WSiP: Wave Superposition Inspired Pooling for Dynamic Interactions-Aware Trajectory Prediction

Renzhi Wang[1], **Senzhang Wang**[1*], **Hao Yan**[1], **Xiang Wang**[2]
[1]Central South University
[2]National University of Defense Technology
rzwang516@gmail.com, szwang@csu.edu.cn,
CSUyh1999@csu.edu.cn, xiangwangcn@nudt.edu.cn

Abstract

Predicting motions of surrounding vehicles is critically important to help autonomous driving systems plan a safe path and avoid collisions. Although recent social pooling based LSTM models have achieved significant performance gains by considering the motion interactions between vehicles close to each other, vehicle trajectory prediction still remains as a challenging research issue due to the dynamic and high-order interactions in the real complex driving scenarios. To this end, we propose a wave superposition inspired social pooling (Wave-pooling for short) method for dynamically aggregating the high-order interactions from both local and global neighbor vehicles. Through modeling each vehicle as a wave with the amplitude and phase, Wave-pooling can more effectively represent the dynamic motion states of vehicles and capture their high-order dynamic interactions by wave superposition. By integrating Wave-pooling, an encoder-decoder based learning framework named WSiP is also proposed. Extensive experiments conducted on two public highway datasets NGSIM and highD verify the effectiveness of WSiP by comparison with current state-of-the-art baselines. More importantly, the result of WSiP is more interpretable as the interaction strength between vehicles can be intuitively reflected by their phase difference. The code of the work is publicly available at https://github.com/Chopin0123/WSiP.

Introduction

Predicting trajectories of moving vehicles (also called agents in this paper) is one of significant tasks in autonomous driving. During driving, an autonomous vehicle usually makes decisions according to its surrounding traffic situations. For example, as shown in Figure 1, vehicle A will first perceive surrounding traffic environment, and then predict the possible motions of its neighboring vehicles (e.g. vehicle B). This will help A plan a safe path for avoiding potential collisions. Accurately predicting the trajectories of vehicles is critically important to infer the intended motions of adjacent agents (Tang and Salakhutdinov 2019; Liu et al. 2021; Wang, Cao, and Yu 2022).

Conventional models predict the motion of a single vehicle, which are mostly based on statistical models such as Kalman Filters (Kalman 1960), Hidden Markov Model (Firl et al. 2012), and Gaussian Processes (Williams and Rasmussen 2006). These models regard vehicles as independent motion entities governed by physical laws or controlled by human drivers' intentions. However, traffic agents are highly interactive and will affect the motions of each other in real driving scenarios, which is called social interaction (Alahi et al. 2016; Mohamed et al. 2020). These models ignore the complex social interactions and usually cause large prediction bias. With the success of Long Short-Term memory (LSTM) networks in modeling non-linear times series data, recent works have adopted them to predict trajectories. Alahi et al. (Alahi et al. 2016) propose a Social LSTM model to predict pedestrian trajectories. Social LSTM for the first time introduces social pooling, which models the social interactions between pedestrians through letting neighboring pedestrians share the motion states. Following Social LSTM, CS-LSTM (Deo and Trivedi 2018) uses convolutional social pooling layers for robustly modeling the inter-dependencies in vehicle motions. Gupta et al. (Gupta et al. 2018) further propose a GAN-based encoder-decoder framework called Social GAN, which introduces a pooling module built upon MLP and max pooling to model the interactions among people in a scene.

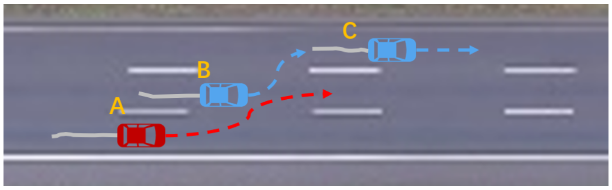


Figure 1: Illustration of high-order interactions between vehicles. Agents A and B both intend to merge to the left lane. Agent C suddenly makes a braking operation, causing agent B to be unable to change lanes. Thus the motion states of C indirectly cause A to go straight ahead.

Although social pooling based methods can better capture the interactions among traffic agents and achieve significant performance gains, they still suffer from the following three limitations. First, existing social pooling methods are not effective to capture the dynamics of interactions among vehicles. For example, as shown in Figure 1, agent A intends to merge to the left lane. If agent B merges to the left lane

[*] Corresponding author

first, A can change lanes safely. However, if B keeps driving straight ahead, A probably will merge to the left lane later considering the high probability of collision with B. The interactions between A and B are different in the two cases. Existing social pooling strategies generally learn a fixed interaction weight between A and B, and thus cannot effectively capture their dynamic motion states. Second, existing methods only focus on the interactions between the target vehicle and its local neighbors. The high-order interactions between the target vehicle and the ones far away are ignored. For example, as shown in Figure 1, although agent C is not close to A, it can still influence the future trajectory of A because its sudden braking operation makes B unable to merge to the left lane. Third, anticipation on the motion states of nearby agents in the near future can also help predict the trajectory of the target agent. For example, if the model can forecast that agent B will not change lanes with a high probability, we can predict that A will probably keep going straight. How to effectively utilize the anticipation on the future motion states of nearby agents to help predict the trajectory of the target agent is not fully explored, either.

