Tracing the Influence of Predecessors on Trajectory Prediction

Mengmeng Liu, Hao Cheng, Michael Ying Yang
University of Twente, The Netherlands

mengmeng.liu1998@gmail.com; {h.cheng-2, michael.yang}@utwente.nl

Abstract

In real-world traffic scenarios, agents such as pedestrians and car drivers often observe neighboring agents who exhibit similar behavior as examples and then mimic their actions to some extent in their own behavior. This information can serve as prior knowledge for trajectory prediction, which is unfortunately largely overlooked in current trajectory prediction models. This paper introduces a novel Predecessor-and-Successor (PnS) method that incorporates a predecessor tracing module to model the influence of predecessors (identified from concurrent neighboring agents) on the successor (target agent) within the same scene. The method utilizes the moving patterns of these predecessors to guide the predictor in trajectory prediction. PnS effectively aligns the motion encodings of the successor with multiple potential predecessors in a probabilistic manner, facilitating the decoding process. We demonstrate the effectiveness of PnS by integrating it into a graph-based predictor for pedestrian trajectory prediction on the ETH/UCY datasets, resulting in a new state-of-the-art performance. Furthermore, we replace the HD map-based scene-context module with our PnS method in a transformer-based predictor for vehicle trajectory prediction on the nuScenes dataset, showing that the predictor maintains good prediction performance even without relying on any map information.

1. Introduction

Trajectory prediction is a fundamental problem in various fields, such as transportation, robotics, and surveillance. Trajectory predictors, typically, rely on observed trajectories of the target and neighboring agents to model their motion dynamics and interactions [1, 22, 16]. In addition, state-of-the-art trajectory prediction methods, especially for autonomous driving, often make use of other prior information, such as semantic maps, *e.g.*, [17, 37, 36, 51], high-definition (HD) maps, *e.g.*, [11, 19, 32, 57, 18, 44, 52, 41, 10, 34], or goal information, *e.g.*, [13, 12, 55, 14, 8].

Besides that, in real-world traffic scenarios, we often observe that pedestrians and car drivers follow the trace of
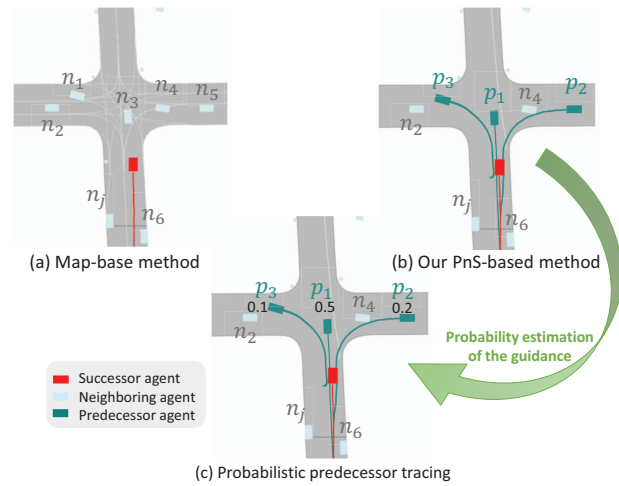

Figure 1. Compared to an HD-map based method (a) that relies on the lane segments for scene-aware trajectory prediction, our PnS-based method identifies predecessors, *i.e.* $\{p_1, p_2, p_3\}$, from the neighboring agents, *i.e.* $\{n_1, n_2, ..., n_j\}$, with similar moving patterns (b) and learns a probabilistic guidance of the predecessors influencing the successor agent (c). It should be noted that all the neighboring agents are still considered for interaction modeling in both methods.

neighbors who have conducted a similar behavior in the same scene, assuming that all the agents behave rationally, *e.g.*, complaint to scene constraints and avoid collisions. As shown in Fig. 1(b), given previously-captured frames in this toy scenario, the future trajectory of the target agent (red box) becomes more predictable if we can identify similar moving patterns in the same space that have demonstrated by other agents (dark green). This implies that trajectories observed from other agents are not only useful for interaction modeling, but to some extent also can be treated as an informative prior to facilitate the prediction.

