Diffusion-based Trajectory Prediction for Vulnerable Road Users (VRUs)

EE260 Introduction to Self-driving Stack
Edison Li, Hung Nguyen, Neda Farahmandi

Abstract

With the development of diffusion models, more and more interesting scenarios are proposed, e.g., image generation. Diffusion models are generative models which can generate new data similar to the data they were trained on. Some researchers have considered generating trajectories using diffusion models. Our motivation is the rising attention on the safety issue of Vulnerable Road User (VRU). VRU trajectory prediction at the intersection is essential for the safety of self-driving cars and should be further considered. Thus, this final project aims to use a diffusion-based model, LEapfrog Diffusion Model for Stochastic Trajectory Prediction (LED) to predict trajectories of VRU at the intersection scenario, explore the effects of traffic signals on the performance of the model, and compare the model's performance with different VRU objects. It is observed that the enhanced LED can generate satisfactory trajectories, and that the trajectories of fast-speed objects are more impacted by the traffic signal.

1. INTRODUCTION

California faces critical challenges in safeguarding vulnerable road users, including pedestrians, cyclists, and motorcyclists, as evidenced by concerning statistics. In 2020 alone, the state recorded 1,013 pedestrian fatalities, alongside 136 cyclist fatalities, according to data from the California Office of Traffic Safety (OTS) [1]. Moreover, the California Highway Patrol (CHP) reported that 1,043 pedestrians and 153 bicyclists were killed during the same period [2].

With the rising attention on the safety issue of VRU, VRU trajectory prediction at the intersection should be further considered. Regarding trajectory prediction, time-sequential methods are suitable for this scenario. Classical methods such as Recurrent Neural Network (RNN), Long short-term memory (LSTM), and Gated Recurrent Unit (GRU) are widely used in trajectory prediction tasks. In addition, Graph Neural Network (GNN) and Transformer are also used for trajectory prediction. Some methods focusing on pedestrian trajectory prediction were proposed based on these methods. Alexandre et al. [3] proposed Social LSTM for pedestrian trajectory prediction in crowded scenarios considering the social interactions among the agents. Similar modification methods are used in Social Attention [4] and Factorized Joint Multi-Agent Motion Prediction (FJMP) [5].

Some researchers are considering using Generative Adversarial Networks (GAN) to generate trajectories. Agrim et al. [6] proposed Social GAN to capture the social interactions during trajectory prediction. With the development of diffusion models, a few researchers consider generating trajectories using diffusion models. Gu et al. [7] proposed motion indeterminacy diffusion (MID), which used a reverse diffusion

process to generate the trajectory from high indeterminacy to low indeterminacy. MID may be the earliest trajectory prediction model that utilizes the diffusion model as part of it. Mao et al. [8] proposed the Leapfrog Diffusion Model (LED) based on MID and improved the efficiency of the denoising process. In the last two years, several new diffusion-based models have been proposed, each of which has its own characteristics. Waymo proposed the MotionDiffuser [9], which learns a highly multimodal distribution that captures diverse future outcomes. Westny et al. [10] proposed a diffusion-based environmentaware trajectory prediction method that can generate realistic trajectories in diverse traffic scenarios considering the interactions. It can also predict the behavior of less cooperative agents in uncertain traffic conditions. SingularTrajectory [11] was proposed to reduce the performance gap of different trajectory prediction tasks, including deterministic, stochastic, domain adaptation, momentary observation, and few-shot.

The diffusion model was used to help unify the pedestrian walking dynamics. In this project, the LED was utilized as the baseline model. This diffusion-based model was used to predict trajectories of VRU at the intersection scenario using SinD dataset [12] and seek for methods to improve the performance of the existing model.

2. MOTIVATION

California faces critical challenges in safeguarding Vulnerable Road Users (VRU), including pedestrians, cyclists, and motorcyclists, as evidenced by concerning statistics. In 2020 alone, the state recorded 1,013 pedestrian fatalities, alongside 136 cyclist fatalities, according to data from the California Office of Traffic Safety (OTS). The California Highway Patrol (CHP) reported 1,043 pedestrians, and 153 bicyclists were killed during the same period.

