2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

Multi-Agent Driving Behavior Prediction across Different Scenarios with Self-Supervised Domain Knowledge

Hengbo Ma*, Yaofeng Sun*, Jiachen Li, Masayoshi Tomizuka, *Life Fellow, IEEE*

***Abstract*— How to make precise multi-agent trajectory prediction is a crucial problem in the context of autonomous driving. It is significant to have the ability to predict surrounding road participants' behaviors in many different, seen or unseen scenarios for enhancing autonomous driving safety and efficiency. Extensive research has been conducted to improve the overall prediction performance based on one enormous dataset or pay attention to some specified scenarios. However, how to generalize the prediction to different scenarios is less investigated. In this paper, we introduce a graph-neural-network-based framework for multi-agent interaction-aware trajectory prediction. In contrast to recent works which use the Cartesian coordinate system and global context images directly as input, we propose to leverage human's prior knowledge such as the comprehension of pairwise relations between agents and pairwise context information extracted by self-supervised learning approaches to attain an effective Frenét-based representation. We evaluate our method across different traffic scenarios with diverse layouts and compare it with state-of-the-art methods. We demonstrate that our approach achieves superior performance in terms of overall performance, zero-shot and few-shot transferability.**

I. Introduction

Multi-agent behavior prediction has a pivotal role in many real-world applications, such as autonomous driving and mobile robot navigation. Making a precise prediction in different situations gives a promise of safety with proper planning algorithms. Human drivers can transfer their prediction and driving ability from previous scenarios to new ones only after driving in the new scenarios a few times. Imagine that a driver Alice has driven in San Francisco (SF) for many years. She goes to New York (NY) for business and rents a car. Although she has never been to NY, where driving behaviors and road layouts are different from those in SF, she can be familiar with the driving patterns in NY very quickly. However, much of the current literature on behavior prediction pays particular attention to improving overall prediction performance. In this work, we focus more on the generalization problems of multi-agent trajectory prediction in autonomous driving applications as shown in Fig. 1.

Recently, several works in machine learning and computer vision indicate that introducing inductive bias is necessary to improve the generalization of the deep learning framework. The inductive biases could be specified deep learning model structures, constraints, and context information, etc. Such inductive bias could be used to extract general representation of data for future usage. For instance, there exist many works proposing different model structures such as graph neural network [1], [2], transformers [3] to capture the multi-agent interaction mechanism. These specified deep learning model structures indeed improve the generalization, while they still lack the subtle domain knowledge of autonomous driving to improve the transferability in advance. Although several works are incorporating more context information such as high-definition maps [4], [5], [6] to improve the prediction accuracy, most of the methods are trained in an end-to-end style. It is not clear that if such context representation and training strategy are efficient for generalization.

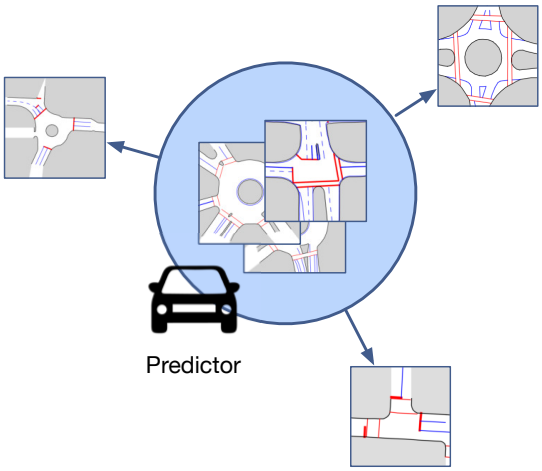


Fig. 1: The illustration of generalization of multi-agent trajectory prediction across different scenarios. The predictor could be trained on a dataset of several scenarios, then tested on several new scenarios without new data (zero-shot) or with a small batch of data (few-shot) for training.

In this work, we propose an approach to utilize the context information such as the references of agents more effectively with leveraging human's prior knowledge. First, we argue that instead of providing the road layout information, i.e., the reference of each vehicle implicitly, such as given the rasterized images of the high-definition map directly as input, we can explicitly incorporate the references information to future trajectories predictions by Frenét transformation. The Frenét transformation can constrain the predicted trajectories around the references, which improves the zero-shot and few-shot transferability. Then we design a set of features based on the human understanding of interaction behaviors in the Frenét coordinate system serving as the inductive bias. Lastly, in contrast to using end-to-end supervised learning, we apply the self-supervised learning technique, which can reduce invariant factors to get a more general representation

*The authors contributed equally to this work.

H. Ma, J. Li and M. Tomizuka are with University of California, Berkeley, CA 94720, USA (e-mail:hengbo_ma, jiachen_li, tomizuka@berkeley.edu). Y. Sun is with Peking University.

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564510

for the intrinsic relative geometry information of references of each interaction pair of agents. After obtaining the feature representation, we use a message-passing graph neural network to capture the interaction behaviors. We argue that such representations not only improve the overall prediction performance but also improve the generalization and transferability significantly.