To address the above limitations, we propose an encoder-decoder based learning framework WSiP coupled with a novel wave superposition inspired pooling (Wave-pooling) method to more effectively capture the dynamic and high-order interactions among vehicles. Specifically, WSiP consists of a historical interaction modulation module, a future trajectory simulation modulation module and a trajectory prediction module. The historical modulation interaction module models the high-order interactions among vehicles, including the agents in both adjacent and non-adjacent lanes. The future trajectory simulation module simulates a short-term driving scenario by anticipating the future motions of the neighboring agents. The trajectory prediction module generates the future trajectory of the target agent. The first two modules use Wave-pooling to capture the inter-dependencies among vehicles. Wave-pooling is motivated by Wave-MLP (Tang et al. 2022), which for the first time models each token of an image as a wave to better capture their semantic correlations. The amplitude represents the content of each token, while the phase modulates the relationship between tokens. Inspired by Wave-MLP, Wave-pooling models each vehicle in a scene as a wave with a particular amplitude and phase. The amplitude reflects the dynamics of each agent, and the phase modulates the interaction between agents. The dynamic interactions between agents can be reflected by the superposition of their corresponding waves. The primary contributions of this work are summarized as follows.

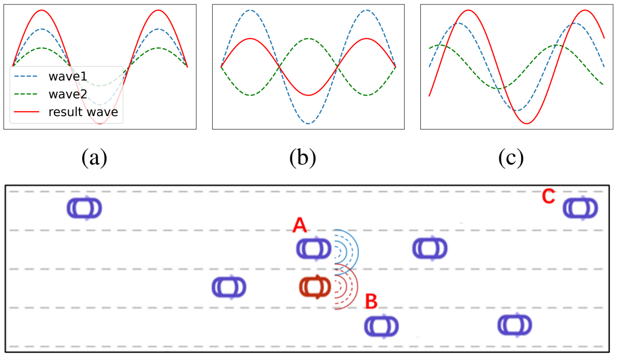
- We for the first time model the social interactions between vehicles as a wave superposition process and propose Wave-pooling mechanism to dynamically aggregate the interactions between agents.
- An encoder-decoder based learning framework WSiP is proposed, which contains a historical interaction modulation module, a future trajectory simulation module and a trajectory prediction module to more accurately predict the vehicle trajectories.
- Extensively evaluations over two real datasets verify the effectiveness of our proposal by comparison with SOTA baselines. The result also shows WSiP is more interpretable as the phase difference can intuitively reflect the interaction strength between the agents.

Related Work

The problem of predicting trajectories has been extensively studied. Existing traditional methods are mostly based on Kalman Filters (Kalman 1960), Logistic regression (Klingelschmitt et al. 2014), Support Vector Machine (SVM) (Aoude et al. 2010), Hidden Markov Model (HMM) (Firl et al. 2012) and Bayesian Networks (Lefèvre, Laugier, and Ibañez-Guzmán 2011), etc. They regard vehicles as independent motion entities. For complex interactive situations they may lead to incorrect estimations of potential threats due to non-consideration of interactions between traffic agents.

Recently, deep learning models such as RNNs and LSTM have been introduced to address the problem of trajectory prediction (Wang et al. 2017; Nawaz et al. 2020). Altché and de La Fortelle (Altché and de La Fortelle 2017) use LSTM to predict longitudinal velocities of vehicles on the highway with the information of local neighbors around the target vehicle. GAIL-GRU (Kuefler et al. 2017) combines the physics-based model and GRU to model human driving behaviors on highways. Some approaches (Kim et al. 2017; Park et al. 2018) generate future locations on the occupancy grid map based on the encoder-decoder framework. One major limitation of RNN and LSTM based models above is that they ignore the interactions between agents.

Agent interactions can be categorized into agent-agent interaction and agent-scene interaction. The agent-scene interaction models how agents interact with static scenes, such as road structures. The high definition (HD) maps are essential for agent-scene interaction modeling (Gao et al. 2020; Liang et al. 2020). Our work focuses on agent-agent interaction. To achieve accurate predictions, the model is required to capture the social interaction between agents. The Social Force model (Helbing and Molnar 1995) is a pioneer work, which predicts pedestrian motions by using a system of forces describing social interactions. There are also some interaction-aware methods based on Dynamic Bayesian Networks (Agamennoni, Nieto, and Nebot 2011; Liebner et al. 2012). However, capturing inter-dependencies of agents involves a large number of latent variables. Therefore, these traditional models could only be used for simple interaction scenarios (e.g. sparse vehicles on roads). Social LSTM (Alahi et al. 2016) is a seminal work introducing social pooling, which shares hidden states between neighboring LSTMs based on social tensor so that it can automatically learn some typical interactions in crowds. Afterwards, a series of studies follow the social pooling mechanism. CS-LSTM (Deo and Trivedi 2018) and PiP (Song et al. 2020) apply convolutional and max pooling layers for modeling the spatial interactions. Social GAN (Gupta et al. 2018) proposes a novel pooling strategy enabling the network to learn social norms in a purely data-driven approach. However, these pooling methods are still



(a) (b) (c)

Figure 2: Upper: The superposition of two waves: (a) constructive interference, (b) destructive interference, and (c) the general case. Lower: The interactions between the target agent B and other agents.

not effective to capture the dynamics of interactions in real driving scenarios.

Problem Definition and Preliminaries

Problem Definition

We formulate the trajectory prediction problem as a sequence generation problem which generates future locations of the target vehicle over a future time horizon T_f according to the historical trajectories of surrounding vehicles and the target itself in the past T_h time steps. We denote the historical trajectory of a vehicle agent a_i at time step t as $\mathbf{X}_i = \left\{X_i^{t-T_h}, \ldots, X_i^{t-1}, X_i^t\right\}$, where $X_i^t = (x_i^t, y_i^t)$ is a 2D coordinate pair. Assuming there are N vehicles in a scene, the input to our model are coordinate sequences $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{tar}, \ldots, \mathbf{X}_N\}$. Our goal is to predict the discrete future positions of the target vehicle $\mathbf{Y}_{tar} = \left\{Y_{tar}^{t+1}, Y_{tar}^{t+2}, \ldots, Y_{tar}^{t+T_f}\right\}$, where Y_i^t is also a 2D coordinate pair.