Self-driving cars play an important role in this context and can improve the road safety or have adverse effects. One approach to make self-driving cars a safer agent on roads is to improve trajectory prediction.

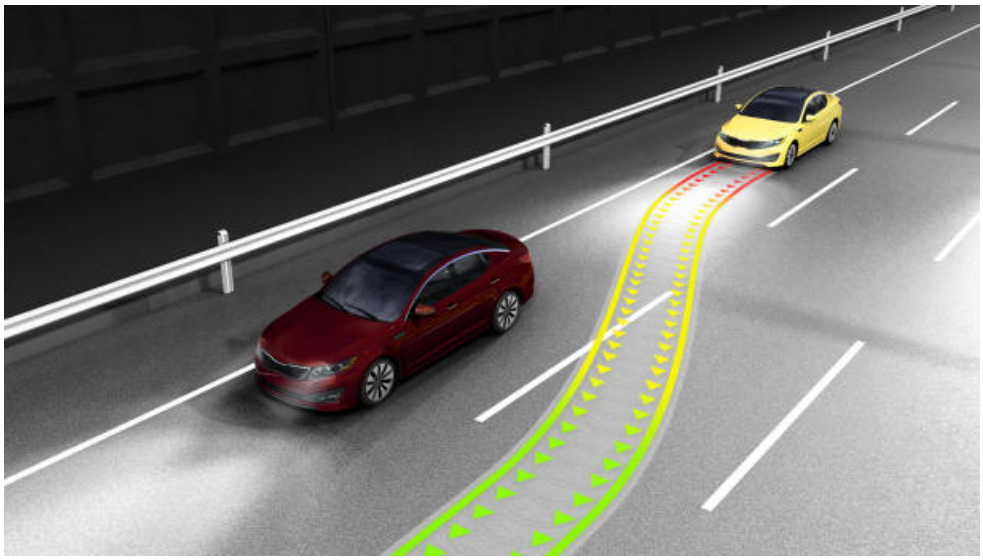

Figure 1. Self-driving car trajectory prediction.

3. RELATED WORK

Force models, RNNs and frequency analysis have been used as deterministic approaches for trajectory prediction. More recently, deep generative models such as Generative Adversarial Network (GAN) structures have been used for generating distributions of multiple future trajectories. Some studies have used Variational Auto-Encoder (VAE) structure to learn the distribution through variational inference.

4. METHODOLOGY

4.1 Diffusion Models

A diffusion model is a type of probabilistic model used in machine learning and statistics to describe how information or quantities spread or diffuse through a system over time. It's based on the concept of random walks, where particles or entities move from one location to another in a stochastic manner. In a diffusion model, the spread of information or quantities is typically described using a diffusion equation, which governs how the distribution of the particle's changes over time. These models are widely used in various fields, including physics, chemistry, biology, finance, and machine learning.

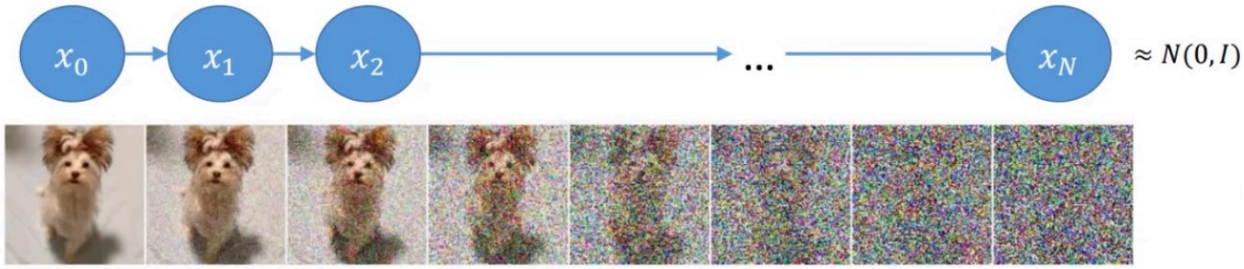

Figure 2. Process of adding noise.

Diffusion models are applied in tasks such as image denoising, data smoothing, signal processing, and generative modeling. They're particularly useful for capturing complex patterns of diffusion in data and making predictions about how information will spread in a system. Examples of diffusion models include Denoising Diffusion Probabilistic Models (DDPM) [13] in image processing and diffusion-based prediction models in finance.