However, valuable dynamic information from the neighboring agents within the same scene to guide the target agent is often overlooked by mainstream trajectory prediction methods that heavily rely on static scene information, as shown in Fig. 1(a). Although existing memory-based search techniques like [30, 47, 24] and clustering methods such as [38, 49, 31] use historical trajectories to iden-

tify similar moving patterns, they require extra efforts to accumulate the historical trajectories before carrying out the searching step. Also, there could be a huge time lag between the current target agent and the historical agents, which can seriously hinder the match of spatial and temporal relationships between the current and past traffic scenes which are associated with many changing factors.

Therefore, the research gap mentioned above motivates us to propose a novel Predecessor-and-Successor (PnS) method to explore the influence of neighboring agents (potential predecessors) on the guidance of the target agent (successor). Predecessors are defined as agents that have been concurrent in the same scene as the target agent and could potentially be part of its guide. In other words, in addition to modeling the interactions between the target and neighboring agents for collision avoidance, we also explicitly model the potential guidance of the predecessors identified from these neighboring agents on the target agent. As shown in Fig. 1 (b) and (c), PnS employs an attention-based probabilistic approach to identify predecessors and learn their influence on the target agent. For example, neighboring agents $\{n_3, n_5, n_1\}$ are identified as potential predecessors $\{p_1, p_2, p_3\}$ indexed by the rank of probabilities based on the alignment between their spatial and temporal relationships. Compared to the memory- and cluster-based methods, no extra efforts are needed to collect all the historical trajectories beforehand, and the time lag is much smaller from observing the predecessors to the successor following the predecessors' trace only after a few seconds.

To demonstrate the effectiveness of the PnS method for both pedestrian and vehicle trajectory predictions, we integrate PnS into a state-of-the-art graph-based predictor GATraj [7] for pedestrian trajectory prediction on the ETH/UCY [35, 23] datasets, and we replace the HD map-based scene-context module with PnS in another transformer-based predictor LAformer [26] that holds a high rank for vehicle trajectory prediction on the nuScenes [3] benchmark. The **main contributions** of this work are summarized as follows:

- Our work proposes a novel Predecessor-and-Successor (PnS) method to learn the probabilistic influence from neighboring agents on the target agent. It effectively explores the predecessors motion as prior information to guide the prediction of the successor's trajectory.

- We demonstrate that leveraging predecessor information further pushes the state-of-the-art performance for the pedestrian trajectory prediction on the ETH/UCY datasets. In the mapless-based setting for the vehicle trajectory prediction on the nuScenes dataset, the PnS method largely mitigates the performance degradation when the HD-map information is removed.

2. Related Work

Multimodal trajectory prediction. Due to the stochastic nature of road agents' future trajectories, arising from *e.g.*, their randomness, subjectivity of intent, mutual influence, and scene constraints, predicting multimodal trajectories has become a prevalent approach in trajectory prediction. Here, multimodal prediction refers to generate multiple plausible trajectories of the target agent. This trend is further motivated by large-scale benchmarks featuring real-world traffic scenarios [35, 23, 3, 5, 53, 46, 21]. Various methods address behavior uncertainty, including using Gaussian or Laplacian mixture models trained with Mixture Density Networks (MDNs) to estimate the likelihood of each mode [57, 37, 11, 10]. Another approach involves modeling multimodality implicitly via latent variables sampled from a prior distribution to generate diverse futures, including Variational Auto-Encoders (VAEs) [37, 51, 6], Generative Adversarial Networks (GANs) [16, 22], and Diffusion models [15, 29]. In this paper, we follow the mainstream approaches to demonstrate our PnS by integrating it into two representative multimodal trajectory predictors [7, 26]. To account for the multimodality of predictions, PnS employs a predecessor tracing module to identify multiple potential predecessors and learn a probabilistic guidance from each of the predecessors on the target agent to generate multiple predictions.

