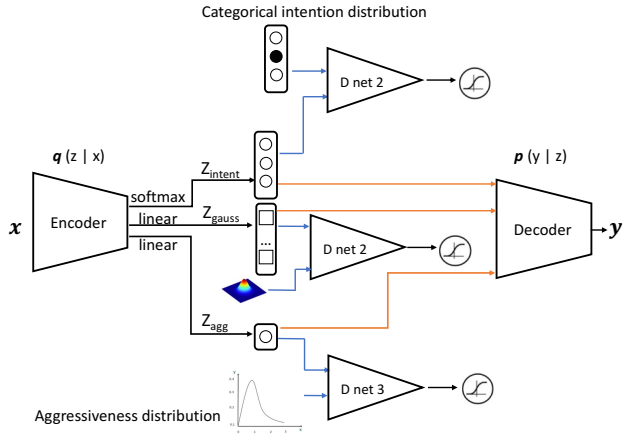


Fig. 5. Behavior-aware adversarial autoencoder.

The input x is a 128-dimensional feature generated by the GCN-based feature extractor. The multi-head encoder based on a two-layer GCN projects the features to a lower-dimensional latent space.

Our proposed AAE model has three parts of the latent space, which follow different distributions. They are three-dimensional intention latent vector z_{intent}, following categorical distributions, one-dimensional aggressiveness latent vector z_{agg}, following a log-normal distribution, and remaining latent vector z_{gauss}, following Gaussian distributions. For the dimension of the remaining latent vector, we notice a trade-off between the generated trajectory's smoothness and behavior's controllability and we allocate six dimensions to the remaining latent space after preliminary experiments.

For each group of latent variables, we input the samples from real target distribution and the generated latent variables to the corresponding discriminators (D net in Fig. 5) to regularize the latent distribution. For instance, the discriminator for intention latent vector is trained to distinguish our generated latent vector from the sample in real categorical distribution. By adversarial learning, we can force the three latent vectors to follow their corresponding distributions.

After the adversarial generation learning stage, we collect data and labels for the semi-supervised aggressiveness and intention modeling. Only part of the vehicles' behaviors can be identified and labeled, but in general, these behaviors are in certain distributions. This is the reason why the semi-supervised learning works for training the behavior vectors. In the semi-supervised training mini-batch, we optimize the encoder to predict the real intentions and aggressiveness levels in the latent space, based on the limited labelled data.

Besides regularizing the latent space and optimizing the encoder, the model needs to generate realistic trajectories. The latent vectors are concatenated and fed to the decoder. The decoder is a three-layer fully-connected network that maps the latent vectors into future trajectories. Finally, we update the whole pipeline including feature extractor, encoder and decoder to make the generated trajectory close to the reference.

The details of behavior modeling and multi-stage optimization process are introduced in the following sections.

C. Latent Space Modeling

Most previous works directly predict or generate the waypoints for future trajectories and the VAE-based works generally use unified latent variables.

With our semi-supervised AAE architecture, we can represent both distribution and label information of important driving features in the latent space if we define them in a learnable way. In this work, we have two behavior latent vectors, aggressiveness and intention.

1) Aggressiveness: Aggressiveness is an important feature of vehicle's behavior. Conservative vehicles and aggressive vehicles may take different actions even in the identical scenario. However, it is still an open question to measure and predict the aggressiveness, especially in a general setting. As mentioned in Section II, some recent works [26], [30] propose different measurements of aggressiveness in specific scenarios such as lane changing or merging. In our work, we consider time headway as a common measurement when building the aggressiveness model, which measures the time difference between two successive vehicles when they cross a given point. We believe that time headway is a feature that we can capture in most driving scenarios and it can stand for vehicles' aggressiveness, especially in the longitudinal direction. Intuitively, the shorter the time headway is, the more aggressive the driving behavior is. We learn such attribute in a semi-supervised way because only some vehicles have close and observable interaction with the vehicle in front of them but we can assume every vehicle has its own intrinsic level of aggressiveness that follows a general distribution.