The main contributions are summarized as follows:

- We demonstrate an effective approach to leveraging Frenét-based trajectory prediction and rule-based interaction-level semantic classification to extract a good feature representation, which enhances the transferability across different scenarios.
- We adopt a self-supervised learning technique to extract the context information of interaction pairs and demonstrate that it can achieve better prediction performance.
- We evaluate the overall prediction accuracy and transferability of the proposed approach on a benchmark dataset including various interactive driving scenarios. The framework achieves significant enhancement compared with state-of-the-art methods.

II. Related Work

A. Behavior and Trajectory Prediction

In recent years, there has been an increasing amount of literature on trajectory prediction due to the rising of topics including autonomous driving and human-robot interaction. Early examples of research focus on using model-based or traditional machine learning methods such as intelligent driving model [7], hidden Markov model [8] to predict the future trajectories. With the increase of the computational power, deep-learning-based methods become more available and achieve superior performances compared with the traditional methods. Furthermore, more particular issues, including how to deal with the different number of agents, probabilistic prediction and how to incorporate map information, are investigated. One line of the methods such as [9], [10], [11], [12], [13], etc. propose generative learning frameworks to obtain the complex, multi-modal future trajectory prediction. The development of graph neural networks [14], [1], [2], [15], [16] and attention mechanism provided powerful tools to solve the multi-agent prediction problems. Several works including [17], [18], [19], [20], [21] successfully adopt such ideas into the multi-agent trajectories problems. Other approaches focus on how to incorporate more information, such as the high-definition (HD) map and point clouds. Convnet [22] proposes to use the rasterized image of maps directly as inputs of a convolutional neural network. Vectornet [4] proposes to encode the vectors of lanes into a graph as the context information. [23] designs a method to utilize both the maps and LiDAR information. In contrast to these methods extracting the future road information of one agent implicitly from the contextual inputs such as the image or vectors of roads, we explicitly constraint the future predicted trajectories by mapping it into the Frenét coordinate system according to the reference of one agent. Frenét representation are well-investigated, especially in motion planning literature [24] while there is little work about how to incorporate it into trajectory prediction framework. We show that this approach is more effective and enhances the performance of generalizations to new scenarios.

B. Self-Supervised Learning

Since the increasing expense of labeling massive data, researchers have shown an increased interest in learning representations from unlabeled data. Sometimes it is impossible to label data before knowing the following tasks. However, since many data have their own particular information structures, e.g., the local relation in the image, it becomes possible to exploit such information to obtain the intrinsic representation for future usage. Self-supervised learning is a technique to extract efficient representation before the task (e.g, classification) is known. Since there is no task information, the auxiliary (pretext) tasks should be defined in order to discover the similarity between different features. These pretext tasks provide pseudo labels as supervision. For instance, color transformation and geometric transformation are usually used in the computer vision area. Some works propose to use the contrastive loss as the self-supervision [25], [26], [27]. The intuitive explanation of contrastive learning is to make similar samples closer and make dissimilar samples repulse each other [28]. Self-supervised learning has been empirically demonstrated to be able to extract better representation when the labels are limited in many applications such as image classification [25], natural language processing [29], and reinforcement learning [30], [31]. Recent work [32] shows that self-supervised learning can also improve the few-shot learning performance in image classification. We adopt the contrastive learning concepts to extract the relative geometry information of references.

III. Problem Formulation

Without loss of generality, we assume that there are N agents in a case. In different cases, there may be a varying number of agents. We have the historical observations $O^i_{t-H+1:t}$ of each agent i, which includes its trajectory $X^i_{t-H+1:t}$ and the reference information R^i. The reference R^i represents the vehicle's routing information. It can be the middle of the lane which the vehicle is following. Such information can be extracted from the HD map. We denote the rasterized image of R^i as I^i, and the rasterized image of each pair of references R^i, R^j as I^{ij}. Given such information, we aim to predict the conditional distribution $P(X_{t+1:t+F}|O_{t-H+1:t})$. We denote H as the length of the historical horizon and F as the length of the prediction horizon. The variables without agent index i are denoted as the collections of different agents' corresponding variables (e.g. $X = \{X^i\}_{i=1:N}$). For the zero-shot / few-shot learning setting, we train our model on one dataset mixing several scenarios and test it on several new scenarios.

