Diversity-Aware Vehicle Motion Prediction via Latent Semantic Sampling

Xin Huang[1,2], Stephen G. McGill[1], Jonathan A. DeCastro[1],
Brian C. Williams[2], Luke Fletcher[1], John J. Leonard[1,2], Guy Rosman[1]

***Abstract*—Vehicle trajectory prediction is crucial for autonomous driving and advanced driver assistant systems. While existing approaches may sample from a predicted distribution of vehicle trajectories, they lack the ability to explore it – a key ability for evaluating safety from a planning and verification perspective. In this work, we devise a novel approach for generating realistic and diverse vehicle trajectories. We extend the generative adversarial network (GAN) framework with a low-dimensional *approximate semantic* space, and shape that space to capture semantics such as merging and turning. We sample from this space in a way that mimics the predicted distribution, but allows us to control coverage of semantically distinct outcomes. We validate our approach on a publicly available dataset and show results that achieve state of the art prediction performance, while providing improved coverage of the space of predicted trajectory semantics.**

***Index Terms*—Autonomous Driving, Motion Prediction, Adversarial Learning, Metric Learning, Explainable AI**

I. Introduction

Vehicle trajectory prediction is crucial for autonomous driving and advanced driver assistant systems. While existing literature relates to improving the accuracy of prediction [1]–[5], the diversity of the predicted trajectories [6], [7] must be explored. High accuracy implies good approximation of the true distribution according to some performance metric, but emphasizing diversity allows prediction approaches to access low-probability but high-importance parts of the state space. Diverse trajectory sampling provides coverage of possible actions for surrounding vehicles and facilitates safe motion planning and accurate behavior modeling for nearby vehicles in simulation. For instance, at an intersection, sampling distinct outcomes, such as left or right turns, rather than simply predicting going forward, provides benefits in verification. Different maneuvers can have radically different outcomes, and missing one of them can be catastrophic. Sampling efficiently proves difficult in such scenarios, as neither the distribution of trajectories nor the definition of semantically distinct outcomes has an analytical form. Additionally, expensive roll-outs of a future trajectory are required to define its utility, which considers environment of the car and nearby agents.

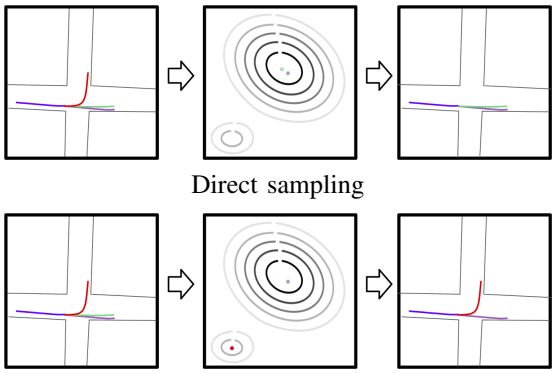


Fig. 1: Top to bottom: direct vs. latent semantic sampling. In latent semantic sampling, representative samples are taken in the latent space, with weights associated from the distribution. In this way, a few samples can capture relevant semantic aspects, while ensuring consistency with the true prediction distribution.

In this paper, we propose a model that handles both accuracy and diversity by incorporating a latent semantic layer into the trajectory generation step. This layer should represent approximate high-level vehicle behaviors, matching semantic distinctions when they exist. We expect it to be effectively low-dimensional, since a driver can perform only a few distinct maneuvers at any given moment. Therefore, enumerating low dimensional samples should be feasible; however, we wish to do so without matching the driver's behaviors into a fixed taxonomy. We illustrate this idea in Figure 1, where the goal is to produce diverse trajectory predictions and cover distinct outcomes. The top row shows traditional sampling, which fail to sample diverse behaviors efficiently. The bottom row demonstrates our latent semantic sampling technique, which is able to capture both maneuvers in the intersection.

We do so by shaping the notion of similarity in the intermediate layer activation via metric learning [8]. We train the latent semantic layer activations to match anno-

This work was part of X. Huang's internship at Toyota Research Institute (TRI). However, this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

[1]Toyota Research Institute, Cambridge, MA 02139, USA
[2]Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA 01239, USA {xhuang}@csail.mit.edu

tations of high-level labels where these exist. Distances between two trajectories should be large if they represent different semantic labels, and should be small otherwise.