Preliminaries

When two or more waves cross at a point, it follows the principle of wave superposition. That is, the displacement of the resultant wave at some point is the vector sum of the individual displacements produced by each of the waves at that point. Motivated by Wave-MLP, the wave-like representation can be formulated as

$$\tilde{\boldsymbol{z}}_j = |\boldsymbol{z}_j| \odot e^{i\boldsymbol{\theta}_j}, j = 1, 2, \cdots, n, \tag{1}$$

where i is the imaginary unit, and $\odot$ is the element-wise multiplication. $|\boldsymbol{z}_j|$ denotes amplitude and $\boldsymbol{\theta}_j$ denotes phase. The superposition of two waves $\tilde{\boldsymbol{z}}_i$ and $\tilde{\boldsymbol{z}}_j$ is modulated by their phases. Supposing $\tilde{\boldsymbol{z}}_r$ is the resultant wave of $\tilde{\boldsymbol{z}}_i$ and $\tilde{\boldsymbol{z}}_j$, the amplitude $|\boldsymbol{z}_r|$ of $\tilde{\boldsymbol{z}}_r$ can be calculated as

$$|\boldsymbol{z}_r| = \sqrt{|\boldsymbol{z}_i|^2 + |\boldsymbol{z}_j|^2 + 2\,|\boldsymbol{z}_i| \odot |\boldsymbol{z}_j| \odot \cos\left(\boldsymbol{\theta}_j - \boldsymbol{\theta}_i\right)}. \tag{2}$$

Wave interference is a typical example following this principle. The superposition result is intuitively shown in Figure 2. Constructive interference (Figure 2(a)) occurs when two waves $\tilde{\boldsymbol{z}}_i$ and $\tilde{\boldsymbol{z}}_j$ have the same displacement in the same direction at any point, i.e. $\boldsymbol{\theta}_i = \boldsymbol{\theta}_j + 2m * \pi, m \in [0, \pm 1, \pm 2, \cdots]$. Destructive interference (Figure 2(b)) occurs when two waves are displaced in opposite directions at any point, i.e. $\boldsymbol{\theta}_i = \boldsymbol{\theta}_j + (2 * m - 1) * \pi, m \in [0, \pm 1, \pm 2, \cdots]$. In short, whether superposition amplitude is enhanced or weakened depends on phase difference.

According to quantum mechanics, all matter exhibits wave-like behaviors and generates matter waves (Thomson and Reid 1927). Wave-MLP (Tang et al. 2022) makes the first attempt to represent each token in an image as a wave with both amplitude and phase information. The phase modulates the aggregating process of different tokens according to their semantic content. Inspired by Wave-MLP, we attempt to represent each vehicle in a scene as a wave with the amplitude and phase. As shown in the lower part of Figure 2, agent A is closest to the target. They may have the highest degree of interactions, which could be regarded as constructive interference (Figure 2(a)). Agent C is far away from the target, so it may have less influence on the target. The interactions between the target and agents like B are more general, which is similar to the situation in Figure 2(c). We expect the phase of a vehicle to dynamically modulate the interactions between agents similar to wave superposition.

Methodology

The model framework is depicted in Figure 3. WSiP is based on an encoder-decoder framework, which contains three modules, historical interaction modulation module, future trajectory simulation module and trajectory prediction module. The first two modules (encoder) capture the inter-dependencies of agents for historical and future time horizons respectively. The trajectory prediction module (decoder) generates the distribution of future locations of the target agent. Next we will introduce three modules in detail.

Historical Interaction Modulation Module

This module consists of historical trajectory encoder and Wave-pooling. We first encode historical trajectories to learn the motion states of agents. Following previous methods (Deo and Trivedi 2018; Song et al. 2020), we assume neighbors of the target vehicle are within ± 90 feet in the longitude direction and within the four adjacent lanes centered on the target vehicle as shown in the left part of Figure 3. The historical trajectory encoder will encode trajectory sequences of the target and its neighbors. MLP is first applied to convert these sequences into the motion embeddings, which are then passed through LSTM networks. For agent a_i, its final hidden state of LSTM $\boldsymbol{h}_i$ is expected to reflect its historical motion states. Next, these hidden states form a target-centric social tensor according to their spatial locations. Additionally, the hidden state of the target is passed through a fully connected layer to obtain dynamics encoding $\boldsymbol{d}_{tar}$. Note that LSTM weights are shared across all the vehicles.