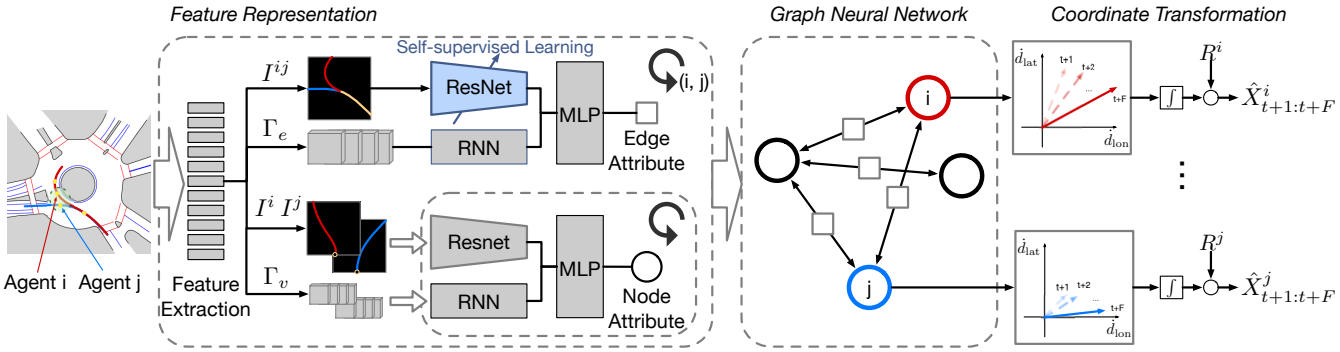


Fig. 2: Overview of the proposed framework. There are three procedures in our approaches: **I** Feature Representation (Sec. IV-B). During the feature representation phase, the original multi-agent trajectory data is processed pairwise. Here we show one pair of agents (Agent i and Agent j) as an example. The blue ResNet block is pre-trained with self-supervised learning (Sec. IV-B.2). **II** Graph Neural Network (Sec. IV-C). Once we obtain the node attributes and edge attributes, we use a graph neural network to capture the interaction mechanism between agents. **III** Frenét-Cartesian Transformation (Sec. IV-B). The graph neural network predicts the future velocities of agents in the Frenét coordinate system. We integrate the predicted velocity of each agent and transform the trajectories into the Cartesian coordinate system.

IV. Methodology

A. Framework Overview

The whole framework is introduced in three parts: feature representation with human's prior knowledge, graph neural network design, and the training loss. The section of feature representation is divided into two parts: Frenét-based trajectory representation and self-supervised context representation. Sec. IV-C introduces the graph neural network, which is divided into the attribute encoding layer, the message passing procedure and multi-modal decoder module. Sec. IV-D introduces the training loss we used. A high-level summary is shown in Fig. 2.

B. Feature Representation

1) Frenét-based Trajectory Representation: The Frenét coordinate system is used in this work since it can represent arbitrary reference paths efficiently. In the Frenét representation, the Cartesian coordinate (x, y) is transformed into longitudinal distance d_{lon} and lateral displacement d_{lat} given the reference R. Since the Frenét representation has already included the geometry information about each vehicle's reference, the final prediction results will incorporate the reference naturally. [33] argues that the topological relationship between any of two references can be decomposed into different types, and we adopt these ideas as prior knowledge to define four types of features: node features and three types of edge features.

For node features, we use the longitudinal speeds $\dot{d}^i_{\text{lon}}$, lateral speeds $\dot{d}^i_{\text{lat}}$, and the rasterized image I^i of the vehicle i's reference R^i as the features. The image I^i use the vehicle i's coordinate system, where the Y-axis direction of image is the velocity's direction. We denote the node feature selection and extraction as a mapping $\Gamma_v : \mathcal{X} \to \mathcal{S}_v$, where $\mathcal{X}$ is the trajectory space and $\mathcal{S}$ is the feature space.

For edge features, we illustrate different conditions of interaction pairs in Fig. 3. In Fig. 3 (a), when the references

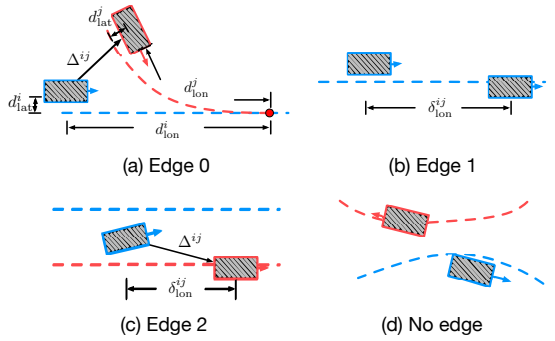


Fig. 3: The illustration of different types of edges. Different colors represent different agents. The dashed lines are the references for agents.

of two vehicles intersect, we set the intersection point as the origin of the Frenét coordinate system. We denote $d^{\text{i, j}}_{\text{lon}}$ as the collection of vehicle i's and j's longitudinal distance to the origin point. $d^{\text{i, j}}_{\text{lat}}$ has the similar definition for the lateral displacement. We denote Δ^{ij} as the relative position of the vehicle i and j in the Cartesian coordinate system, where the origin point is the location of the vehicle i and the Y-axis direction is its velocity direction. We also employ the context information of the relation between the references of a pair of agents as one of the features C^{ij}. The details of the feature C^{ij} will be introduced in Sec IV-B.2. In Fig. 3 (b), if two vehicles share the same reference, we define the relative longitudinal distance δ^{ij}_{lon} between them. In Fig. 3 (c), if two vehicles' references do not intersect while lane change is feasible, we use δ^{ij}_{lon} and Δ^{ij} as features. In Fig. 3 (d), if two references are mutually exclusive, there will be no edge. Table I summaries the feature selections for different types of features. We denote the edge feature selection and extraction as a mapping $\Gamma_e : \mathcal{X} \times \mathcal{X} \to \mathcal{S}_e$, where $\mathcal{X}$ is the trajectory space and $\mathcal{S}$ is the feature space. We also denote the edge type of one pair of agents i and j as α_{ij}.