In addition to prediction, our model can produce behavior samples for simulation and verification. Verification of safety properties for a given driving strategy is challenging, since it requires numerous simulations using predictive models instantiated over a large sampling space of initial agent conditions, road configurations, etc. A semantically-meaningful, low-dimensional latent space provides the advantage of efficient sampling of all possible behaviors, requiring fewer simulations to find rare events that affect safety (e.g. collisions between cars).

Finally, our proposed latent state affords some interpretation of the network, which is crucial in safety-critical tasks such as autonomous driving. By tuning the high-level latent state, our samples better cover the human intuition about diverse outcomes.

Our work has three main contributions. i) We extend a generative adversarial network to produce diverse and realistic future vehicle trajectories. We process the noise samples into two independent latent vectors, utilizing loss functions to disentangle them. The high-level vector captures semantic properties of trajectories, while the low-level layer maintains spatial and social context. ii) We describe an efficient sampling method to cover the possible future actions, which is important for safe motion planning and realistic behavior modeling in simulation. iii) We validate our approach on a publicly available dataset with vehicle trajectories collected in urban driving. Quantitative results show our method outperforming state-of-the-art approaches, while in qualitative scenarios it efficiently generates diversified trajectories.

The remainder of the paper is organized as follows. We introduce relevant work in Section I-A, and our problem formulation and proposed method in Section II. We demonstrate results in vehicle motion prediction in Section III, followed by a summary and a discussion of future work in Section IV.

A. Related Works

Our work relates to several topics in probabilistic trajectory prediction. Unlike deterministic alternatives [1], it allows us reason about the uncertainty of driver's behaviors. There are several representations that underlie reasoning about trajectories. [2], [3], [9]–[11] predict future vehicle trajectories as Gaussian mixture models, whereas [12] utilizes a grid-based map. In our work, we focus on generating trajectory samples directly from an approximated distribution space, using a sequential network, similar to [6], [7].

For longer term prediction horizons, additional context cues are needed from the driving environment. Spatial context, including as mapped lanes, not only indicates the possible options a vehicle may take (especially at intersections), but also improves the prediction accuracy, as vehicles usually follow lane centers closely [4], [13]. Another important cue is social context based on nearby agents, affording reasoning about interaction among agents [5], [7], [9], [14]. Our method takes advantage of these two cues by feeding map data and nearby agent positions into our model, improving the accuracy of predictions over a few seconds.

Recently proposed generative adversarial networks (GANs) can sample trajectories by utilizing a generator of vehicle trajectories and a discriminator that distinguishes real trajectories and trajectories produced by the generator [7], [14]–[16]. Despite their success, efficiently producing unlikely events, such as lane changes and turns, remains a challenge. These events are important to consider as they can pose a significant risk and affect driving decisions.

Hybrid maneuver-based models [17] are effective in producing distinct vehicle behaviors. They first classify maneuvers based on vehicle trajectories, and then predict future positions conditioned on a maneuver. As such, they are restricted to cases where pre-defined maneuvers are well defined. Similar to [10], our method allows more general cases dealing with undefined semantics, including multi-vehicle interactions.

Beyond prediction, recent learning models use an intermediate representation in probabilistic network models to improve sample efficiency and coverage. [10] utilizes a set of discrete latent variables to represent different driver intentions and behaviors. [18] has shown that there exist semantics in the latent space of generative adverserial networks (GANs), and [19] successfully decomposes the latent factor in a GAN into structured semantic parts. In addition to GANs, [20] has learned disentangled latent representations in a variational autoencoder (VAE) framework to ground spatial relations between objects. Unlike the information bottleneck motivation of [19], we use metric learning [8] to capture information such as maneuvers and interactions. The low dimensionality of the semantics space allows us to obtain distinct vehicle behaviors efficiently. In a relevant work, [21] proposes to generate samples in a potential field learned by the discriminator to approximate the real probability distribution of data accurately, and to ensure sample diversity.