Then we apply Wave-pooling to learn inter-dependencies of vehicles. We represent each agent in a scene as a wave with the amplitude and phase. For agent a_i, the amplitude

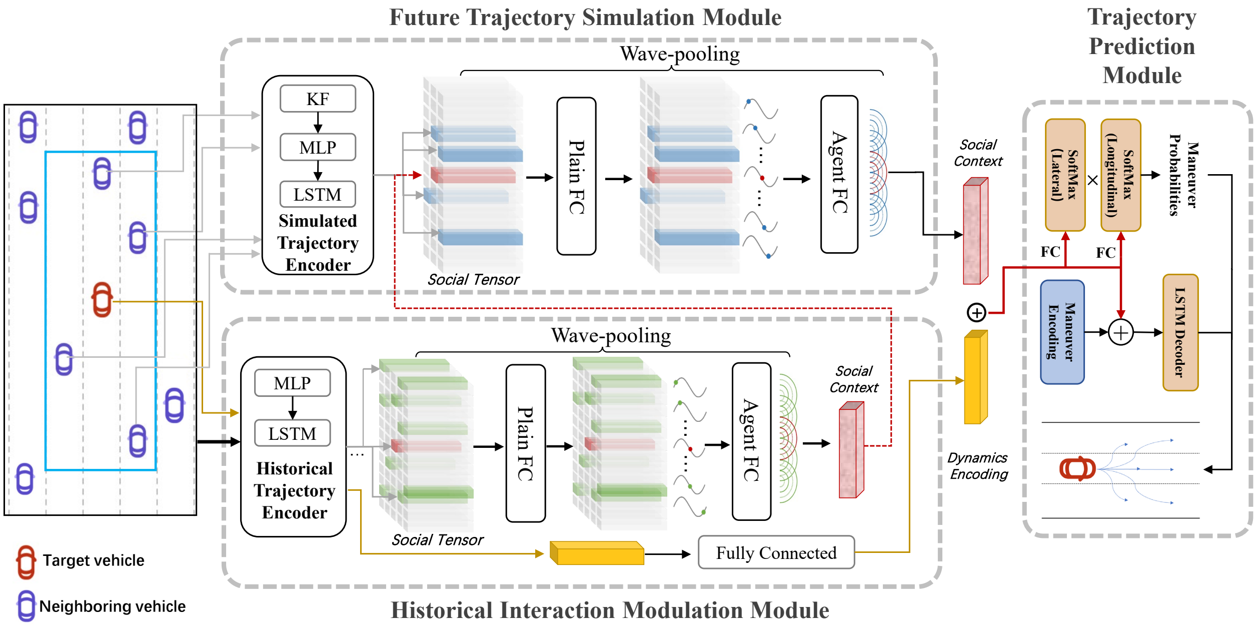


Figure 3: The framework of WSiP, which contains historical interaction modulation module, future trajectory simulation module and trajectory prediction module. Historical interaction modulation module and future trajectory simulation module adopt Wave-pooling to capture the historical and future inter-dependencies of agents separately. The trajectory prediction module outputs a multi-modal distribution of the future locations for the target agent.

embedding $\boldsymbol{z}_i$ is obtained by feeding its final hidden state $\boldsymbol{h}_i$ to a plain fully connected layer (Plain-FC). Therefore, amplitude could represent the dynamics of the vehicle. Phase is used to modulate the aggregation of information from neighboring agents. In order to obtain the phase $\boldsymbol{\theta}_i$ according to motion states of agent a_i, we also apply Plain-FC to learn phase embedding. Thus, the phase $\boldsymbol{\theta}_i$ adjusts dynamically according to the motion states of an agent. The amplitude and phase embeddings of agent a_i are calculated by

$$\boldsymbol{z}_i = \text{Plain-FC}\ \left(\boldsymbol{h}_i, W^z\right) = W^z \boldsymbol{h}_i, \tag{3}$$

$$\boldsymbol{\theta}_i = \text{Plain-FC}\ \left(\boldsymbol{h}_i, W^\theta\right) = W^\theta \boldsymbol{h}_i, \tag{4}$$

where W^z and W^θ are learnable weights. In MLP-based vision architecture (Tolstikhin et al. 2021; Touvron et al. 2021), token mixing layer allows communication from different tokens and aggregates information of tokens. We describe token mixing operation in this work as Agent-FC, which aggregates different vehicle interactions. Suppose there are n vehicle waves denoted as $\tilde{\boldsymbol{Z}} = [\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{z}}_2, \cdots, \tilde{\boldsymbol{z}}_n]$. Agent-FC operation can be written as:

$$\boldsymbol{o}_j = \text{Agent-FC}\left(\tilde{\boldsymbol{Z}}, W^t\right)_j = \sum_k W^t_{jk} \odot \tilde{\boldsymbol{z}}_k, \quad j = 1, 2, \cdots n, \tag{5}$$

where W^t is learnable weights, $\odot$ is the element-wise multiplication and j indicates the j-th output representation. In the extreme case, Agent-FC can be regarded as depth-wise convolutions of a full receptive field and parameter sharing (Tolstikhin et al. 2021). We employ Agent-FC to aggregate wave-like representations of all agents in a scene as shown in Figure 3. With the wave function unfolded by Euler's formula and the common quantum measurement methods, the output of Agent-FC can be calculated as follows

$$\boldsymbol{o}_j = \sum_k W^t_{jk} \boldsymbol{z}_k \odot \cos \boldsymbol{\theta}_k + W^i_{jk} \boldsymbol{z}_k \odot \sin \boldsymbol{\theta}_k, \quad j = 1, 2, \cdots, n. \tag{6}$$

where W^t and W^i are both learnable weights. $\boldsymbol{o}_j$ is the j-th output. In practice, $\boldsymbol{z}_i \odot \cos \boldsymbol{\theta}_i$ is first concatenated with $\boldsymbol{z}_i \odot \sin \boldsymbol{\theta}_i$ and then fed to the Agent-FC. Different agents interact with each other with both the amplitude and phase information through Agent-FC. $\boldsymbol{o}_{tar}$ is the social context containing the interaction information between the target and other agents.

Future Trajectory Simulation Module

Future motion states of agents are highly correlated. Anticipation on the motion states of nearby agents in the near future and modeling their interactions can help predict the trajectory of the target. PiP (Song et al. 2020) addresses this issue by employing the fully convolutional network for capturing future inter-dependencies between agents. However, it utilizes future trajectory information, which is unavailable in real-world driving scenarios.