In order to generate such trajectory representation, there

TABLE I: The different types of features.

Features	Node	Edge0	Edge1	Edge2
d^i_{lon}	✓			
d^i_{lat}	✓			
R^i	✓			
$d^{i,j}_{\text{lon}}$		✓		
$d^{i,j}_{\text{lat}}$		✓		
Δ^{ij}		✓		✓
C^{ij}		✓		
δ^{ij}_{lon}			✓	✓

are two submodules here: reference path extraction and coordinate transformation.

Reference Path Extraction This module aims to extract each vehicle's reference. If a high-definition map is provided and the road layout is very simple, we can directly use the reference defined in the HD map. However, if the road layout is complicated, such as roundabouts, using the hand-crafted references is not accurate. One better solution is to use a few trajectory data with the same starting area and ending area to get the approximate references. We first collect the trajectories according to the starting and ending areas. Each starting and ending area is defined as a quadrilateral area indicating an agent coming from or heading to this path. Then we can select a few data with this reference. For a set of trajectories with the same starting and ending area, we use a polynomial function to fit them. Then we sample points from the fitted curve every 0.05 meter and use them as the reference path.

Coordinate Transformation To transform coordinates from Cartesian to Frenét, we can find the nearest point on the references, and then calculate the lateral displacement d_{lat} to the projection point and the longitudinal distance d_{lon}. Through the inverse process, we can transform the trajectories in the Frenét coordinate system back to the Cartesian coordinate system.

2) Context Representation with Self-Supervised Learning: Despite the Frenét coordinate system is used in the feature representation, it only contains the geometric information of the reference path of each vehicle. When two vehicles are interacting with each other, their behaviors also depend on the relations between the two reference paths. The relations between two references have different levels of abstractions such as topology and geometry [33]. Under the circumstance that the references of two vehicles i and j intersect, we can rasterize those two references in Image I^{ij} with the intersection point as the center point of the image. We use different pairs of colors to indicate the vehicle's moving direction in order to distinguish two references. One example is shown in Fig. 4.

We suggest that the relation between two references is invariant to different positions (views) of vehicles. For instance, the drivers of two different vehicles will have the same understanding of the relation between the two references. Besides, the orientation information of reference for each agent has already been provided in the node

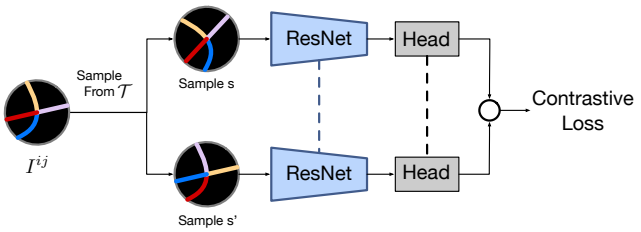


Fig. 4: The contrastive learning framework. Sample s indicates one sample from the rotation transformation, and s' indicates one sample from the semantic exchanging transformation. I^{ij} is one example of the rasterized image of two intersected references. The blue-yellow curve represents one reference and the direction is from blue to yellow. The red-purple curve represents the other reference and the direction is from red to purple.

feature, i.e., the image I^i. We assume that there is an effective abstraction that can represent the relation between two references and use the self-supervised technique to extract such representation. We utilize a similar approach in SimCLR [25] to discover the similarities between different objects by designing some pretext tasks. These pretext tasks serve as data augmentation for generating positive samples. Considering the property of the context images as we suggest above, we define the family of tasks $\mathcal{T}$ as:

- **Rotation**: The original image is centered at the intersection point, which is defined as the first point at which two references intersect. Then the original image is rotated randomly from 0 to 2π. In Fig. 4, s shows one sample by rotation.
- **Semantic Exchanging**: We need to use distinct colors to indicate different references to avoid vagueness (i.e. which segment belongs to which reference, what is the direction of one reference). Since the topology and geometry are not related to the semantic order, we exchange the colors of two different references. In Fig. 4, s' shows one sample by semantic exchanging.

Under those pretext tasks $\mathcal{T}$, the loss function for a positive pair of samples I^{ij}_s and $I^{ij}_{s'}$ is:

$$l(I^{ij}_s, I^{ij}_{s'}) = -\log \frac{e^{\beta \cos(z_s, z_{s'})}}{\sum_{k \neq s} e^{\beta \cos(z_s, z_k)}}, \tag{1}$$

where $z_s = \text{h}(\text{enc}(I^{ij}_s))$, and $I^{ij}_s, I^{ij}_{s'} \sim \tau(I^{ij})$ are the random samples. h is a projection head. $\tau(\cdot) : \mathcal{I} \to \mathcal{I}$ is a random function sampled from the pretext task family $\mathcal{T}$. After training with the contrastive loss, we set $C^{ij} = \text{enc}(I^{ij})$ as the representation of context information.