Finally, our work has applications to sampling and estimation of rare events for verification, which is its own active field, see [22]–[26] and references therein. The closest work to ours is [24], [26], which also propose sample-based estimation of probabilities. As opposed to probability estimation under standard driving, our work focuses explicitly on sampling from diverse modes of

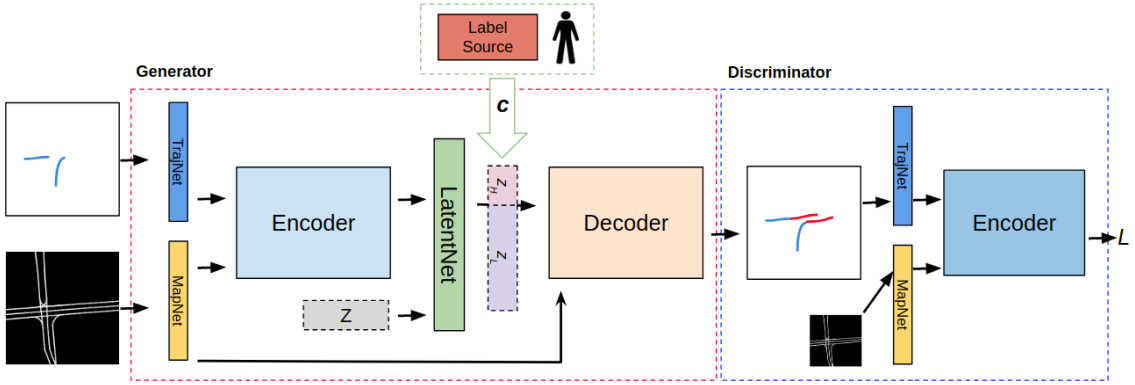


Fig. 2: Architecture diagram of prediction model. We shape the space of the intermediate vector z_H to resemble a human's concept of distances and then use it to modify the samples that are fed to the decoder.

behaviors.

II. Model

Here, we present the problem formulation and describe the model underlying our work, including loss functions and our proposed sampling procedure.

A. Problem Formulation

The input to the trajectory prediction problem includes a sequence of observed vehicle trajectories $\mathbf{X} = X_1, X_2, \ldots, X_{t_{obs}}$, as well as the surrounding lanes, given as their centerline coordinates, denoted as M. The goal is to predict a set of possible future trajectories $\hat{\mathbf{Y}} = \hat{Y}_{t_{obs}+1}, \hat{Y}_{t_{obs}+2}, \ldots, \hat{Y}_{t_{obs}+t_{pred}}$, where the acausal future trajectories are denoted as $\mathbf{Y} = Y_{t_{obs}+1}, Y_{t_{obs}+2}, \ldots, Y_{t_{obs}+t_{pred}}$.

In the probabilistic setting, since multiple future trajectory sets are possible, the goal is to estimate the predicted probability distribution $P(\mathbf{Y}|\mathbf{X}, M)$. Many of the modern approaches sample from $P(\mathbf{Y}|\mathbf{X}, M)$ in the lack of a closed-form expression for it, requiring some form of sample generation approaches, such as traditional ones such as MCMC and particle filters [27], planning based approaches such as RRTs [28], and GANs and other probabilistic generative networks [5], [6].

B. Model Overview

We now describe the network structure and sampling approach, as illustrated in Figure 2. The trajectory generator takes the past trajectory of target vehicles, a map of lane centerlines and a noise sample, before producing samples of future trajectories. The discriminator identifies whether the generated trajectory is realistic.

In addition to the generator and discriminator networks, we require a source of semantic labels about trajectories. These labels can include maneuvers such as merging, turning or slowing down, or interaction patterns such as giving right of way or turning at a four-way-stop junction. For simplicity, these labels may be boolean or unknown values, and they are arranged into a vector $\mathbf{c}$ with elements $c_l \in \{-1, 1, \phi\}$, where ϕ denotes that c_l is unknown or undefined. We stress that for some values of $\mathbf{c}$, in some instances the any choice does not make sense. For example, a labels of "the vehicle is next on a stop sign intersection" and "is vehicle waiting on a red line or not" do not co-exist. This motivates a representation that avoids a single taxonomy of all road situations with definite semantic values.

C. Trajectory Generator

The trajectory generator predicts realistic future vehicle trajectories given inputs of the past trajectories and the map information. It embeds the two inputs before sending them into a long short-term memory (LSTM) network encoder that captures both the spatial and temporal aspect from the inputs. The encoder output is combined with a noise vector generated from a standard normal distribution, and fed into a latent network that separates the information into a high-level vector and a low-level vector. The decoder, taking these two vectors, produces the trajectory samples.

1) Trajectory Network: A series of fully connected layers that embed spatial coordinates into a trajectory embedding vector [5].