DDPM is the core of diffusion-based generative models. It is a class of generative models that leverage diffusion processes for image generation and denoising. Unlike traditional generative models that directly model the data distribution, DDPM learns to transform a noisy version of an image into a clean version.

The key idea behind DDPM is to iteratively apply a diffusion process to the noise level of an image, gradually reducing the noise until the original clean image is recovered. During training, DDPM learns to estimate the parameters of this diffusion process, enabling them to generate realistic samples and perform image denoising tasks. DDPM has gained attention for its ability to produce high-quality images and effectively handle various types of noise, making it a promising approach for tasks such as image generation, inpainting, and super-resolution.

Fig. 2 shows the forward process of DDPM: Adding noise (labels) to the original image. This is the process of making training images and labels. The noise aligns with Gaussian distribution. For the denoising process, the model predicts the noise and adds it to the noisy image step by step, until it gets the clean image. The structure of the denoising model is shown in Fig. 3. Knowing the state of current time, the noise can be predicted. Similarly, the future trajectory can also be generated by the DDPM model step by step, by training with existing trajectories.

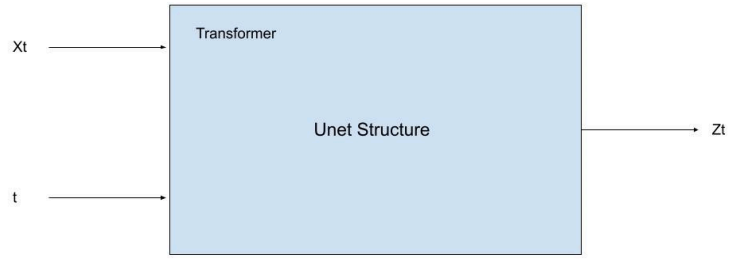

Figure 3. Structure of denoising model.

4.2 LEapfrog Diffusion Model (LED)

The Leapfrog Diffusion Model is developed based on the original DDPM model and provides real-time, precise and diverse trajectory predictions. The main idea is to utilize a trainable leapfrog initializer to directly learn an expressive multi-model distribution of future trajectories. This method jumps from the initialized randomness to the middle step, then starts to denoise from the middle step, which saves a lot of time. This eliminates many denoising steps and accelerates the inference phase. In addition, the initializer is trained to allocate correlated samples appropriately and add diversity to improve the model performance significantly. As depicted in Fig. 4, the blue square represents the adding noise process. The traditional denoising steps are jumped over; thus, only several denoising steps are needed.

About the key innovation, leapfrog initializer, it utilizes the existing past trajectories as input, processing the original coordinates data to get the absolute position, the relative position and the velocity of each agent. It uses social and temporal encoders to get the embedding information and generate the variance, estimation and mean of the trajectory with a multi-layer perceptron decoder. After this, the denoising steps are the same as the normal process.

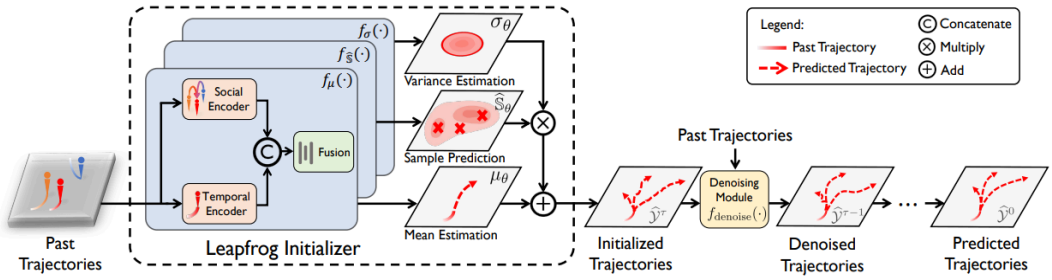

Figure 4. Structure of the leapfrog diffusion model.

LED predicts K initialized trajectories at the τ th denoised step through a trainable leapfrog initializer. After a few denoising steps, LED makes the final predictions. With Leapfrog initializer, LED learns the statistic and generates correlated samples with reparameterization.