C. Graph Neural Network Design

We can represent the agents in the traffic as a graph and use a graph neural network to capture the interaction behaviors. The graph can be defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_i\}, i \in \{1, \ldots, N\}$ and $\mathcal{E} = \{e_{ij}\}, i, j \in \{1, \ldots, N\}$. v_i, e_{ij} denote the node attribute and the edge attribute, respectively. Specifically, e_{ij} denotes the edge attribute from v_j to v_i. Given a set of trajectory observations, we can apply

the feature selection and extraction process mentioned in Sec. IV-B to compute the initial attributes for each node and edge. We name this layer as the attribute encoding layer:

$$\begin{aligned} v_i^0 &= f(\text{RNN}(\Gamma_v(X_{t-H+1:t}^i)), \text{ResNet}(I_t^i)), \\ e_{ij}^0 &= h(\text{RNN}(\Gamma_e(X_{t-H+1:t}^i, X_{t-H+1:t}^j)), C_t^{ij}), \end{aligned} \tag{2}$$

where f and h are the node and edge embedding functions.

Then we use the message passing mechanism similar as [15] to reason the multi-agent interaction mechanisms. For each node and edge, we have:

$$\begin{aligned} e_{ij}^m &= f_{e,\alpha_{ij}}^m([v_i^{m-1}, v_j^{m-1}]), \\ \gamma_{ij}^m &= \text{softmax}(e_{ij}^m), \\ v_i^m &= f_v^m(\sigma(\sum_{j \in \mathbf{N}(v_i)} \gamma_{ij}^m \mathbf{W} v_j^{m-1})),\ m = 1, \ldots, n. \end{aligned} \tag{3}$$

where $f_{e,\alpha_{ij}}$ denotes different edge embedding functions for different edge types α_{ij}. f_v denotes the embedding function for nodes. The superscripts of v_i^m, e_{ij}^m, f_v^m, $f_{e,\alpha_{ij}}^m$ denote the m-th message passing. $\mathbf{N}(v_i)$ denotes the neighbors of v_i.

In order to output multi-modal trajectory prediction, a Gaussian mixture model is used as the multi-modal decoder to generate the future velocities in the Frenét coordinate system:

$$\begin{aligned} &w_j = \text{softmax}(f_w^j(v_i^n)), \mu_j = f_\mu^j(v_i^n), \Sigma_j = f_\Sigma^j(v_i^n), \\ &\{\dot{d}_{\text{lon},t+1:t+F}, \dot{d}_{\text{lat},t+1:t+F}\} \sim \sum_j w_j \mathcal{N}(\mu_j, \Sigma_j), \end{aligned} \tag{4}$$

where w_j, μ_j, and Σ_j describe the weight, mean, and variance of the j-th Gaussian function.

After the predicted velocities of each agent are obtained, a first-order integrator is applied to get the predicted future positions in the Frenét coordinate system. Then the predicted trajectories would be transformed to the Cartesian coordinate system to evaluate the performance. We illustrate the Frenét-Cartesian transformation in Fig. 2.

D. Training Loss

We use the negative log-likelihood $\mathcal{L}(\theta, \mathcal{D})$ as the objective function during the training phase:

$$\mathbb{E}_{(O_{t-H+1:t}, X_{t+1:t+F}) \sim \mathcal{D}}[-\log P_\theta(X_{t+1:t+F} | O_{t-H+1:t})], \tag{5}$$

where θ represents all the parameters of our model. Since the direct outputs of our model are the predicted velocities $(\dot{d}_{\text{lon}}, \dot{d}_{\text{lat}})$ based on the Frenét coordinate system, we can also directly optimize the empirical loss based on the Frenét coordinate system during training.

V. Experiment Results

This section introduces the dataset, evaluation metrics and baselines in Sec. V-A and Sec. V-B firstly. Then the comparisons between our method and other baseline approaches of the overall prediction performance and transferability are demonstrated in Sec. V-C and Sec. V-D.

A. Dataset

The experiments were conducted on the INTERACTION dataset [34], which contains naturalistic motions of various traffic participants in a variety of highly interactive driving scenarios, including roundabouts, unsignalized intersection, and lane merging. In each scenario, the data is sampled from different locations. We choose this dataset for the following reasons. First, the geometry of road layouts is complicated. Most of the cases in other datasets are collected with simple road layouts like straight multi-lane and cross-style intersections. In contrast, the INTERACTION dataset contains more curved, challenging road layouts such as roundabouts. Second, it has a higher detection accuracy than other datasets and more highly interactive cases. Therefore, it is suitable for testing transferability across different scenarios and multi-agent interactive behavior prediction. We selected five urban representative scenarios (MA, FT, SR, EP-T, and EP-R), which have various road layouts in our experiments. We predicted the future 10 time steps (5.0s) based on the historical 4 time steps (2.0s) in all experiments.