Interaction modeling. Effectively modeling the interactions among agents is crucial to account for their mutual influences. Many of the approaches only focus on the interactions between the target and neighboring agents in the observation time for interaction modeling [40]. Their motion encodings are aggregated by *e.g.*, pooling [1, 16, 48], message passing [54, 4, 24] via a multi-scale graph using Graph Convolutional Networks (GCNs) [45], and attention mechanisms [42] that focus on the salient spatial and temporal features of the encodings [51, 27, 6, 26, 18]. Our work goes beyond by modeling the interactions between the target and neighboring agents in the observation time using the attention mechanisms. We also condition the target agent's future trajectory on the historical trajectories by learning a probabilistic interactive influence of the predecessor agents. As we assume that these predecessors conducted their behavior rationally following the speed profiles complaint to scene constraints; The moving patterns derived from the predecessors in the same scene as the target agent is treated as a prior to better guide the prediction.

Prior information from historical trajectories. Memory-based methods [30, 47, 24] utilize historical trajectories to improve prediction performance. For example, [30] reads trajectories that are most likely to occur in future from stored observations in memory and uses them to augment the encodings of the target agent. Similarly, [47] employs a
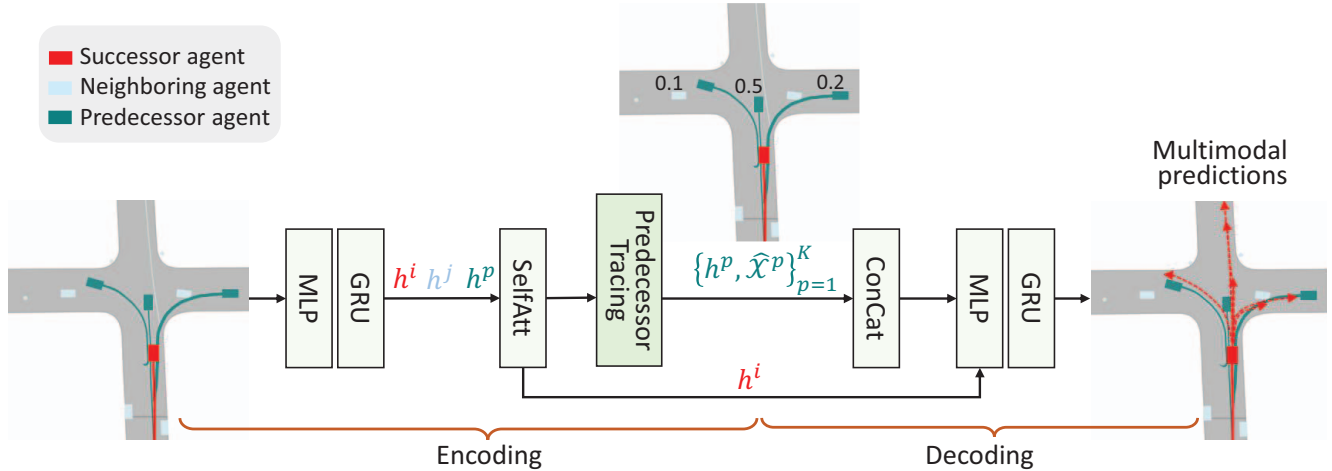
Figure 2. The framework of the Predecessor-and-Successor (PnS)-based trajectory prediction model consists of several components. PnS utilizes a stack of Multi-Layer Perceptron (MLP), Gated Recurrent Unit (GRU), and self-attention layers to extract motion encodings h^i, h^j, h^p of the successor, neighboring, and predecessor agents, respectively. The predecessor tracing module employs across-attention mechanisms to determine the probability of a predecessor, denoted as p, influencing the successor, denoted as i, among all the potential predecessors. This module learns the probabilities of predecessors' influence on the successor. To accommodate the stochastic behavior of the successor, the encoding of the top K predecessors, based on the ranking of their probabilities, is aggregated with the encoding of the successor. This aggregation facilitates the decoding of multiple future trajectories for the successor.