2) Map Network: In order to simplify the task of learning to interact with the map, we using the following representation for the lanes. First, we find the nearest point to the vehicle from each lane at the predicting time. Second, we traverse each lane starting at its nearest point to generate an arclength-parameterized curve before computing polynomial coefficients up to second order. Third, we create monomials for the coefficients of the target vehicle using the vehicle velocity and 1,2 sampling time steps – $(v\delta t)^d$ and $(2v\delta t)^d$ for $d = 0, 1, 2$.

Last, we feed the products to allow the encoder and discriminator to learn lane behavior.

3) Encoder: A series of LSTM units process the spatial and map embedding vectors from time steps 1 to t_{obs}. The output is a hidden vector that stores the relevant information up to the current time step.

4) Latent Network: A series of fully connected layers takes the encoder's hidden vector and a noise sample from a standard normal distribution. The outputs are two activation vectors: a vector $z_H \in \mathbb{R}^{d_H}$ that represents high level information such as maneuvers, and a vector $z_L \in \mathbb{R}^{d_L}$ that represents low level information such as vehicle dynamics. To sample efficiently from z_H at test time, d_H is designed to be much smaller than d_L. We train the vectors to be uncorrelated, with z_H matching semantic labels in terms of distances between samples This representation disentangles semantic concepts from low-level trajectory information, in a fashion resembling information bottlenecks [19], but driven by human notions of semantic similarity as learned from the labels.

5) RNN-based decoder: A series of LSTM units takes z_H, z_L, and a map embedding vector, to output a sequence of future vehicle positions.

D. Trajectory Discriminator

An LSTM-based encoder converts the past trajectory and future predictions into a label L = {fake, real}, where fake means a trajectory is generated by our predictor, while real means the trajectory is from data. The structure of the discriminator mirrors that of the trajectory encoder, except in its output dimensionality.

E. Losses

Similar to [7], we measure the performance of our model using the average displacement error (ADE) of Equation 1 and the final displacement error (FDE) of Equation 2.

$$\mathcal{L}_{ADE}(\hat{Y}) = \frac{1}{t_{pred}} \sum_{t=t_{obs}+1}^{t_{obs}+t_{pred}} ||Y_t - \hat{Y}_t||_2 \tag{1}$$

$$\mathcal{L}_{FDE}(\hat{Y}) = ||Y_{t_{obs}+t_{pred}} - \hat{Y}_{t_{obs}+t_{pred}}||_2 \tag{2}$$

1) Best prediction displacement loss: Also as in [7], we compute the Minimum over N (MoN) losses to encourage the model to cover groundtruth options while maintaining diversity in its predictions:

$$\mathcal{L}_{MoN} = \min_n \left(\mathcal{L}_{ADE} \left(\hat{Y}^{(n)} \right) \right), \tag{3}$$

where $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(N)}$ are samples generated by our model. The loss, over N samples from the generator, is computed as the average distance between the best predicted trajectories and acausal trajectories. Although minimizing MoN loss leads to a diluted probability density function compared to the groundtruth [29], we use it to show that our method can estimate an approximate distribution efficiently. We defer a different, more accurate, supervisory cue to future work.

2) Adversarial loss: We use standard binary cross entropy losses, $\mathcal{L}_{GAN,G}, \mathcal{L}_{GAN,D}$, to compute the loss between outputs from the discriminator and the labels. This loss is used to encourage diversity in predictions and is assigned with a higher weight once best prediction displacement loss is reduced to a reasonable scale.

3) Independence loss: The independence loss enforces that the cross-covariance between the two latent vectors $\mathbf{z}_H$ and $\mathbf{z}_L$ remain small, encouraging z_L to hold only low-level information. While this does not guarantee independence of the two, we found this to suffice as regularization.

$$\mathcal{L}_{\textbf{ind}} = \left(\sum_{i=1}^{d_H} \sum_{j=1}^{d_L} z_H^i z_L^j \right)^2. \tag{4}$$

4) Latent space regularization loss: The latent loss regularizes z_H and z_L in terms of their mean and variance and helps to avoid degenerate solutions.

$$\mathcal{L}_{\textbf{lat}} = \begin{array}{l} \|\Sigma_{z_H} - Id\|_F^2 + \|\mu_{z_H}\|_F^2 + \\ \|\Sigma_{z_L} - Id\|^2 + \|\mu_{z_L}\|^2 \end{array}, \tag{5}$$

where $\| \cdot \|_F^2$ denotes the Frobenius norm.