The leapfrog diffusion model uses the leapfrog initializer, illustrated in Fig. 5, to estimate the denoised distribution and substitutes a long sequence of traditional denoising steps, accelerating inference and maintaining representation capacity.

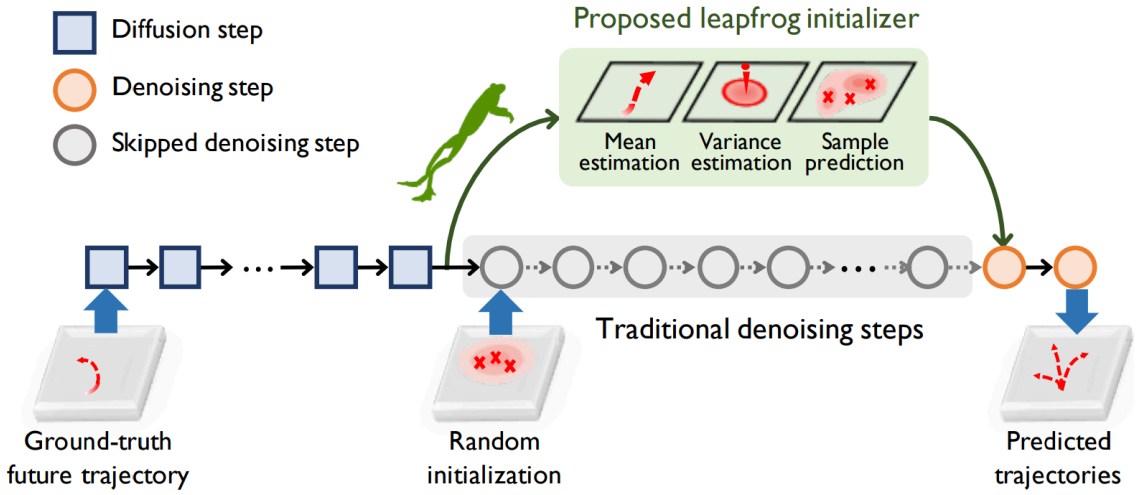

Figure 5. Structure of leapfrog initializer.

4.3 Modifications

In the LED, the authors only encoded the past trajectories and got the position and velocity embedding information. However, for the scenarios at the intersection, the traffic light may be a key factor to the VRU trajectory prediction. Thus, the SinD dataset was utilized to train the enhanced LED model, which contains the trajectory and the synchronized traffic signal timing information. Besides, it collected the trajectories of pedestrians, bicycles, tricycles, buses and vehicles, which gives more space to explore the trajectory characteristics of different kinds of road users and their relevance to the traffic light signals.

To further extract the feature of the past trajectories, another two dimensions were added, which are the accelerations in x and y coordinates, as depicted in Fig. 6. Regarding the traffic light signal, there are five phases, as shown in Table I.

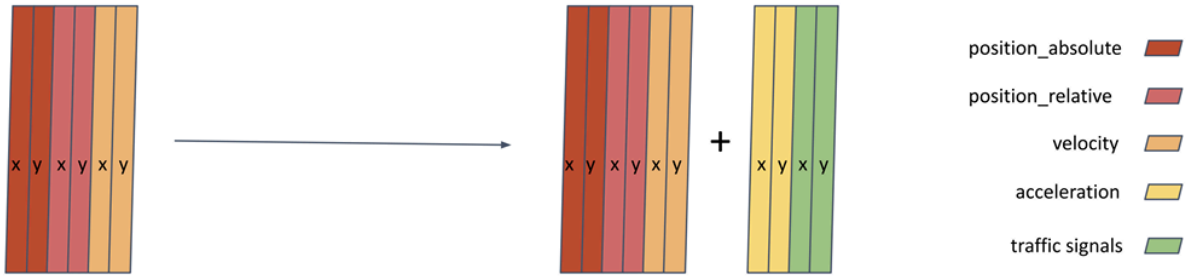

Figure 6. Dimensions of input features.

The signals were unified in south and north to one code and signals in east and west to another code. Thus, two embedded codes will also be added to the feature matrix. For example, Phase1 represents Green (3) in north and south direction, and Red (0) in east and west direction.