B. Metrics and Baselines

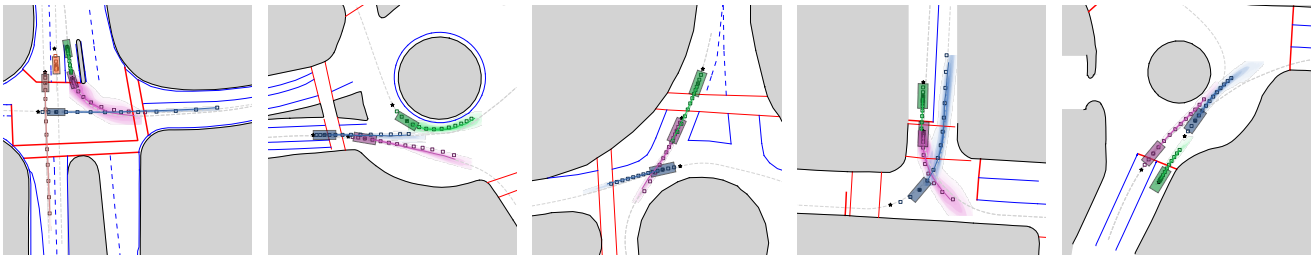
We adopt two widely used probabilistic prediction metrics. One is the minimum average displacement error (mADE), which computes the Euclidean distance between the ground truth positions and the closest trajectory from K candidates, which are sampled from the predicted probability distribution. The other is the minimum final displacement error (mFDE) that evaluates only the displacement error at the last time step. Both metrics are suitable to measure probabilistic prediction. We compare our method with five baseline approaches about the overall performance and transferability. We provide the same input information for all the methods. The following are the algorithms we compare: i) LSTM. Long-short term memory is a kind of recurrent neural networks used widely to learn the time-series pattern. We use it as a prediction model which does not consider the interaction explicitly; ii) Social LSTM (S-LSTM) [35]. The model designs a social pooling mechanism based on LSTM; iii) Social GAN (S-GAN) [18]. The model employs generative adversarial learning into S-LSTM; iv) Trajectron++ [36]. One of the state-of-the-art approaches employs spatial-temporal graph with dynamic constraints; v) Graph Neural Network (GNN). The network structure of this method is quite similar to our proposed method. The difference is that GNN uses the historical trajectories in the Cartesian coordinate system as the node features and does not use the routing-related edge features such as the edge types and the relative positions in the Frenét Coordinate system. Also, GNN does not employ our contrastive learning method to extract context features. Hence, GNN can also serve as an ablation method.

C. Prediction Performance in All Scenarios

1) Quantitative Results: First, we show the general prediction performance of our method compared with the baselines. We test all the models with all the data in different

TABLE II: Comparison of mADE / mFDE (Meters) in All Scenarios

	LSTM	S-LSTM	S-GAN	GNN	Trajectron++	Ours
@3.0s	0.344 / 0.597	0.438 / 0.692	0.400 / 0.652	0.271 / 0.534	0.185 / 0.336	**0.139 / 0.284**
@4.0s	0.598 / 1.128	0.611 / 1.071	0.599 / 1.042	0.469 / 0.969	0.360 / 0.677	**0.268 / 0.562**
@5.0s	0.918 / 1.888	0.879 / 1.721	0.845 / 1.528	0.695 / 1.457	0.608 / 1.167	**0.432 / 0.913**



(a) MA (b) FT (c) SR (d) EP-T (e) EP-R

Fig. 5: The visualization of prediction results. The box markers are the ground truth trajectories. The solid lines are the sample trajectories with the smallest mADE. The density maps around the solid lines are generated by using kernel density estimation (KDE) to fit the sampled trajectories. The grey dash lines are the references (R) of each vehicle. The star markers are the starting positions of the historical trajectories.

scenarios. The prediction results on all scenarios are shown in Table II. The units of reported metrics are meters in the Cartesian coordinate system. We observe that all the methods which model the interaction explicitly, such as S-LSTM, S-GAN, GNN, Trajectron++, and Ours, are better than LSTM, which predicts each vehicle independently. Our approach improves about 24.9% in mADE with 3 seconds prediction horizon and 28.9% in mADE with 5 seconds prediction horizon compared with the best baseline (Trajectron++). It is also about 15.5% and 21.8% improvement in mFDE with 3 seconds and 5 seconds prediction horizon, respectively. It shows that our method has significant improvement compared with baselines.

2) Qualitative Results: We visualize five typical cases where there are more than two vehicles in Fig. 5. The ground truth and predicted trajectories are shown in the same color for each vehicle. We find that the prediction results in all the scenarios are accurate. Besides, Our method is capable of capturing subtle behaviors such as yielding or not yielding in complicated scenarios.