pair of memory banks to store representative instances from the training set, acting as the prefrontal cortex in the neural system. It also employs a trainable memory addresser to adaptively search for situations similar to current ones in the memory bank, acting as the "basal ganglia". [24] forecasts multiple paths based on historical trajectories by modeling multi-scale graph-based spatial transformers combined with a trajectory smoothing algorithm named "Memory Replay" using a memory graph. Moreover, cluster-based methods aggregate moving patterns, such as grouping target agent-neighbors [38, 49] and focusing on historical group trajectories [31], for trajectory prediction. Also, [39] clusters the trajectories into a fixed number of categories for trajectory classification. However, there could be a huge time lag between the current and past traffic scenes that are associated with many changing factors, like traffic density and scene contexts. This makes it difficult for the existing memory-based search techniques and clustering methods to distinguish the impact of historical trajectories on the target agent, taking into account its current traffic situation and motion dynamics. Moreover, these methods require to accumulate historical trajectories to build up the memory base beforehand, which causes extra burden and may limit their application in new scenes. In contrast, we propose a probabilistic Predecessor-and-Successor method that leverages trajectories of the neighboring agents to identify representative predecessors as reference trajectories for the target agent. Because the target and neighboring agents appear at the same time and share the same scene, their spatial and temporal relationships can be better modeled.

3. Method

3.1. Problem formulation

Following mainstream methods [11, 14, 57, 41], we define the motion forecasting problem as predicting the subsequent trajectory $Y^i_{1:t_f}$ of a target agent i given the set of observed trajectories $\mathbf{X}$ of a total of N agents in the same scenario, including both the target and neighboring agents. To differentiate the target and neighboring agents, we use $X^i_{t_h-1:0}$ to denote the observed trajectory of target agent i and $\hat{\mathbf{X}} = \mathbf{X} \setminus X^i$ the remaining set of the observed trajectories of neighboring agents. The observation time horizon is $\{t_h - 1, \ldots, 0\}$ and the future time horizon is $\{1, \ldots, t_f\}$. To simplify the notation, time steps are omitted if they are not otherwise noted, and we call the target agent *successor* in the remaining of this paper.

To fully exploit the observed trajectories of the neighboring agents $\hat{\mathbf{X}}$, a predecessor tracing function $g(\hat{\mathbf{X}}, X^i)$ learns a probabilistic guidance from the predecessors on the successor i. Consequently, the prediction $\hat{Y}^i = f(\mathbf{X}, g(,))$ conditions $\hat{Y}$ on $\mathbf{X}$ and $g(,)$. Compared to many previous trajectory predictors formulated as $\hat{Y}^i = f(\mathbf{X})$, the only added component is $g(,)$, which does not require any extra input data. Hence, this predecessor tracing can be easily incorporated into these predictors.

3.2. Predecessor tracing

To effectively trace the impact of predecessors on the successor, we propose a time step-wise relationship predictor to identify their spatial relationship and use attention

mechanisms to learn the impact between them.

In order to model the influence from predecessors, we propose to use a cross-attention mechanism to design the interaction mapping function $\pi(\,,\,)$. Specifically, we use linear projections to transform predecessor p's motion encoding h^p into a query vector Q, and the successor's motion encoding h^i into key (K) and value (V) vectors. These vectors are then used as inputs to a scaled dot-product attention block to calculate the attention score $s^{i,p}$.

$$s^{i,p} = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \qquad (1)$$

where d_k is the dimensionality of the key vectors. For more details on the encodings, refer to Sec. 3.3.