5) Embedding loss: After enforcing $\mathbf{z}_H$ and $\mathbf{z}_L$ are independent vectors, we introduce an embedding loss to enforce the correlation between high-level latent vector $\mathbf{z}_H$ and prediction coding $\mathbf{c}$. Similar to [30], if two data samples have the same answer element for label l, we expect the differences in their high-level latent vectors to be small. On the other hand, if two predictions have different codings, we want to encourage the difference to be large. This can be written as

$$\mathcal{L}_{\textbf{emb}} = \sum_{m=1}^{B} \sum_{n=1}^{B} \sum_{l=1}^{s} \text{sign} \left(c_l^{(m)}, c_l^{(n)} \right) ||\mathbf{v}^{(m)} - \mathbf{v}^{(n)}||_2, \tag{6}$$

where B is batch size, $c_l^{(m)}, c_l^{(n)}$ denote the label l answers on examples m, n respectively, and $\text{sign}(\cdot, \cdot) = 0$ if either argument is ϕ.

6) Total loss: In total, we combine the losses listed above together with appropriate coefficients that are adjusted dynamically during training.

$$\mathcal{L}, \mathcal{D} = \mathcal{L}_{GAN,D} \tag{7}$$

$$\mathcal{L}, \mathcal{G} = \lambda_1 \mathcal{L}_{MoN} + \lambda_2 \mathcal{L}_{GAN,G} + \lambda_3 \mathcal{L}_{\textbf{ind}} + \lambda_4 \mathcal{L}_{\textbf{lat}} + \lambda_5 \mathcal{L}_{\textbf{emb}} \tag{8}$$

F. Sampling Approach

We now describe how we sample from the space of z_H in Alg. 1. We generate a set of latent samples, selecting from them a subset of representatives using the Farthest Point Sampling (FPS) algorithm [31], [32]. We store the nearest representative identity as we compute the distances, to augment the FPS representatives with a weight proportional to their Voronoi cell. This gives us a weighted set of samples that converges to the original distribution, but favors samples from distinct regions of space. FPS allows us to emphasize samples that represent distinct high level maneuvers encoded in z_H.

Algorithm 1 Semantic Sampling

1: **for all** $i = 1..N_{all}$ **do**
2: Sample from $z_{(i)} \sim Z$.
3: Generate latent sample $(z_{H,(i)}, z_{L,(i)})$.
4: **end for**
5: Perform Farthest Point Sampling on $\{\mathbf{z}_{H,(i)}\}$ to obtain N representative samples, $(\mathbf{z}_{H,(j)}, \mathbf{z}_{L,(j)}), j = 1..N$, where (j) denotes a sample of N.
6: Compute Voronoi weights w_j for each sample (j) based on the N samples.
7: Decode from $(\mathbf{z}_{H,(j)}, \mathbf{z}_{L,(j)})$ a full prediction $Y_{(j)}$, store along with weights w_j.
8: Return $\{(Y_{(j)}, w_j)\}_{j=1}^{N}$

The samples cover (in the sense of an ϵ-covering) the space of possible high-level choices. The high level latent space is shaped according to human labels of similarity. With this similarity metric shaping, FPS techniques can leverage its 2-optimal distance coverage property in order to capture the majority of semantically different roll-outs in just a few samples.[1]

III. Results

In this section, we describe the details of our model and dataset, followed by a set of quantitative results against state-of-the-art baselines and qualitative results on diverse prediction.

A. Model Details

The *Trajectory Network* utilizes two stacked linear layers with dimensions of (32, 32). The *Map Network* uses four stacked linear layers with dimensions of (64, 32, 16, 32). An LSTM with one layer and a hidden dimension of 64 forms both the *Encoder* and *Decoder* in the *Trajectory Generator*. The *Latent Network* takes inputs from the Encoder and a noise vector with dimension of 10. This network is composed of two individual linear layers with output dimensions of 3 and 71 for the high-level and low-level layers, respectively. The *Discriminator* is an LSTM with the same structure as the Generator's Encoder, followed by a series of stacked linear layers with dimensions of (64, 16, 1), activated by a sigmoid layer at the end. All linear layers in the Generator are followed by a batch norm, ReLU, and dropout layers. The linear layers in the Discriminator utilize a leakyReLU activation instead. The number of samples n we use for the MoN loss is 5.

The model is implemented in Pytorch and trained on a single NVIDIA Tesla V100 GPU. We use the Argoverse forecasting dataset [13] for training and validation, and select the trained model with the smallest MoN ADE loss on validation set.