D. Transferability to Other Scenarios

In this section, we compare the transferability of different methods. All the methods are trained on the mixed data of two scenarios: a roundabout (FT) and an intersection (MA). We chose these two scenarios since they can cover most of the different types of road layouts, including roundabout and intersection so that we can train a good basic predictor at the beginning. Then we evaluate the zero-shot and few-shot performance of transferring to another three different scenarios (two of them are different roundabouts, and the other is a different intersection). The results with 5 seconds prediction horizon are shown in Table IV. The mADE of our approach is 32.4%, 11.0%, and 38.4% better than the baseline methods in zero-shot transfer to SR, EP-T, and EP-R, respectively. For few-shot learning, we randomly sample 100 trajectories for each scenario as the training data. We demonstrate that our method performs better than the others with 40.6%, 30.8%, and 37.4% improvements in mADE for SR, EP-T, and EP-R. We observe that the improvement of zero-shot / few-shot learning on EP-T is relatively small compared with the other two scenarios. We suggest that the reason is that the road layout of EP-T is very similar to the one in MA, since they both include 90-degree intersections. It also implies that the more the scenarios are similar, the easier generalization will be. The observation also adheres to our intuition.

TABLE III: The Models used for Ablation

	Frenét-based	Contrastive	Image
Ours	✓	✓	✓
Ours-E2E	✓		✓
Ours-no_image	✓		
GNN			✓

VI. Ablative Analysis

We intend to answer the following questions with the ablation models in Table III.:

- **Does the self-supervised learning technique improve the performance, and is the context image I^{ij} useful?** We compare Ours, Ours-E2E, and Ours-no_image in Table III. The difference between them is the way they process the context images. Ours-no_image does not utilize the context image I^{ij}. Ours-E2E uses the context image but does not use contrastive learning.
- **Is the performance of the Frenét-coordinate-based trajectory prediction better than the Cartesian-coordinate-based one?** Here we compare the model GNN and Ours-E2E in Table III, since the only difference between these two models is whether the Frenét-based features are used.

TABLE IV: Comparison of mADE / mFDE (meters) in the Different Scenarios

		LSTM	S-LSTM	S-GAN	GNN	Trajectron++	Ours
SR	0-shot	1.849 / 4.470	1.927 / 3.625	1.801 / 3.539	1.977 / 4.714	1.395 / 2.333	**0.943 / 2.266**
	few-shot	1.184 / 2.452	1.458 / 3.010	1.267 / 2.385	1.180 / 2.430	1.043 / 1.816	**0.620 / 1.214**
EP-T	0-shot	1.278 / 2.824	1.594 / 3.207	1.655 / 3.465	1.587 / 3.607	1.092 / **1.703**	**0.972** / 2.279
	few-shot	0.978 / 1.873	1.037 / 2.071	1.098 / 2.239	1.028 / 2.064	0.838 / 1.528	**0.580 / 1.003**
EP-R	0-shot	2.268 / 5.306	2.824 / 5.531	2.590 / 5.130	2.388 / 5.514	2.074 / 3.478	**1.277 / 2.742**
	few-shot	1.453 / 2.970	1.520 / 3.281	1.483 / 2.739	1.393 / 2.743	1.328 / 2.376	**0.831 / 1.462**

A. *Self-Supervised Learning Ablation*

1) Quantitative Analysis: To prove the effectiveness of the self-supervised learning method, we conduct experiments on the whole dataset, including all the scenarios for the three methods: Ours, Ours-E2E, and Ours-no_image. Ours-E2E trains the whole model in an end-to-end fashion, and Ours-no_image does not utilize the context image I^{ij} as one of the edge features. In Table V, we find that Ours is about 10.2% better than Ours-E2E and approximately 16.4% better than Ours-no_image with 5.0s prediction horizon. Thus, it illustrates that employing self-supervised auxiliary tasks to pre-train the feature embedding layers does help to improve the prediction performance in general. Also, Ours-E2E improves about 7.0% more than Ours-no_image, which shows that the context information is indeed useful.

2) Qualitative Analysis: In Fig. 6, we show the feature extraction results of the self-supervised learning method. We use t-SNE to illustrate the relations between each extracted feature. It shows that similar pairs of road references are gathered into the same group, and the different ones are separated. We also find that the self-supervised procedure could also discover the similarities between the pictures from new coming scenarios, which means our self-supervised method has a good transferability.

TABLE V: Ablation of the Frenét Coordinate System and Self-Supervised Learning (mADE / mFDE).

methods	@3.0s	@4.0s	@5.0s
Ours	**0.139 / 0.284**	**0.268 / 0.562**	**0.432 / 0.913**
Ours-E2E	0.151 / 0.318	0.295 / 0.646	0.481 / 1.052
Ours-no_image	0.162 / 0.337	0.315 / 0.685	0.517 / 1.124
GNN	0.271 / 0.534	0.469 / 0.969	0.695 / 1.457

B. *Frenét-based Trajectory Prediction Ablation*

In this analysis, we compare GNN and Ours-E2E. Notice that GNN does not use the Frenét-based features and the self-supervised learning, so the only difference between GNN and Ours-E2E is whether the Frenét-based features are used. In Table V, we find that Ours-E2E improves about 30.8% mADE in the future 5s, which implies that our proposed Frenét-based trajectory representation improves the prediction performance remarkably.