The influence from the predecessor p to successor i is then mapped by taking as inputs the concatenation of $(h^i, h^p, s^{i,p})$. Here, we implement a Multi-Layer Perceptron (MLP) for the mapping. The output of the MLP is denoted by $\pi(h^i, h^p)$.

$$\pi(h^i, h^p) = \text{MLP}(h^i, h^p, s^{i,p}). \qquad (2)$$

Rather than using a deterministic approach to model the influence of predecessors, we propose a probabilistic approach that better considers the stochasticity of the successor's behavior. Specifically, we use a Softmax function to calculate the probability of predecessor p influencing the successor i at time step t among all the potential $N \setminus i$ predecessors.

$$\hat{\mathcal{X}}_t^p = g(X^p, X^i) = \frac{\exp(\pi(h_t^i, h_t^p))}{\sum_{n=1}^{N \setminus i} \exp(\pi(h_t^i, h_t^n))}. \qquad (3)$$

3.3. Trajectory encoding and decoding

In this section, we describe the trajectory encoding and decoding process of our proposed model. Similar to previous attention-based models [51, 27, 6, 26, 18], we utilize the self-attention mechanism to encode the motion dynamics of the successor, neighboring, and predecessor agents, and decode the future trajectories of the successor, as shown in Fig. 2.

Encoding. We start by extracting the motion encodings of the successor, neighboring, and predecessor agents using a MLP followed by a Gated Recurrent Unit (GRU).

$$h_t^o = \text{GRU}(\text{MLP}(X_t^o)), \qquad (4)$$

where $o = \{i, j, p\}$ denotes the index of the successor, neighboring, or predecessor agent, and $j, p \in N \setminus i$.

Next, we aggregate the interaction information among these three types of agents using the self-attention mechanism, as shown in Eq. (5):

$$h_t^o = h_t^o + \text{SelfAtt}(h_t^o), \qquad (5)$$

where SelfAtt represents the self-attention function. In this step, the pair-wise interaction information among them is learned by the attention and added to their original encodings via the skip connection. In this way, their interconnections are aggregated.

Decoding. To generate multimodal predictions for the successor, we implement a Laplacian Mixture Density Network (MDN) decoder following [57, 26].

First, instead of focusing on the impact of a single predecessor, we include multiple potential predecessors to enable the decoder to mimic the successor's stochastic behavior. For example, the predecessors may turn into different directions at an intersection or drive at different speeds. To achieve this goal, we select K predecessors based on their probability score $\hat{\mathcal{X}}_t^p$ in a descending order and aggregate these predecessors' motion encodings, as shown in Eq. (6):

$$\hat{\mathcal{X}}_t^K = \text{ConCat}[\{(h_t^p, \hat{\mathcal{X}}_t^p)\}_{p=1}^K]. \qquad (6)$$

After that, the decoder takes as input the motion encoding h_t^i of the successor and $\hat{\mathcal{X}}_t^K$ of the K most influencing predecessors and outputs the distributions of the successor's future positions. More specifically, The outputs are parameterized by the location μ^m and scale b^m parameters of a total of M components, where $m \in M$ and M corresponds to the different modalities of the predictions.

Similar to the encoding process, we use a MLP layer followed by a GRU layer to implement the decoder. It outputs the location and scale parameters of each component of the MDN, as well as the associated likelihood π_m.

3.4. Incorporating with the existing models

To analyze the effectiveness of the proposed PnS method, we incorporated it into two trajectory predictors – GATraj [7] and LAformer [26]. These two models are chosen based on the following reasons: First, both GATraj and LAformer are one of the latest models and have shown excellent performances on the ETH/UCY datasets for pedestrian trajectory prediction and nuScenes dataset for vehicle trajectory prediction, respectively; Second, they apply different mainstream frameworks for trajectory prediction. Namely, GATraj exploits a graph-based framework with massage passing, while LAformer employs the transformer framework with attention mechanisms, to learn spatial and temporal information for trajectory prediction. These two different models are representative examples to demonstrate the compatibility of the PnS method. Third, both models use a MDN decoder, the PnS method can be smoothly incorporated into the backbones and jointly trained without drastic modification. It should be noted that because the original LAformer heavily relies on lane segments from an HD map as a strong prior to guide the prediction. Hence, we substitute the lane alignment module in LAformer with

our PnS method to guide the prediction.