To model the latent variable of aggressiveness by proposed AAE, we should have prior knowledge of the distribution of the aggressiveness, which could be influenced by many different factors such as traffic scenarios and vehicles' behaviors [31]. Log-normal distribution, Gamma distribution and normal distributions are potential distribution to model the time headway in various scenarios. Here we focus on the urban scenario and calculate time headway of all valid cases in the Argoverse [32] motion forecasting dataset. The data histograms and fitted distributions are shown in Fig. 6.

We notice that the log-normal distribution fits the aggressiveness data best with the lowest KL divergence (0.017) and sum of squared error (0.16). In our model, we use

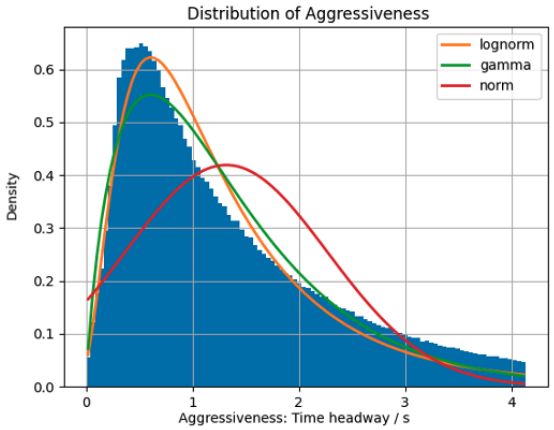


Fig. 6. The histogram and fitted distribution of valid aggressiveness (time headway) in the Argoverse motion forecasting dataset.

the log-normal distribution as the prior distribution for the discriminator of aggressiveness latent vector.

In the semi-supervised learning phase, we collect labelled aggressiveness mini-batch and train the latent vectors to match their true values as a regression problem.

2) Intention: In our work, we represent intentions by three simple but reasonable classes: moving forward, turning/changing lane to the left, and turning/changing lane to the right. We are inspired by human-driving vehicles that inform other vehicles of the ego vehicle's intentions by using turn signals. These three intentions are discrete by nature and we model them with categorical distribution. We only label the vehicles that show clear intention in a long enough time frame (5 seconds in experiments).

D. Optimization Pipeline

To produce realistic, diverse and controllable trajectories, our model is designed to optimize and balance several different targets. The optimization process consists of three phases: prediction phase, regularization phase and semi-supervised phase.

First, in the prediction phase, the whole model including the feature extractor, encoder and decoder is optimized to produce accurate and realistic trajectories. We apply the smooth L1 loss as shown below in (2) on all time steps to calculate the distance between the generated trajectory $\hat{y}$ and ground truth y.

$$Loss_{pred}(y_i,\hat{y}_i) = \begin{cases} 0.5(y_i-\hat{y}_i)^2 & \text{if } \|y_i-\hat{y}_i\|<1 \\ \|y_i-\hat{y}_i\|-0.5 & \text{otherwise} \end{cases} \quad (2)$$

To model the latent space, we apply both the adversarial learning loss and the semi-supervised learning loss. We utilize three different generators and discriminators to regularize the distribution of the latent space by adversarial learning. The adversarial regularization loss is shown in (3). Here x represents the input of the encoder G, and m equals 3, corresponding to the three distributions: Gaussian, Log-normal and Categorical distributions.

$$Loss_{adv}(x) = \frac{1}{m}\sum_{i=1}^{m}\log\left(1-D_i\left(G_i\left(x\right)\right)\right) \quad (3)$$

For the discriminator D, we train them by maximizing the average of the log probability of real latent samples s and the log of the inverse probability for fake latent samples:

$$Loss_D(x,s) = \log D_i\left(s_i\right) + \log\left(1-D_i\left(G_i\left(x\right)\right)\right) \quad (4)$$

In the semi-supervised phase, we update the encoder on labelled data of the behavior-aware latent space to make the latent variables explainable. To model the latent variable of aggressiveness z_{agg}, the encoder is trained to minimize the mean square error of the predicted aggressiveness and labeled ones l_{agg}. To represent the intention vector z_{int}, the encoder is updated to minimize the cross entropy cost on a labeled mini-batch (class labels are l_{int}), which is modeled as a classification problem. The total loss of semi-supervised learning phase is shown in equation (5).

$$Loss_{Semi}(z_{agg},z_{int},l_{agg},l_{int}) = -\sum_{i=1}^{3} l_{int}\log z_{int} + (l_{agg}-z_{agg})^2 \quad (5)$$

The optimization pipeline is illustrated as Algorithm 1.