In fact, when driving on highways, maneuver changes of vehicles are far less frequent than driving on urban streets. The work (Deo and Trivedi 2018) has pointed out that the vehicle maneuvers could be classified by lateral and longitudinal maneuvers. Lateral maneuvers consist of lane-keeping, left and right lane changes. Longitudinal maneuvers include normal driving and braking. Based on our data analysis on the dataset NGSIM US-101 and I-80 (Halkias and Colyar 2006), we observe that more than 96% of track records indicate corresponding vehicles are keeping their lanes straight and no more than 20% shows vehicles performed braking operation. Therefore, we can use some simple models to predict a rough future trajectory of each neighbor. For example, by observing an agent braking and slowing down, we can roughly predict it probably will not change lanes in the future. We use this coarse-grained prediction of neighbors to help more accurately predict the trajectories of the target. Kalman Filter (KF for short in Figure 3) is employed here to predict the future locations of local neighbors. We define local neighbors as agents within ± 60 feet in the longitude direction and within the two adjacent lanes centered on the target (e.g. vehicles framed by blue box in Figure 3). With the estimated trajectories of local neighbors, we simulate a future driving scenario for the target. Simulated trajectories are encoded by the simulated trajectory encoder in Figure 3 to obtain the final hidden states. LSTMs here have different parameters from LSTMs in the historical trajectory encoder as they belong to different time horizons.

Next, as shown in the middle upper part of Figure 3, Social context $\boldsymbol{o}_{tar}$ obtained in the historical interaction modulation module and final hidden states of local neighbors fill up a social tensor according to their spatial location. Wave-pooling is applied once again for aggregating all the information accessible within the social tensor. Finally, the output of Wave-pooling $\boldsymbol{o}_{tar}^{'}$ is a social context which covers both historical and future social information.

Trajectory Prediction Module

Influenced by various factors, such as weather, road conditions and drivers' emotions, future motions of an agent have multiple possibilities. Therefore, trajectory prediction tends to be inherently multi-modal. To address this issue, our trajectory prediction decoder is built upon the work (Deo and Trivedi 2018) to produce multi-modal distributions based on 6 maneuver classes $M = \{m_i \mid i = 1, 2, \ldots, 6\}$ including 3 lateral classes (lane-keeping, left and right lane changes) and 2 longitudinal classes (normal driving and braking). Moreover, it estimates the probability of each maneuver class, denoted as $P(m_i)$. Social context $\boldsymbol{o}_{tar}^{'}$ is first concatenated with dynamics encoding $\boldsymbol{d}_{tar}$ to form a trajectory encoding $\mathcal{T}_{tar}$. As shown in the right part of Figure 3, $\mathcal{T}_{tar}$ will be fed to a pair of fully connected layers followed by two soft-max layers to output the lateral and longitudinal maneuver probabilities $P(m_i \mid \mathcal{X})$ respectively. $\mathcal{X}$ is the historical trajectories of agents in a scene. We concatenate $\mathcal{T}_{tar}$ with maneuver encoding of 6 classes respectively. The maneuver encoding of each class contains one-hot vectors of corresponding lateral and longitudinal maneuver class. Concatenated embedding will be decoded through LSTMs for generating the parameters of a bivariate Gaussian distribution. The predicted locations Y_i^t at time step t are given by

$$Y_i^t \sim \mathcal{N}\left(\mu_i^t, \sigma_i^t, \rho_i^t\right), \tag{7}$$

where μ_i^t and σ_i^t are the means and variances of future locations respectively, and ρ_i^t is the correlation coefficient. We denote Θ as parameters of Gaussian distribution. Then, the posterior probability of the target's future trajectories could be estimated by

$$\mathrm{P}(\mathbf{Y} \mid \mathcal{X}) = \sum_i \mathrm{P}_\Theta\left(\mathbf{Y} \mid m_i, \mathcal{X}\right) \mathrm{P}\left(m_i \mid \mathcal{X}\right), \tag{8}$$

We would like to minimize the negative log likelihood loss $\mathcal{L}$ of the true trajectories under the true maneuver class m_{true} of targets. The final loss function is as follows,

$$\mathcal{L} = -\log\left(\mathrm{P}_\Theta\left(\mathbf{Y} \mid m_{true}, \mathcal{X}\right) \mathrm{P}\left(m_{true} \mid \mathcal{X}\right)\right). \tag{9}$$

Implementation Details

A 13 × 5 spatial grid is defined around the target, where each column corresponds to a single lane, and the rows are separated by a distance of 15 feet. MLP that embeds historical trajectories is composed of a fully connected layer with size 32 and ReLU as the activation function. Both encoder and decoder in our model are based on LSTM. The dimension of the hidden state for encoder LSTM is 64 and for decoder LSTM is 128. The model is implemented using Pytorch and trained in an end-to-end manner using Adam with a learning rate 0.001.

Experiments

Dataset

We use two public datasets NGSIM (Halkias and Colyar 2006) and highD (Krajewski et al. 2018) for evaluaiton. NGSIM (Halkias et al. 2016) contains 2 freeway trajectory datasets US-101 and I-80. Each dataset of NGSIM contains 45 minutes of vehicle trajectory data recorded at 10Hz. highD (Krajewski et al. 2018) is a vehicle trajectory dataset recorded on German highways. It is collected at six different locations and includes more than 110,500 vehicles. We split the whole dataset into training, validation and testing sets. 70% of the data are used for training, 10% for evaluation and 20% for testing. We split each of the trajectories into 8s segments consisting of 3s of past and 5s of future trajectories.