3.5. Training

The predecessor tracing module is optimized using the binary cross-entropy loss $\mathcal{L}_{\text{PnS}}$ to improve the probability estimation. The loss function is defined as follows:

$$\mathcal{L}_{\text{PnS}} = \sum_{t=1}^{t_{\text{f}}} \mathcal{L}_{\text{CE}}(\mathcal{X}_t^p, \hat{\mathcal{X}}_t^p), \tag{7}$$

where $\mathcal{X}_t^p$ and $\hat{\mathcal{X}}_t^p$ denote the ground truth and predicted probability of predecessor p influencing the successor i at time step t, respectively.

In the training phase, we identify the predecessor that has the closest spatial relationship and moving patterns with the successor at each time step. This is achieved by identifying predecessor p from $\hat{\mathbf{X}}$ using a distance metric ϕ.

$$\mathcal{X}_t^p = \begin{cases} 1 & \text{if } p = \underset{p \in N \setminus i}{\arg\min} \; \phi(Y_t^i, \hat{X}), \\ 0 & \text{otherwise.} \end{cases}$$

Specifically, we use the L_2 distance metric to find the predecessor that is closest to the successor at time step t. The identified predecessor is labeled as the true predecessor, while all other agents from $\hat{\mathbf{X}}$ are labeled as false predecessors.

Following previous works [28, 57, 10], we optimize the decoder in both GATraj and LAformer using the negative log-likelihood (NLL) of the best predicted mode m^* of the Laplacian MDN.

$$\mathcal{L}_{\text{NLL}} = \frac{1}{t_f} \sum_{t=1}^{t_f} -\log P(Y_t | \mu_t^{m^*}, \mathbf{b}_t^{m^*}), \tag{8}$$

where μ^{m^*} and b^{m^*} are the location and scale parameters of the component, respectively, and m^* represents the mode with the minimum L_2 error of the predicted and ground truth trajectories among the total M components. We utilize the cross-entropy to optimize the mode classification.

$$\mathcal{L}_{\text{cls}} = \sum_{m=1}^{M} -\pi_m \log(\hat{\pi}_m), \tag{9}$$

where π_m denotes the target probability of the mode. This target probability is defined by a soft displacement function, using the same method proposed in [57].

The overall objective function is formulated as:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{PnS}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{NLL}}, \tag{10}$$

where λ is a hyperparameter that controls the weights of loss terms in the objective function, allowing us to balance their respective contributions.

4. Experiments

4.1. Experimental setup

Dataset. We utilize the ETH/UCY datasets [35, 23] to train and test our PnS method using GATraj [7] for pedestrian trajectory prediction. These datasets consist of multiple subsets captured at different locations, each with varying pedestrian densities, including *Eth*, *Hotel*, *Uni*, *Zara1*, and *Zara2*. Following the most common setting [1, 16, 37, 51, 31], the trajectories are down-sampled to a frequency of 2.5 Hz, with an observation time horizon of 3.2 s and a prediction time horizon of 4.8 s. For training and testing, we follow the standard leave-one-out data partitioning to train the models on four out of these five subsets and test them on the remaining one. This procedure is repeated for each subset. Moreover, we employ the nuScenes dataset [3] for training and testing our PnS method using LAformer [26] for vehicle trajectory prediction. This dataset encompasses various driving scenarios involving complex intersections and interactions with pedestrians, cyclists, and other elements. It comprises a total of 245,414 trajectory instances across 1,000 driving scenes, each lasting 20 seconds and sampled at 2Hz. The observation time horizon is set to two seconds, while the prediction time horizon is set to six seconds. To facilitate offline training and validation, 850 scenes are provided with ground truth information, while the remaining 150 scenes are reserved for online testing.

Evaluation metrics. We adhere to standard evaluation metrics to assess the prediction performance [1, 3, 57]. Specifically, we employ displacement errors (DE) to evaluate the performance on the ETH/UCY and nuScenes datasets. In particular, we utilize two metrics: mFDEK and mADEK, which measure the minimum L_2 errors in meters at the Final step and the Average of each step, respectively, for predicting K modes. The letter "m" signifies the minimum error among the K modes. In the case of nuScenes, we set K to five or ten, while for ETH/UCY, it is set to 20. It is important to note that for all evaluation metrics, a lower value indicates better performance.