Algorithm 1 Optimization Pipelines

1: **Initialize:** feature extractor F, AAE encoder G, decoder R, discriminator D_i, target distribution p_i, i = 1,2,3.
2: **Input:** past trajectories t and map graph m.
3: **for** each batch **do**
4: Let features $x = F(t,m)$.
5: Let latent vectors $z = G(x)$.
6: Sample s_i from target distribution p_i and calculate $D_i(z_i)$ and $D_i(s_i)$.
7: Update G by adversarial generation loss $Loss_{adv}$ (3).
8: Update D_i by discrimination loss $Loss_D$ (4).
9: Obtain the labelled mini-batches for intention and aggressiveness, respectively.
10: Calculate $Loss_{semi}$ (5) and update G.
11: Concatenate the latent vectors and feed to decoder R $\hat{y} = R(z)$.
12: Calculate the prediction loss $Loss_{pred}(y,\hat{y})$ and update F, G, R.
13: **end for**

E. Diverse Generation and Multi-modal Prediction

In our preliminary tests, we noticed that our model only had limited capacity to generate diverse trajectories, even though we had already shaped and trained the latent space to model the aggressiveness and intention using our proposed architecture. We found that in the training, the sampling was aimed to maximize the likelihood that may only produce samples corresponding to the major modes of the data distribution [33]. We also did not have control over all latent variables, and particularly, we only used one-dimensional variable to represent the aggressiveness. To address these

problems, we introduce a diversity-promoting prior over samples as a diversity objective to optimize the latent mappings for improving sample and decoding diversity. We calculate the diversity loss as in the equation (6) [33] based on a pairwise Euclidean distance among generated trajectories. In (6), x_i is the i-th generated trajectory and σ_d is used to normalize the distance.

$$Loss_d(X) = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i}^{K} \exp\left(-\frac{D^2(\mathbf{x}_i, \mathbf{x}_j)}{\sigma_d} \right) \quad (6)$$

In the diversity optimization stage, we sample different behaviors and feed corresponding latent vectors to the decoder. By combining this loss with prediction loss, we can promote the generation diversity of different modes and improve the aggressiveness' control over the trajectories.

For trajectory prediction, we add an additional classifier to select a most possible trajectory from different ones, which enhances the performance of multi-modal prediction.

IV. Experimental Results

A. Experiment Settings

We train our model on the Argoverse motion forecasting dataset [32] and evaluate the generation and prediction performance on the corresponding validation and test sets. The Argoverse motion forecasting benchmark has more than 30K scenarios collected in Miami and Pittsburgh. Each scenario has detailed graph of road map and multiple agent trajectories sampled at 10 Hz. In the motion forecasting and generation tasks, trajectories of the first 2 seconds are offered as input data. The dataset contains the straight road and intersection scenarios, most of which are easy and safe cases.

We train our model on an NVIDIA Titan RTX platform for 30 epoches. The batch size is 32. The learning rates are set as $1e-4$, $1e-5$, $1e-5$ and $5e-5$ for the Adam optimizers of prediction, adversarial generation learning, discrimination and semi-supervised phases, respectively.

B. Diverse and Controllable Trajectory Generation

To measure the performance of the trajectory generation, we 1) calculate the cluster numbers of the dataset to evaluate the augmented complexity and diversity, 2) visualize the generated trajectories to demonstrate the controllability and interpretability in generation, and 3) sweep the behavior latent space and count the risky cases to test the capability of generating safety-critical scenarios.