B. Semantic Annotations

In order to test our embedding over a large scale dataset, we devised a set of classifiers for the data as surrogates to human annotations. They check for specific high-level trajectory features, and each of them outputs a ternary bit representing whether the feature exists, does not exist, or is unknown, as a k-dimensional vector $\mathbf{c}$ that includes the outputs from all filters. The list of feature filters used in this paper includes: accelerate, decelerate, turn left, turn right, lane follow, lane change, move to left latitudinally, and move to right latitudinally.

C. Quantitative Results

1) Prediction: Over 1 and 3 second prediction horizons, with $N = 5$ samples, we compute the MoN ADE (1) and FDE (2) losses, respectively. In addition to our method, we introduce a few baseline models to demonstrate the prediction accuracy of our method. The first two baselines include a linear Kalman filter with a constant velocity (CV) model and with a constant acceleration (CA) model, respectively. We sample multiple trajectories given the smoothing uncertainties. The third baseline is an LSTM-based encoder decoder model [13], which produces deterministic predictions. In addition, we introduce a few variants of a vanilla GAN-based model taking different input features, where social contains the positions of nearby agents and map contains the nearby lane information as described in II-C2. The results are summarized in Table I. The first two rows indicate that physics-based models can produce predictions with reasonable accuracy. Using only five samples, the CV Kalman Filter outperforms a deterministic deep model with results shown on the third row. The rest of the table shows that a generative adversarial network improve upon accuracy by a large margin compared to physics-based models using five samples. It is observed that the map features contribute more to long horizon predictions. Additionally, our method is competitive compared to standard ones, after regularizing the latent space, while adding sample diversification.

[1]We note that a modified FPS [33] can trade off mode-seeking with coverage-seeking when generating samples.

Model Name	1 Second ADE	1 Second FDE	3 Seconds ADE	3 Seconds FDE
Kalman Filter (CV)	0.51	0.79	1.63	3.62
Kalman Filter (CA)	0.69	1.22	2.87	7.08
LSTM Encoder Decoder	0.57	0.94	1.81	4.13
GAN	0.42	**0.62**	1.55	3.09
GAN+social	0.44	0.66	1.68	3.04
GAN+social+map	0.44	0.63	1.34	2.75
DiversityGAN+social+map	**0.41**	0.65	1.35	2.74
DiversityGAN(FPS)+social+map	0.44	**0.62**	**1.33**	**2.72**

TABLE I: MoN average displacement errors (ADE) and final displacement errors (FDE) of our method and baseline models with $N = 5$ samples.

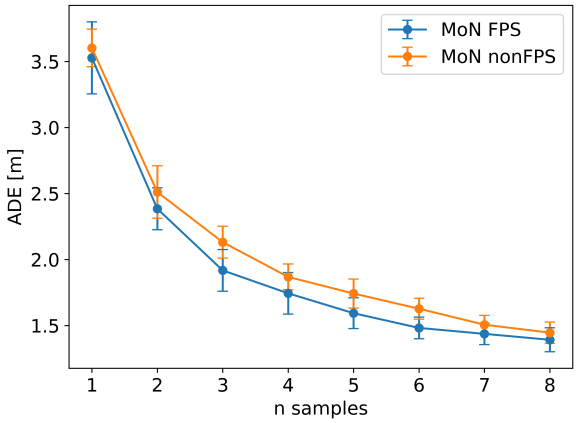
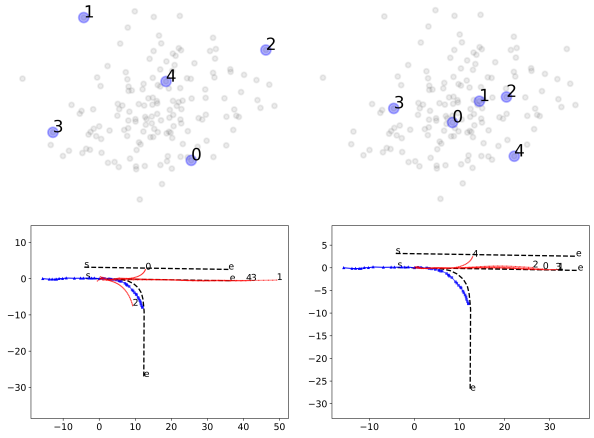