Evaluation Metrics and Baselines

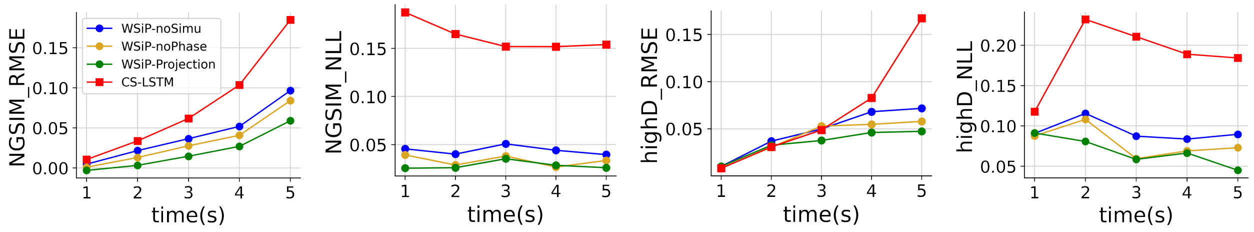
We evaluate the result in terms of RMSE over a prediction horizon of 5s. For multi-modal methods, we select the predicted trajectory corresponding to the maneuver with the highest probability. The RMSE at each time step t can be calculated by $RMSE(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i^t - \hat{Y}_i^t\right)^2}$, where N is the total number of test instances. Y_i^t and $\hat{Y}_i^t$ are the ground-truth and predicted coordinates of agent a_i at time step t, respectively.

RMSE is limited for measuring the performance of multi-modal methods since it tends to average all the predicted results. To address this issue, we also adopt the negative log-likelihood (NLL) of the true trajectories under the predictive distributions fitted by the models following CS-LSTM (Deo and Trivedi 2018).

We compare our model with the following baselines.

Dataset	Time	Metric(RMSE/NLL)						
		CV	V-SLTM	S-LSTM	CS-LSTM	S-GAN	PiP-noPlan	WSiP(ours)
NGSIM	1s	0.73/3.72	0.68/2.14	0.59/2.10	0.58/1.96	0.57/-	0.57/1.83	**0.56/1.77**
	2s	1.78/5.37	1.66/3.81	1.29/3.66	1.27/3.46	1.32/-	1.24/3.41	**1.23/3.30**
	3s	3.13/6.40	2.96/4.76	2.13/4.61	2.11/4.32	2.22/-	2.05/4.30	**2.05/4.17**
	4s	4.78/7.16	4.56/5.42	3.21/5.37	3.19/4.95	3.26/-	**3.07**/4.94	3.08/**4.80**
	5s	6.68/7.76	5.44/6.03	4.55/5.99	4.53/5.48	4.41/-	4.34/5.49	**4.34/5.32**
highD	1s	0.33/1.94	0.22/0.55	0.21/0.46	0.24/0.43	0.30/-	0.21/0.36	**0.20/0.31**
	2s	0.78/3.09	0.65/2.65	0.65/2.55	0.68/2.54	0.78/-	0.62/2.41	**0.60/2.31**
	3s	1.62/4.85	1.32/3.94	1.31/3.81	1.26/3.72	1.46/-	1.26/3.60	**1.21/3.51**
	4s	2.43/6.12	2.22/4.87	2.16/4.67	2.15/4.51	2.34/-	2.14/4.40	**2.07/4.32**
	5s	3.67/7.03	3.43/5.59	3.29/5.35	3.31/5.13	3.41/-	3.27/5.03	**3.14/4.95**

Table 1: Comparison results between baselines and our method on NGSIM and highD datasets. We report two error metrics RMSE and NLL for 5s prediction horizon.



(a) RMSE of NGSIM

(b) NLL of NGSIM

(c) RMSE of highD

(d) NLL of highD

Figure 4: Comparison between WSiP and the variants for ablation study.

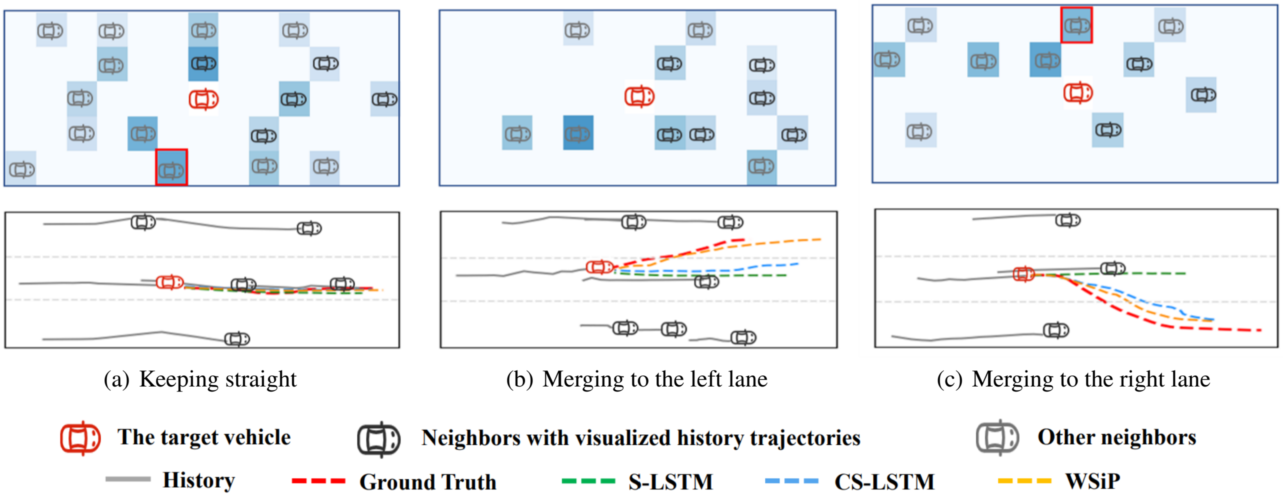
- *CV*: Constant Velocity based on the second order Kalman Filter is used as the representative baseline of traditional methods.
- *V-LSTM*: Vanilla LSTM uses a single LSTM to encode historical trajectories of the target vehicle without considering the interaction with surrounding agents.
- *S-LSTM*: Social LSTM (Alahi et al. 2016) uses fully connected layers and generates the uni-modal distribution of the future locations.
- *CS-LSTM*: Convolutional Social LSTM (Deo and Trivedi 2018) uses the convolutional social pooling and also generates multi-modal trajectory predictions.
- *S-GAN*: Social GAN (Gupta et al. 2018) trains a GAN based adversarial learning framework to generate diverse trajectories for multi-agent in a spatial-centric manner.
- *PiP-noPlan*: PiP (Song et al. 2020) uses the convolutional social pooling and a fully convolutional network to generate muiti-modal trajectory predictions. We remove the planning coupled module (PiP-noPlan) as our baseline because the future motions of the controllable ego vehicle is unavailable in real-world driving scenarios.