An Squeeze-and-Excitation Module has been added to self-enhance the important features, as shown in Fig. 7. In this module, the importance of some dimensions is increased. The fully connected layer is used to predict the importance of each channel. The output of the excitation is multiplied by each original channel to achieve recalibration of the original channel.

Table I. Traffic light signal codes.

Traffic Signal	**Direction**			
	North	**South**	**East**	**West**
Phase 1	3	3	0	0
Phase 2	1	1	0	0
Phase 3	0	0	0	0
Phase 4	0	0	3	3
Phase 5	0	0	1	1

*0: Red; 1: Yellow; 3: Green.

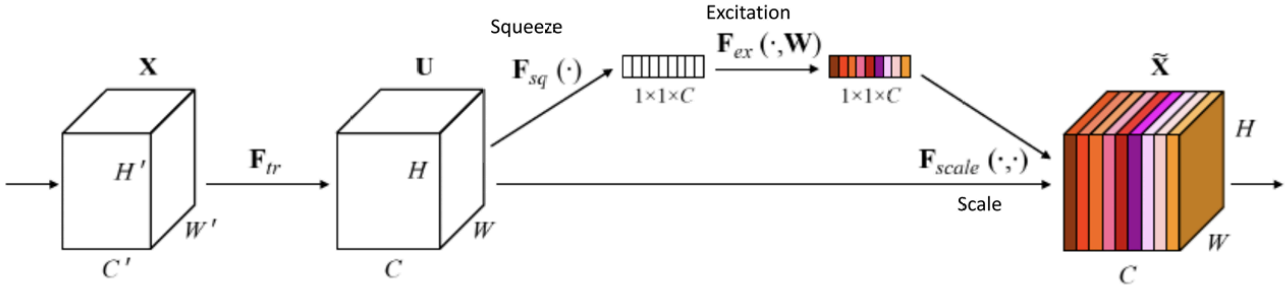

Figure 7. Squeeze-and-Excitation Module.

5. EXPERIMENTS

5.1 Dataset

SinD Dataset was used to retrain and evaluate the enhanced LED model. This dataset includes VRUs such as pedestrians, bicycles, and tricycles. SinD dataset Provides Lanelet2 format map of the signalized intersection. Data visualization is illustrated in Fig. 8 and Fig. 9. The SinD dataset contains 191888 data samples for training and 47972 data samples for evaluation. We also showed the recorded result data structure in Table II.

Table II. Data recording format of the results.

File	Source in Code	Content Description
past.pt	*data['pre_motion_3D']*	3D motion data representing the past trajectories.
future.pt	*data['fut_motion_3D']*	3D motion data representing the future ground truth trajectories.
p_var.pt	*sample_prediction*	Normalized prediction samples scaled by variance and mean.
p_mean.pt	*mean_estimation*	Mean predictions for the trajectories.
p_sigma.pt	*variance_estimation*	Variance estimation for the predictions.
prediction.pt	*pred_traj*	Predicted trajectories sampled from the probabilistic model.
p_mean_denoise.pt	*pred_mean*	Mean trajectories generated using the denoised process.

We have extracted the data for pedestrians, bicycles, tricycles and motorcycles from SinD dataset and saved them as SinD_7_train_light_p.npy, SinD_5_train_light_b.npy, SinD_20_train_light_t.npy, SinD_3_train_light_m.npy. The .npy data files can be loaded using the dataloader_nba.py. The VRU motion state is visualized in Fig.10.

The processed data format is aligned with the NBA dataset, which contains 11 agents' continuous 30-frames-length trajectories in a period of time. For the SinD dataset, blank data has been added into the .npy, due to the fluctuation in the number of the agents in one frame. The number of the agents will be the largest one that appears during the recording period.

Figure 8. Examples in SinD dataset.