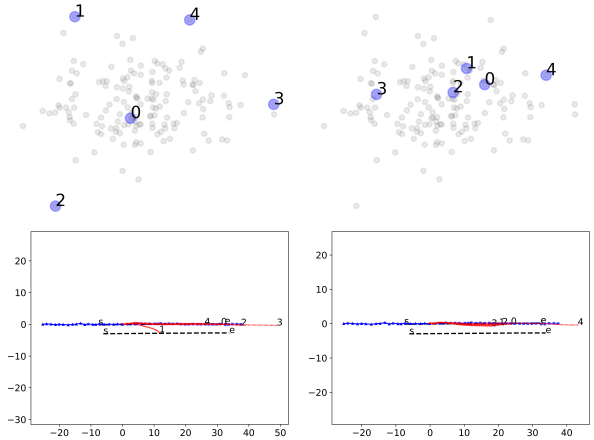
Fig. 3: MoN ADE loss of FPS sampling (blue) and direct sampling (orange) over 3 seconds with N from 1 to 8. The gap between two curves indicates the improvement using FPS, especially when N is from 2 to 6. Error bars represent one standard deviation from five runs with different random seeds.

To show the effectiveness of our latent sampling approach, we measure the MoN loss with and without the FPS method. We test using a challenging subset of the validation dataset that filters out straight driving with constant velocity scenarios, resulting in a trajectory distribution that emphasizes rare events in the data. As indicated in Figure 3, when the number of samples increases, the prediction loss using FPS drops faster compared to direct sampling. We note the improvement is larger in the regime of 2-6 samples, where reasoning about full roll-out of multiple hypotheses is still practical in real-time systems, and we obtain an improvement of 8%. However, beyond the gain in average accuracy, the importance of the method is that it is able to obtain some samples from the additional modes of the distribution of trajectories. We demonstrate the advantage of our methods with a small number of samples in Section III-D.



FPS Direct Sampling

(a) FPS provides accurate coverage of acausal trajectory by generating rare turning samples.



FPS Direct Sampling

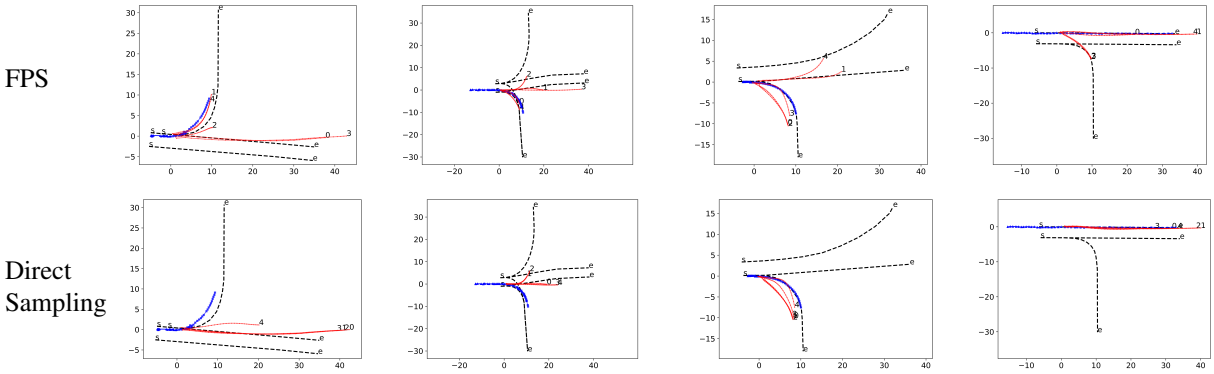
(b) FPS covers a low-likely lane change event that matters for decision making for the ego car.

Fig. 4: Illustrations of how our approach captures rare events by selecting samples that are farther away. The left column highlights the samples selected by FPS and their associated predictions, where $N = 5$. The right column highlights the selected samples using direct sampling and their associated predictions, which cover only high likely events. Blue: observed and acausal trajectories. Red: predicted trajectory samples. Black: lane centers.

D. Qualitative Results

We first show how FPS can be used to improve both prediction accuracy and diversity coverage by illustrating two examples in Figure 4.

In the first example as illustrated in Figure 4(a), our method, as described in Algorithm 1, first generate $N_{all} = 200$ samples in grey, and select $N = 5$ samples



(a) Predicting diversified events helps reduce prediction error in challenging scenarios.

(b) Predicting merging and turning events enables robust and safe decision making for the ego car.