To study whether each module of WSiP is useful, we also compare WSiP with the following variants.

- *WSiP-noSimu*: It is a variant of WSiP by removing the future trajectory simulation module.
- *WSiP-noPhase*: We use WSiP-noPhase which aggregates interactions among agents without the phase information to evaluate the effectiveness of Wave-pooling.
- *WSiP-Projection*: We use WSiP-Projection which treats the final hidden states of agents as their phase information to verify Plain-FC can capture different motion states and learn better phase information for aggregation.

Result

We compare our method against baselines on two metrics RMSE and NLL, as shown in Table 1. It shows that CV and V-LSTM perform worst in all the baselines because they simply use the target vehicle's historical trajectories without the consideration of the interactions of neighbors. Both S-LSTM and CS-LSTM perform much better than CV and V-LSTM, which indicates considering the inter-dependencies between vehicles can truly improve the performance of trajectory prediction. S-GAN samples trajectories rather than generating distributions, and it selects a sample with the minimal error. Therefore, it does not have the NLL result. By fusing the encoding among different agents, PiP-noPlan performs better than other baselines, which suggests that considering future motion states of other agents is also helpful. WSiP achieves the best results in terms of two metrics in most cases with only one exception (RMSE on NGSIM dataset when the time is 4s). Specifically, in terms of RMSE on dataset highD, WSiP outperforms PiP-noPlan by 4% and S-GAN by 8% when the prediction time horizon is 5s. This verifies Wave-pooling can better model the interactions between agents, and the future trajectory simulation module can anticipate the future motion states of neighbors.


(a) Keeping straight

(b) Merging to the left lane

(c) Merging to the right lane

Figure 5: Visualization of three driving scenarios. Upper: heatmaps showing the cosine similarity of the target and neighbors. Lower: three example cases predicted by S-LSTM, CS-LSTM and WSiP.

Ablation Study

We next investigate the effect of each part of our model to the performance. Figure 4 shows the error increase in terms of RMSE and NLL of the three variants and CS-LSTM compared with WSiP. Higher curve values mean the performance of the corresponding model is worse than WSiP. One can see that WSiP-noSimu obviously performs worse than other variants due to its non-consideration of future motions of neighboring agents. Note that WSiP-noSimu still outperforms CS-LSTM, implying the proposed Wave-pooling can effectively capture the interactions between agents and it is superior to other social pooling strategies. Phase is used to modulate the interactions between agents. Without using phase embedding, the prediction error of WSiP-noPhase model is higher than WSiP. This verifies the phase is helpful in aggregating information from neighbors. WSiP-Projection directly uses the final hidden states of agents as their phase embedding. Although WSiP-Projection is superior to other variants, it still performs worse than WSiP which uses Plain-FC to obtain phase embedding according to motion states of agents. It suggests that Plain-FC could learn better representations for aggregating information from neighbors. This result verifies that wave-pooling and future trajectory simulation module are both indispensable to boosting the model performance.

Case Study for Interpretability Analysis

We visualize the prediction results of several cases in Figure 5 to further show the effectiveness of WSiP and its interpretability. We select three driving scenarios including the target keeping straight, merging to the left lane and merging to the right lane. According to Eq. 2, the phase difference affects the interaction aggregation process. A smaller phase difference between two agents means a higher correlation and stronger interaction. We show the cosine similarity between phase embeddings of the target and neighboring agents with heatmaps in the upper part of Figure 5. Darker color means higher cosine similarity. From the heatmaps in Figure 5, one can see that in general the distance between the target and its neighbors is positively correlated to their phase similarity, which indicates strong social interactions usually happen between close neighboring agents. This is consistent with our intuition. However, some agents on non-adjacent lanes (e.g. agents framed by red boxes in the upper part of Figure 5) may also be highly correlated to the target due to their high-order interactions, which is largely ignored by existing works. As shown in Figure 5(a), S-LSTM, CS-LSTM and WSiP all make accurate predictions that the target will keep going straight. Figure 5(b) shows the target merges to the left lane. S-LSTM wrongly predicts that it will keep going straight because it is not a multi-modal prediction model and thus fails to predict the possible multiple trajectories. Both CS-LSTM and WSiP predict the target will merge to the left lane. WSiP has smaller error and the direction of predicted trajectory is closer to the ground truth. Figure 5(c) shows the target merges to the right lane. Both CS-LSTM and WSiP make correct predictions, but the predicted trajectory of WSiP is closer to the ground truth.