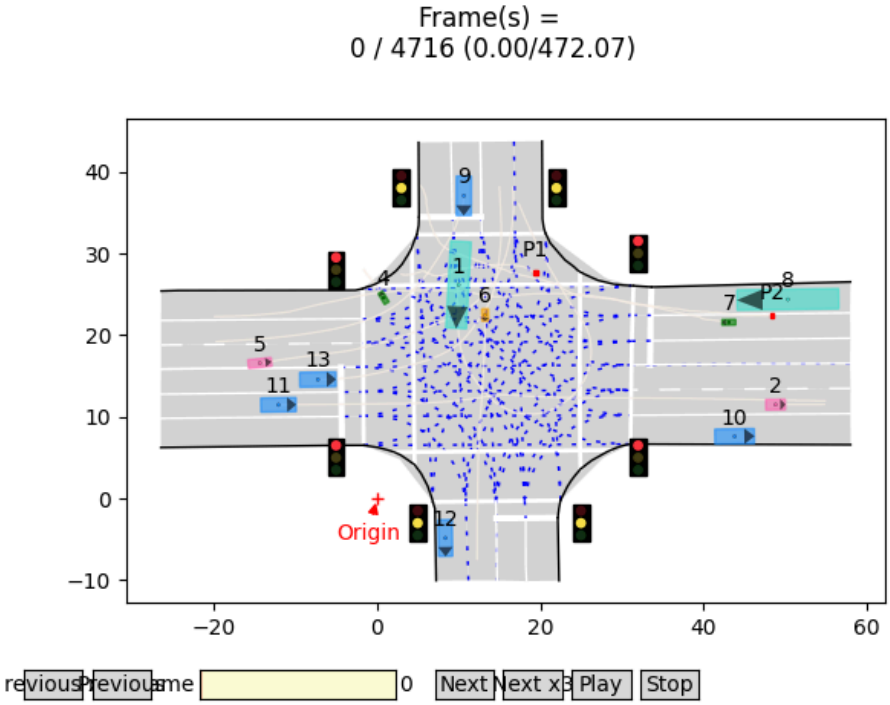

Figure 9. Visualization of the dataset.

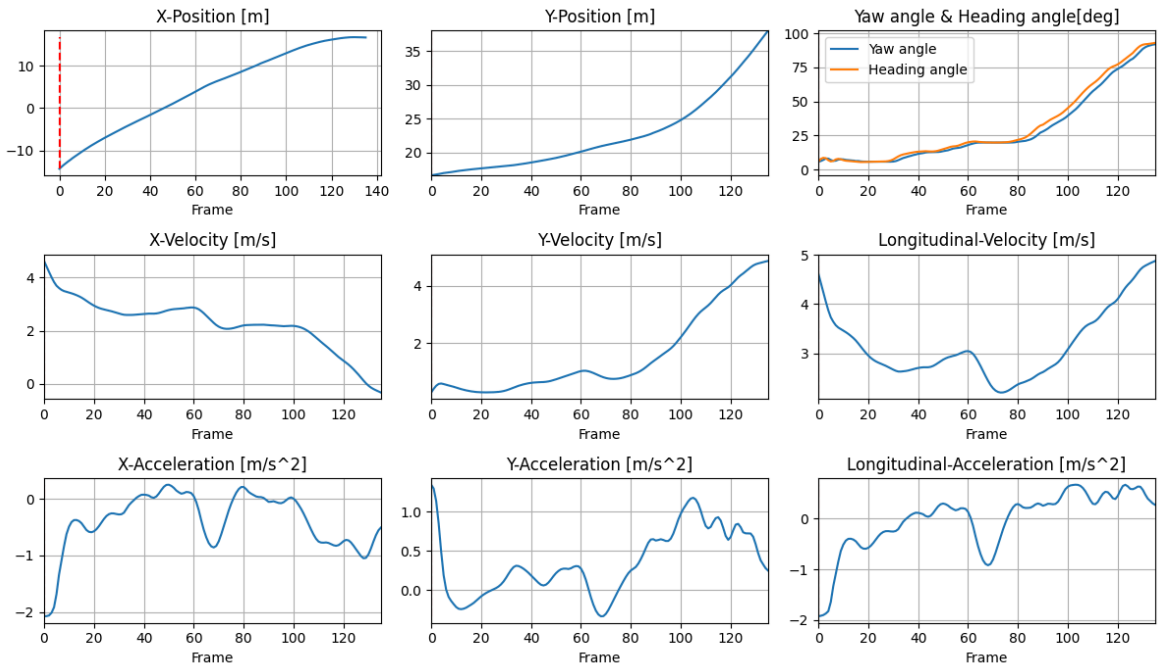

Figure 10. Motion states of VRUs in the SinD dataset.