Implementation details. To ensure compatibility between PnS and GATraj [7] and LAformer [27], we set the hidden dimensions of all feature vectors to match the original configuration. We set the predecessor number K to 2, aiming to strike a balance between the number of predecessors and the diversity of predictions. The value of λ is set to 0.5, which helps maintain a balance among the loss terms in the objective function indicated by Eq. (10). For the activation function of the intermediate layers, we use the Rectified Linear Unit (ReLU). In the decoder, we employ the Exponential Linear Unit (ELU), specifically $\text{ELU}(.) + 1 + \epsilon$, as the activation function to generate positive probability estimations. Here, ϵ is set to $1e^{-3}$. All the models were trained on eight RTX 3090 GPUs using the Adam optimizer [20].

4.2. Quantitative results and comparison

To effectively demonstrate the performance enhancement achieved by incorporating the PnS method into GATraj and LAformer, we present a comparison of their performance with and without PnS in Tables 1 and 2.

Table 1 reveals that GATraj+PnS exhibits improved performance on Eth, Uni, and Zara2, as measured by FDE_{20}. Despite GATraj already yielding minimal errors in each subset, the addition of our PnS method further reduces prediction errors, particularly in datasets with high pedestrian density such as Uni by 2.6% in $mFDE_{20}$ and Zara2 by 8.3% and 4.8% in $mADE_{20}$ and $mFDE_{20}$, respectively. This is because the predecessor tracing module can derive information cues from more neighboring agents.

Models	Eth	Hotel	Uni	Zara1	Zara2
GATraj [7]	0.26/0.42	0.10/0.15	0.21/0.38	0.16/0.28	0.12/0.21
GATraj [7]+PnS	0.26/0.40	0.10/0.15	0.21/0.37	0.16/0.28	0.11/0.20
Improvement	-/4.7%	-/-	-/2.6%	-/-	8.3%/4.8%

Table 1. Quantitative results on ETH/UCY [35, 23] measured by $mADE_{20}/mFDE_{20}$.

Table 2 showcases the performance of PnS integrated into LAformer on the nuScenes dataset. The comparison between LAformer and LAformer+PnS reveals that LAformer's performance is significantly compromised when it relies solely on observed trajectories without any map information. However, the inclusion of the PnS method greatly mitigates this issue, resulting in a reduction of approximately 5.1% in $mADE_5$ and 9.8% in $mADE_{10}$. This advantage indicates that the PnS method offers a practical alternative for vehicle trajectory prediction in scenarios where map information is unavailable, such as when a vehicle enters a new location without access to HD map data.

Interestingly, we also observe that providing both HD map and predecessor information to LAformer does not yield a combined improvement. Our conjecture is that the HD map already contains rich contextual details, including lane segments, road geometry, and traffic rules. Consequently, when we simply concatenate these two types of information, the PnS method may not exert a strong influence on the successors in this specific setup.

Model	$mADE_5$	$mADE_{10}$
LAformer [27]	1.57	1.32
LAformer [27]+PnS	1.49	1.19
Improvement	5.1%	9.8%
LAformer [27]+HD	1.19	0.93
LAformer [27]+HD+PnS	1.20	0.93
Improvement	-0.8%	-

Table 2. The results on the *nuScenes* [3] test set.

Furthermore, we conduct a benchmark comparison of GATraj+PnS and LAformer+PnS with current models in Tables 3 and 4 for pedestrian and vehicle trajectory prediction, respectively.