First, we cluster normalized generated trajectories and obtain the cluster numbers with different thresholds. Generally, a larger cluster number represents a higher level of complexity and diversity, while the threshold constrains the minimum proportion the clusters. Since we do not have clear labels for generated trajectories, we utilize Dirichlet Process Gaussian Mixture Models (DPGMM) [34] to cluster the dataset. DPGMM is an infinite mixture model with the Dirichlet Process as a prior distribution on the number of clusters, so it does not need predefined cluster number.

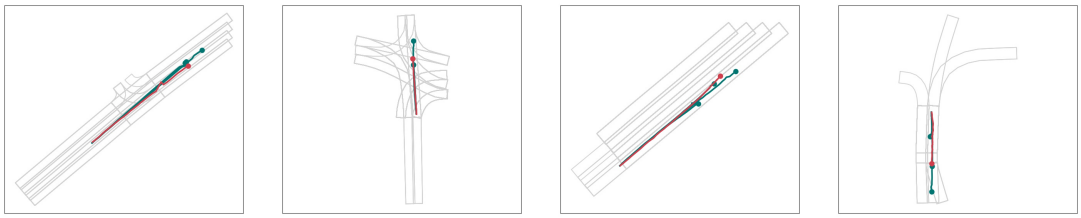
In the experiments, we generate trajectories based on the scenarios in the Argoverse validation set. For each scenario, we generate six trajectories that are: 1) the most likely, 2) aggressive, 3) conservative, 4) turning (changing lane to) left, 5) turning (changing lane to) right, and 6) moving forward. We compare the results with trajectories generated by other representative and state-of-the-art trajectory generator/predictors including GCN-based [11], transformer-based [1] and autoencoder-based works. All the results are in the same scenarios and of the same number of trajectories. We only count the clusters containing more data than the threshold ratio. The result in Table I shows that **our model can generate more diverse and complex scenarios** based on past reference trajectories. Our model significantly outperforms other methods, especially when the threshold is high, which means that our model can effectively augment rare behaviors and scenarios in the dataset (e.g., changing lane on the straight road), and balance their distribution. This augmentation can benefit the training and evaluation of prediction and planning modules.

TABLE I
Generation Diversity of Argoverse Validation Dataset

Model \ Threshold	0.05↑	0.03↑	0.01↑
GCN+Multi-head predictor[11]	6	9	29
mmTransformer[1]	2	6	33
Vanilla AAE	2	5	18
Ours	10	13	35

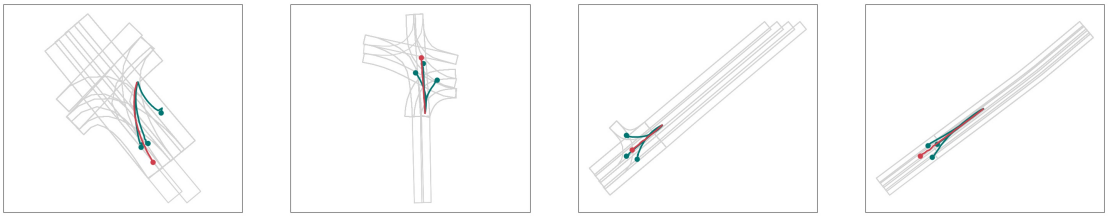
Second, for the visualization, after inputting the past trajectories and road map, we adjust the aggressiveness (Fig. 7) and intention (Fig. 8), respectively. We can find that the behaviors are represented and disentangled in the latent space. The change of intention mainly leads to turning or lane changing according to the contexts. And the change of aggressiveness can be decoded to different accelerations in the longitudinal direction.

Finally, to test the capability of generating safety-critical scenarios, we count the number of risky scenarios with different aggressiveness and intention settings. We define the risky scenario as situations where closest distance between two vehicles is less than 0.5 meter. We assume that the ego vehicle has an ideal planner that exactly follows the reference trajectory in the dataset while we can manually change other vehicles' behaviors by TAE. We first get the most-likely generation result with its aggressiveness (*agg*) and sweep the aggressiveness from most conservative to most aggressive. The result in Table II shows that more aggressive behaviors will cause more risky situations on the road *exponentially*. A conservative vehicle will be safer in general, although, being too conservative could also be unfavorable for safety, which matches our driving experience. We also switch the intention to the values representing more actively turning or lane changing behavior and observe an increase of the risky scenarios by 35.5%.