5.2 Training

There are 2 phases of training: diffusion model and initializer. During training, 10 past frames of trajectories are used to predict the next 20 frames of future trajectories. Training of the model was object-class based, in which, model was trained based on different classes of VRUs within the SinD dataset.

5.3 Model Performance

With the mentioned dataset and the modified model, the models were trained for different objects, respectively. 10 frames past trajectory are used to predict 20 frames' trajectory in the future. All the same kinds of agents in one frame are trained and tested at the same time. Examples of predicted trajectory visualization are illustrated in Figure 11.

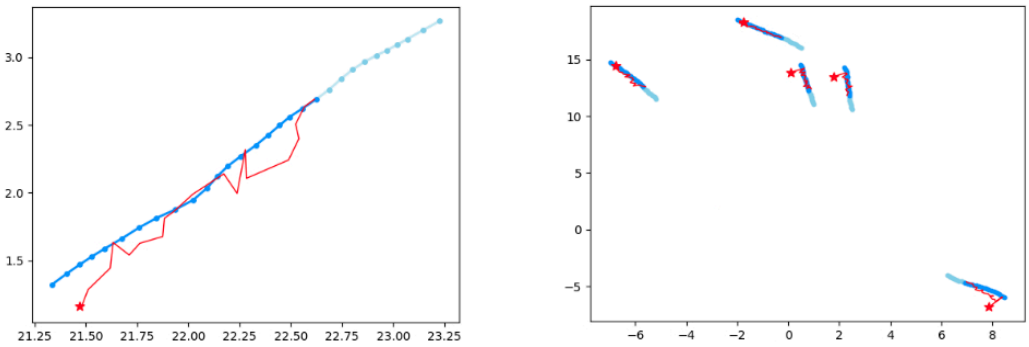

Figure 11. Visualization of predicted trajectory.

Regarding the evaluation metrics, ADE and FDE are used in this project. ADE calculates the minimum time-averaged distance among predictions and the ground-truth future trajectory; FDE measures the minimum distance among the predicted endpoints and the ground-truth endpoints.

For pedestrians, the model converges at around 90 epochs. As can be seen in Table III, the performance of the new trained model with modifications is worse. The reason may be the low relevance between pedestrians and traffic signals. Because pedestrians walk slowly, trajectories affected by traffic lights account for a minority of the training data, and 20 frames' data cannot fully represent the variation of the pedestrian at different light signals.

Table III. ADE and FDE of pedestrians.

ADE/FDE	**Time**			
	5 Frames	**10 Frames**	**15 Frames**	**20 Frames**
ADE (original)	0.0294	0.0351	0.0430	0.0527
ADE (modified)	0.0283	0.0371	0.0463	0.0564
FDE (original)	0.0318	0.0370	0.0582	0.0844
FDE (modified)	0.0308	0.0442	0.0621	0.0908

Similarly, the models of bicycles and tricycles were trained and tested in the same way, the results are shown in Table IV and Table V. As the velocity of the bicycle is much faster than pedestrians, the modified model is a little bit better than the original model, especially in FDE comparison. The modified model also performs better in the FDE metric.

Table IV. ADE and FDE of bicycles.

ADE/FDE	**Time**			
	5 Frames	**10 Frames**	**15 Frames**	**20 Frames**
ADE (original)	0.0153	0.0203	0.0256	0.0320
ADE (modified)	0.0161	0.0208	0.0261	0.0320
FDE (original)	0.0148	0.0221	0.0320	0.0461
FDE (modified)	0.0157	0.0225	0.0314	0.0441

Table V. ADE and FDE of tricycles.

ADE/FDE	Time			
	5 Frames	10 Frames	15 Frames	20 Frames
ADE (original)	0.0157	0.0179	0.0202	0.0233
ADE (modified)	0.0152	0.0171	0.0194	0.0233
FDE (original)	0.0147	0.0178	0.0224	0.0293
FDE (modified)	0.0130	0.0165	0.0223	0.0289

To further test the affection of the velocity on the performance of the modified model, the motorcycle category was tested. The ADE performance of the modified model improves 5.37% and FDE improves 7.87%, respectively, which proves the relevance of the traffic signal and the velocity of the agents. However, the performance of the motorcycle is worse than other slow VRUs, which is a normal phenomenon because high-speed agents have more possible variations within the same 20 frames of the future trajectories.