VII. Conclusion and Future Work

In this work, a graph-neural-network-based framework with self-supervised domain knowledge is proposed to solve the multi-agent human driving behavior prediction problem. We incorporate human's prior knowledge and self-supervised learning techniques to enhance the generalizability and transferability across different traffic scenarios. Experiments demonstrate that our approach achieves significant improvement compared with other state-of-the-art methods. The ablative analysis demonstrates the effectiveness of the proposed feature representation technique. There are several future directions to investigate, such as how to incorporate more domain knowledge into this framework, including traffic rules and related knowledge about other types of traffic participants. Another interesting topic is how to design better pretext tasks for self-supervised learning.

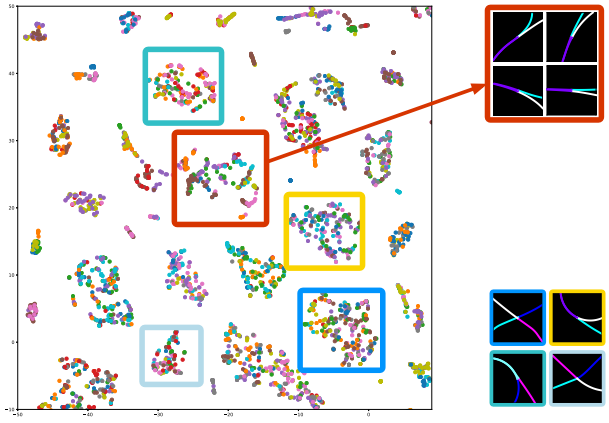


Fig. 6: The t-SNE illustration of context information. Different colors of scatters represent different pairs of references in all scenarios. We illustrate several groups of extracted features here and show their corresponding rasterized images. The red rectangle shows that the new pairs of references are clustered into the same group. The other rectangles show the differences between different groups of extracted features.

VIII. Implementation Details

In the attribute encoding layer, we use GRU as the cell of RNN to extract the historical trajectory information for nodes and edges. We use ResNet18 [37] to encode all the rasterized images. We concatenate those two features and use a MLP as function f, h in Sec. IV-C. The ResNet for context images I^{ij} is pretrained by contrastive learning with a batch size of 1024. The dimensions of the image representation and the latent variables are 32 and 16. For the message passing procedure, f_v^m and $f_{e,\alpha}^m$ are MLPs. We use different MLPs for different types of edges. The message passing number is 3. For the multi-modal decoder module,

the number of Gaussian kernels is 4, and we use Gumbel-Softmax to sample 20 trajectories to calculate the mADE and mFDE. f_w, f_μ, and f_Σ are also MLPs. All the MLP modules in our model are two-layer fully connected networks with an activation function of ReLU and a hidden size of 256. For the experiments of all scenarios, we mix all the data from five scenarios and divide it into 5:2:3 for training, validation, and testing. We train the model with a batch size of 64 for 100 epochs using Adam optimizer with an initial learning rate of 0.001. For the transferability experiments, the datasets of different scenarios are also divided into 5:2:3 for training, validation, and testing. The initial model is trained on MA+FT using the same hyperparameters as the all-scenarios experiments. We sample 100 cases from the training set of the new scenarios for the few-shot adaptation and fine-tune the model with a batch size of 20 and an initial learning rate of 0.0005. For all experiments, we show the average results of three random initializations.

REFERENCES

[1] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," *arXiv preprint arXiv:1802.04687*, 2018.

[2] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," in *Advances in neural information processing systems*, 2020.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[4] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

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

[6] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction and tracking," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[7] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[8] W. Wang, J. Xi, and D. Zhao, "Learning and inferring a driver's braking action in car-following scenarios," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3887–3899, 2018.

[9] H. Ma, J. Li, W. Zhan, and M. Tomizuka, "Wasserstein generative learning with kinematic constraints for probabilistic interactive driving behavior prediction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2477–2483.

[10] 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.

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

[12] A. Cui, A. Sadat, S. Casas, R. Liao, and R. Urtasun, "Lookout: Diverse multi-future prediction and planning for self-driving," *arXiv preprint arXiv:2101.06547*, 2021.

[13] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6150–6156.

[14] Y. Hoshen, "Vain: Attentional multi-agent predictive modeling," in *Advances in Neural Information Processing Systems*, 2017.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[16] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[17] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3960–3966.

[18] 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.

[19] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, "Graph neural networks for modelling traffic participant interaction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 695–701.

[21] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

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

[23] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Conference on Robot Learning*, 2018, pp. 947–956.

[24] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[27] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4182–4192.

[28] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2021.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[30] A. Srinivas, M. Laskin, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," *arXiv preprint arXiv:2004.04136*, 2020.

[31] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine, "Learning invariant representations for reinforcement learning without reconstruction," *arXiv preprint arXiv:2006.10742*, 2020.

[32] J.-C. Su, S. Maji, and B. Hariharan, "When does self-supervision improve few-shot learning?" in *European Conference on Computer Vision*. Springer, 2020, pp. 645–666.

[33] W. Zhan, J. Chen, C.-Y. Chan, C. Liu, and M. Tomizuka, "Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 632–639.

[34] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[35] 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.

[36] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," *arXiv preprint arXiv:2001.03093*, 2020.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.