(a) (b) (c) (d)

Fig. 7. Trajectories generated with different levels of aggressiveness. The green trajectories are generated ones and the red trajectories are the references.



(a) (b) (c) (d)

Fig. 8. Trajectories generated with different intentions. The green trajectories are generated ones and the red trajectories are the references. We generate 1) most-likely trajectory, 2) turning/changing lane to the left, 3) turning/changing lane to the right.

TABLE II

SAFETY CRITICAL SCENARIOS CHANGES

$agg-3$	$agg-2$	$agg-1$	$agg+0.5$	$agg+1$	$agg+1.5$
-10.0%	-10.8%	-6.0%	$+8.4\%$	$+65.9\%$	$+227.5\%$

C. Behavior-aware Trajectory Prediction

Without manipulating in the latent space, we can directly obtain the behavior-aware motion predictor. We evaluate the accuracy of 1) behavior prediction in the latent space, and 2) most-likely trajectory in the final stage. The Table III shows the performance of behavior prediction. The intention prediction can be regarded as a classification problem and the accuracy of our approach is 89.16%. For the aggressiveness prediction, we use mean square error (MSE) to measure the accuracy. Our model can achieve an average MSE of 0.36 s^2, given the standard deviation of the aggressiveness over the dataset is 0.96.

TABLE III

BEHAVIOR PREDICTION RESULTS

	Accuracy/MSE
Intention ↑	89.16%
Aggressiveness ↓	0.36

To assess the average performance of trajectory prediction, we measure the average displacement error (ADE) between predicted and ground truth waypoints, and final displacement error (FDE) between last-predicted and ground truth waypoint. Table IV shows the results of our model and recent state-of-the-art works. Despite focusing more on long-tail events and diverse trajectory generation, our model achieves similar prediction performance in these average metrics and the results show that our model can generate natural and realistic trajectories based on a small latent space.

TABLE IV

PREDICTION RESULTS

Model \ Metrics	ADE↓	FDE↓
Argoverse Baseline (NN)[32]	3.45	7.88
Jean[35]	1.74	4.24
TNT[12]	1.77	3.91
LaneGCN[11]	1.71	3.78
WIMP[36]	1.82	4.03
TPCN[2]	1.66	3.69
Ours	1.73	3.83

D. Discussions

By explicitly modeling the behavior-level vehicle intention and aggressiveness in the latent space, our framework can provide more diverse and controllable trajectory generation as well as good prediction performance in a unified architecture. And we believe that the semi-supervised latent space modeling can be extended to more behaviors.

During experiments, we have a few observations that can be further explored. In latent space modeling, we find that there exist distribution imbalances in some attributes like intentions (more than 60% scenarios are moving forward), and it reveals a promising direction of improving the trajectory modeling. We demonstrate that our model can produce realistic trajectories with a smaller latent space compared to

other works. However, there still exists an information loss when encoding features to the lower dimensional latent space and this is part of the reason why works such as TPCN [2] have better prediction displacement error than ours. In future work, we plan to utilize our framework to further model more driving behaviors, test other context extractors, and add safety constraints to the current framework.

V. Conclusion

In this work, we propose a behavior-aware trajectory autoencoder (TAE) for both vehicle trajectory generation and prediction. We embed the domain knowledge such as intention and aggressiveness into the latent space and optimize the model with limited labelled data. Our method can generate realistic, diverse, and controllable trajectories, which could greatly benefit reliable decision making and planning evaluation in critical scenarios.

References

[1] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.

[2] M. Ye, T. Cao, and Q. Chen, "Tpcn: Temporal point cloud networks for motion forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 318–11 327.

[3] X. Liu, G. Zhao, N. Masoud, and Q. Zhu, "Trajectory planning for connected and automated vehicles: cruising, lane changing, and platooning," *SAE International Journal of Connected and Automated Vehicles*, October 2021.