Fig. 5: Predictions of rare events in complicated driving scenarios help improve both accuracy (a) and diversity (b). Top to bottom: FPS and direct sampling with $N = 5$ trajectory samples.

using FPS (highlighted on the left column) and direct sampling (highlighted on the right column) to produce predictions. By selecting samples that are farther away, FPS is able to produce rare events such as right turn, as labelled in 2, that match with the acausal trajectory and thus improve the prediction accuracy. On the other hand, direct sampling tends to sample points from denser regions, which lead to high likelihood events. We show two additional challenging examples in Figure 5(a), where FPS is able to reduce the prediction error by covering turning events when the vehicle is approaching an off-ramp and a full intersection, respectively.

In the second example as illustrated in Figure 4(b), although our method predicts rare events that do not improve displacement losses compared to direct sampling, they are still important for decision making and risk estimation. Although the target vehicle is most likely to go forward, it is useful for our predictor to cover lane change behavior, as labelled in 1, even with a low likelihood, since such prediction could help avoid a possible collision if our ego car is driving on the right lane. Similarly, in the other two examples as shown in Figure 5(b), our method produces events such as merging and turning that are unlikely to happen but are important to consider for robust and safe decision making for the ego car.

IV. Conclusion

We propose a vehicle motion prediction method that caters to both prediction accuracy and diversity. We achieve this by dividing a latent variable into a learned semantic-level part encoding discrete options that the target vehicle can possibly take, and a low-level part encoding other information. The method is demonstrated to achieve state-of-the-art prediction accuracy, while efficiently obtaining trajectory coverage by near-optimal sampling of the high-level latent vector. Future work includes adding more complicated semantic labels such as vehicle interactions, and exploring other sampling methods beyond FPS.

References

[1] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 4363–4369.

[2] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *2012 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2012, pp. 141–146.

[3] X. Huang, S. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9718–9724.

[4] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.

[5] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[6] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.

[7] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[8] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[9] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[10] Y. C. Tang and R. Salakhutdinov, "Multiple futures prediction," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[11] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[12] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 399–404.

[13] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[14] J. Li, H. Ma, and M. Tomizuka, "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6658–6664.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[16] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," *arXiv preprint arXiv:1905.01631*, 2019.

[17] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1179–1184.

[18] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," *arXiv preprint arXiv:1907.10786*, 2019.

[19] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[20] Y. Hristov, D. Angelov, M. Burke, A. Lascarides, and S. Ramamoorthy, "Disentangled relational representations for explaining and learning from demonstration," *arXiv preprint arXiv:1907.13627*, 2019.

[21] T. Unterthiner, B. Nessler, C. Seward, G. Klambauer, M. Heusel, H. Ramsauer, and S. Hochreiter, "Coulomb GANs: provably optimal nash equilibria via potential fields," in *2018 International Conference on Learning Representations (ICLR)*, 2018.

[22] R. Y. Rubinstein, *Combinatorial Optimization, Cross-Entropy, Ants and Rare Events*. Boston, MA: Springer US, 2001, pp. 303–363.

[23] J. Bucklew, *Introduction to rare event simulation*. Springer Science & Business Media, 2013.

[24] M. Koren and M. Kochenderfer, "Efficient autonomy validation in simulation with adaptive stress testing," *arXiv preprint arXiv:1907.06795*, 2019.

[25] M. Koschi, C. Pek, S. Maierhofer, and M. Althoff, "Computationally efficient safety falsification of adaptive cruise control systems," in *2019 IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2019.

[26] M. O'Kelly, A. Sinha, H. Namkoong, J. Duchi, and R. Tedrake, "A scalable risk-based framework for rigorous autonomous vehicle evaluation," 2019.

[27] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2543–2549.

[28] G. Aoude, J. Joseph, N. Roy, and J. How, "Mobile agent trajectory prediction using bayesian nonparametric reachability trees," in *Infotech@ Aerospace 2011*, 2011, p. 1512.

[29] L. A. Thiede and P. P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," *arXiv preprint arXiv:1907.10178*, 2019.

[30] G. Rosman, L. Paull, and D. Rus, "Hybrid control and learning with coresets for autonomous vehicles," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6894–6901.

[31] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293 – 306, 1985.

[32] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Math. Oper. Res.*, vol. 10, no. 2, pp. 180–184, May 1985.

[33] M. Volkov, G. Rosman, D. Feldman, J. W. Fisher, and D. Rus, "Coresets for visual summarization with applications to loop closure," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3638–3645.