Conclusion

This paper proposes a wave superposition inspired pooling method WSiP under an encoder-decoder learning framework for vehicle trajectory prediction. WSiP uses Wave-pooling for capturing the high-order dynamic interactions among vehicles. Wave-pooling novelly represents each vehicle as a wave with the amplitude and phase. The phase modulates the interactions among vehicles according to their motion states. WSiP anticipates the future motion states of neighboring vehicles through future trajectory simulation to help predict the trajectory of the target. Experimental results on two real datasets verify the effectiveness of WSiP.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (Nos. 62172443 and 61802424), Hunan Provincial Natural Science Foundation of China (No. 2022JJ30053), CAAI-Huawei MindSpore Open Fund and the High Performance Computing Center of Central South University.

References

Agamennoni, G.; Nieto, J. I.; and Nebot, E. M. 2011. A Bayesian approach for driving behavior inference. In *IEEE IV*.

Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*.

Altché, F.; and de La Fortelle, A. 2017. An LSTM network for highway trajectory prediction. In *IEEE ITSC*.

Aoude, G. S.; Luders, B. D.; Lee, K. K.; Levine, D. S.; and How, J. P. 2010. Threat assessment design for driver assistance system at intersections. In *IEEE ITSC*.

Deo, N.; and Trivedi, M. M. 2018. Convolutional social pooling for vehicle trajectory prediction. In *CVPR Workshop*.

Firl, J.; Stübing, H.; Huss, S. A.; and Stiller, C. 2012. Predictive maneuver evaluation for enhancement of car-to-x mobility data. In *IEEE IV*.

Gao, J.; Sun, C.; Zhao, H.; Shen, Y.; Anguelov, D.; Li, C.; and Schmid, C. 2020. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*.

Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; and Alahi, A. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*.

Halkias, J.; and Colyar, J. 2006. Next Generation SIMulation Fact Sheet. Technical report, Federal Highway Administration (FHWA). FHWA-HRT-06-135.

Helbing, D.; and Molnar, P. 1995. Social force model for pedestrian dynamics. *Physical review E*, 51(5): 4282.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*.

Kim, B.; Kang, C. M.; Kim, J.; Lee, S. H.; Chung, C. C.; and Choi, J. W. 2017. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *IEEE ITSC*.

Klingelschmitt, S.; Platho, M.; Groß, H.-M.; Willert, V.; and Eggert, J. 2014. Combining behavior and situation information for reliably estimating multiple intentions. In *IEEE IV*.

Krajewski, R.; Bock, J.; Kloeker, L.; and Eckstein, L. 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *IEEE ITSC*.

Kuefler, A.; Morton, J.; Wheeler, T.; and Kochenderfer, M. 2017. Imitating driver behavior with generative adversarial networks. In *IEEE IV*.

Lefèvre, S.; Laugier, C.; and Ibañez-Guzmán, J. 2011. Exploiting map information for driver intention estimation at road intersections. In *IEEE IV*.

Liang, M.; Yang, B.; Hu, R.; Chen, Y.; Liao, R.; Feng, S.; and Urtasun, R. 2020. Learning lane graph representations for motion forecasting. In *ECCV*.

Liebner, M.; Baumann, M.; Klanner, F.; and Stiller, C. 2012. Driver intent inference at urban intersections using the intelligent driver model. In *IEEE IV*.

Liu, Y.; Zhang, J.; Fang, L.; Jiang, Q.; and Zhou, B. 2021. Multimodal motion prediction with stacked transformers. In *CVPR*.

Mohamed, A.; Qian, K.; Elhoseiny, M.; and Claudel, C. 2020. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*.

Nawaz, A.; Zhiqiu, H.; Senzhang, W.; Hussain, Y.; Khan, I.; and Khan, Z. 2020. Convolutional LSTM based transportation mode learning from raw GPS trajectories. *IET Intelligent Transport Systems*, 14(6): 570–577.

Park, S. H.; Kim, B.; Kang, C. M.; Chung, C. C.; and Choi, J. W. 2018. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In *IEEE IV*.

Song, H.; Ding, W.; Chen, Y.; Shen, S.; Wang, M. Y.; and Chen, Q. 2020. Pip: Planning-informed trajectory prediction for autonomous driving. In *ECCV*.

Tang, C.; and Salakhutdinov, R. R. 2019. Multiple futures prediction. In *NeurIPS*.

Tang, Y.; Han, K.; Guo, J.; Xu, C.; Li, Y.; Xu, C.; and Wang, Y. 2022. An image patch is a wave: Phase-aware vision mlp. In *CVPR*.

Thomson, G. P.; and Reid, A. 1927. Diffraction of cathode rays by a thin film. *Nature*, 119(3007): 890–890.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*.

Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. 2021. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*.

Wang, S.; Cao, J.; and Yu, P. 2022. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(8): 3681–3700.

Wang, S.; Zhang, X.; Cao, J.; He, L.; Stenneth, L.; Yu, P. S.; Li, Z.; and Huang, Z. 2017. Computing urban traffic congestions by incorporating sparse GPS probe data and social media data. *ACM Transactions on Information Systems*, 35(4): 1–30.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*. MIT press Cambridge, MA, USA.