[4] X. Liu, N. Masoud, Q. Zhu, and A. Khojandi, "A markov decision process framework to incorporate network-level data in motion planning for connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 136, p. 103550, 2022.

[5] X. Liu, R. Jiao, B. Zheng, D. Liang, and Q. Zhu, "Neural network based interactive lane changing planner in dense traffic with safety guarantee," *arXiv preprint arXiv:2201.09112*, 2022.

[6] Q. Zhu, C. Huang, R. Jiao, S. Lan, H. Liang, X. Liu, Y. Wang, Z. Wang, and S. Xu, "Safety-assured design and adaptation of learning-enabled autonomous systems," in *26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2021.

[7] Q. Zhu, W. Li, H. Kim, Y. Xiang, K. Wardega, Z. Wang, Y. Wang, H. Liang, C. Huang, J. Fan, and H. Choi, "Know the unknowns: Addressing disturbances and uncertainties in autonomous systems," in *Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD'20)*, 2020.

[8] R. Jiao, H. Liang, T. Sato, J. Shen, Q. A. Chen, and Q. Zhu, "End-to-end uncertainty-based mitigation of adversarial attacks to automated lane centering," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 266–273.

[9] W. Ding, M. Xu, and D. Zhao, "Cmts: A conditional multiple trajectory synthesizer for generating safety-critical driving scenarios," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4314–4321.

[10] Z.-H. Yin, L. Sun, L. Sun, M. Tomizuka, and W. Zhan, "Diverse critical interaction generation for planning and planner evaluation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7036–7043.

[11] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *European Conference on Computer Vision*. Springer, 2020, pp. 541–556.

[12] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "Tnt: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.

[13] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, "Scene transformer: A unified multi-task model for behavior prediction and planning," *arXiv e-prints*, pp. arXiv–2106, 2021.

[14] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Gohome: Graph-oriented heatmap output forfuture motion estimation," *arXiv e-prints*, pp. arXiv–2109, 2021.

[15] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2021.

[16] X. Liu, C. Huang, Y. Wang, B. Zheng, and Q. Zhu, "Physics-aware safety-assured design of hierarchical neural network based planner," *13th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'22)*, 2022.

[17] W. Ding, B. Chen, B. Li, K. J. Eun, and D. Zhao, "Multimodal safety-critical scenarios generation for decision-making algorithms evaluation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1551–1558, 2021.

[18] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.

[19] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9813–9823.

[20] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 307–313.

[21] C. M. Bishop, "Mixture density networks," 1994.

[22] W. Ding and S. Shen, "Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9610–9616.

[23] Y. Moukafih, H. Hafidi, and M. Ghogho, "Aggressive driving detection using deep learning-based time series classification," in *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2019, pp. 1–5.

[24] Y. Ma, Z. Zhang, S. Chen, Y. Yu, and K. Tang, "A comparative study of aggressive driving behavior recognition algorithms based on vehicle motion data," *IEEE Access*, vol. 7, pp. 8028–8038, 2018.

[25] A. B. R. Gonzalez, M. R. Wilby, J. J. V. Diaz, and C. S. Ávila, "Modeling and detecting aggressiveness from driving signals," *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 4, pp. 1419–1428, 2014.

[26] H. Yu, H. E. Tseng, and R. Langari, "A human-like game theory-based controller for automatic lane changing," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 140–158, 2018.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] D. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

[29] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[30] X. Liu, N. Masoud, and Q. Zhu, "Impact of sharing driving attitude information: A quantitative study on lane changing," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1998–2005.

[31] D.-H. Ha, M. Aron, and S. Cohen, "Time headway variable and probabilistic modeling," *Transportation Research Part C: Emerging Technologies*, vol. 25, pp. 181–201, 2012.

[32] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[33] Y. Yuan and K. Kitani, "Dlow: Diversifying latent flows for diverse human motion prediction," in *European Conference on Computer Vision*. Springer, 2020, pp. 346–364.

[34] D. Görür and C. Edward Rasmussen, "Dirichlet process gaussian mixture models: Choice of the base distribution," *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 653–664, 2010.

[35] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9638–9644.

[36] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.