Table VI. ADE and FDE of motorcycles.

ADE/FDE	Time			
	5 Frames	10 Frames	15 Frames	20 Frames
ADE (original)	0.0279	0.0366	0.0494	0.0652
ADE (modified)	0.0251	0.0340	0.0465	0.0617
FDE (original)	0.0288	0.0492	0.0808	0.1232
FDE (modified)	0.0265	0.0446	0.0756	0.1135

6. CONCLUSIONS

With training and testing the LED model in the SinD dataset with some encoding modifications, this project explores the influence of the traffic light signals on the performance of the modified model. After evaluation, the modified model performs better and better, as the velocity of the traffic agents rises. Thus, modified models with traffic signals are more suitable for short-term prediction of the motorized agents or long-term prediction of the VRUs.

In the future, more work should be done to investigate the mechanism of this diffusion-based model considering the traffic signals, and the performance of the prediction of the VRU and motorized road users should be both considered.

7. ACKNOWLEDGEMENTS

Special thanks to the authors of the LED model and SinD dataset. The modifications are deployed based on their open-source LED code at https://github.com/MediaBrain-SJTU/LED and SinD dataset at https://github.com/SOTIF-AVLab/SinD.

REFERENCES:

1. "California Traffic Safety Quick Stats — Office of Traffic Safety," Ca.gov, 2021. https://www.ots.ca.gov/ots-and-traffic-safety/score-card/
2. "SWITRS-2020-Report,"www.chp.ca.gov.https://www.chp.ca.gov/programs-services/services-information/switrsinternet-statewide-integrated-traffic-records-system/switrs-2020-report
3. Alexandre Alahi, K. Goel, V. Ramanathan, Alexandre Robicquet, L. FeiFei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," Computer Vision and Pattern Recognition, Jun. 2016, doi: https://doi.org/10.1109/cvpr.2016.110.
4. Anirudh Vemula, K. Muelling, and J. Oh, "Social Attention: Modeling Attention in Human Crowds," International Conference on Robotics and Automation, May 2018, doi: https://doi.org/10.1109/icra.2018.8460504.
5. L. Rowe, M. Ethier, Eli-Henry Dykhne, and K. Czarnecki, "FJMP: Factorized Joint Multi-Agent Motion Prediction over Learned Directed Acyclic Interaction Graphs," Jun. 2023, doi: https://doi.org/10.1109/cvpr52729.2023.01321.
6. A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018, doi: https://doi.org/10.1109/cvpr.2018.00240.
7. T. Gu et al., "Stochastic Trajectory Prediction via Motion Indeterminacy Diffusion," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2022, doi: https://doi.org/10.1109/cvpr52688.2022.01660.
8. W. Mao, C. Xu, Q. Zhu, S. Chen, and Y. Wang, "Leapfrog Diffusion Model for Stochastic Trajectory Prediction," Jun. 2023, doi: https://doi.org/10.1109/cvpr52729.2023.00534.
9. Chiyu "Max" Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, and Dragomir Anguelov, "MotionDiffuser: Controllable MultiAgent Motion Prediction Using Diffusion," Jun. 2023, doi: https://doi.org/10.1109/cvpr52729.2023.00930.
10. T. Westny, B. Olofsson, and E. Frisk, "Diffusion-Based Environment-Aware Trajectory Prediction," arXiv.org, Mar. 18, 2024. https://arxiv.org/abs/2403.11643 (accessed Jun. 14, 2024).
11. I. Bae, Y.-J. Park, and H.-G. Jeon, "SingularTrajectory: Universal Trajectory Predictor Using Diffusion Model", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 17890–17901.
12. Y. Xu et al., "SIND: A Drone Dataset at Signalized Intersection in China," 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Oct. 2022, doi: https://doi.org/10.1109/itsc55140.2022.9921959.
13. J. Ho, A. Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models.," neural information processing systems, vol. 33, pp.6840–6851, Jan. 2020.