In Table 3, we compare GATraj+PnS with the most recent top models. Among the other models, particularly the Retrospective-Memory-based *MemoNet* [47] and the scene history-based *SHENet* [31] share similarities with the PnS concept to use existing trajectories as prior information. We also include the latest diffusion-based *LED* [29] that holds the top performance for this task. GATraj+PnS achieves the best performance across all subsets, except for Zara1 in terms of FDE_{20} compared to *LED*. This sets a new state-of-the-art performance on the ETH/UCY datasets for pedestrian trajectory prediction, even without the effort to accumulate historical trajectories for building up a memory base.

Models	Eth	Hotel	Uni	Zara1	Zara2
Social-GAN[16]	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84
Trajectron++[37]	0.67/1.18	0.18/0.28	0.30/0.54	0.25/0.41	0.18/0.32
STAR [50]	0.36/0.65	0.17/0.36	0.31/0.62	0.26/0.55	0.22/0.46
AgentFormer [51]	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24
MID [15]	0.39/0.66	0.13/0.22	0.22/0.45	0.17/0.30	0.13/0.27
LB-EBM [33]	0.30/0.52	0.13/0.20	0.27/0.52	0.20/0.37	0.15/0.29
PCCSNet [39]	0.28/0.54	0.11/0.19	0.29/0.60	0.21/0.44	0.15/0.34
GP-Graph [2]	0.43/0.63	0.18/0.30	0.24/0.42	0.17/0.31	0.15/0.29
MemoNet [47]	0.40/0.61	0.11/0.17	0.24/0.43	0.18/0.32	0.14/0.24
SHENet [31]	0.41/0.61	0.13/0.20	0.25/0.43	0.21/0.32	0.15/0.26
LED [29]	0.39/0.58	0.11/0.17	0.26/0.43	0.18/**0.26**	0.13/0.22
GATraj [7]+PnS	**0.26/0.40**	**0.10/0.15**	**0.21/0.37**	**0.16**/0.28	**0.11/0.20**

Table 3. The comparison on the ETH/UCY datasets measured by ADE_{20}/FDE_{20}. The best performance is in boldface.

We compare LAformer with PnS to recent models on the nuScenes dataset. To ensure a fair comparison, we ensure that all models utilize the same input data. First, we compare LAformer+PnS with AgentFormer [51] and SG-Net [43], as all these models solely rely on observed trajectories without incorporating map information. Next, we compare PnS with LBA [56], which utilizes a Local Behavior-Aware module to leverage historical trajectories that have previously traversed the scene. This module bears resemblance to PnS in the way it explores existing trajectories to guide predictions. Additionally, since the backbone models of LBA[1], namely P2T [9] and LaneGCN [25], also incorporate HD map information, we compare them with LAformer+PnS+HD. As demonstrated in Table 4, LAformer+PnS exhibits significantly better performance compared to AgentFormer and SG-Net when the HD map is removed. In the map-based setting, LAformer+PnS+HD outperforms P2T+LBA+HD and achieves comparable performance to LaneGCN+LBA+HD. It is worth mentioning that P2T+LBA+HD and LaneGCN+LBA+HD require access to historical trajectories, which necessitates pre-accumulation of these trajectories. In contrast,

---

[1]We were unable to implement LBA with LAformer and directly compare it with LAformer+PnS+HD due to the broken link to the code of [56].

P2T+LBA+HD directly traces predecessors from concurrent neighboring agents within the same scene without extra input data in a more challenging yet practical setting.

Model	mADE$_5$	mADE$_{10}$	mFDE$_1$
AgentFormer [51]	1.97	1.58	-
SG-Net [43]	1.86	1.40	-
LAformer [27]+PnS	**1.49**	**1.19**	8.08
P2T [9]+LBA [56]+HD	-	1.08	9.25
LaneGCN [25]+LBA [56]+HD	-	0.95	**6.78**
LAformer [27]+PnS+HD	**1.20**	**0.93**	6.99

Table 4. The prediction comparison on the nuScenes [3] test set. The best values are highlighted in boldface.

4.3. Ablation study

To analyze the effectiveness of the proposed components in the PnS method, we conduct an ablation study using the train-val split provided by the nuScenes dataset.

Firstly, we ablate the predecessor tracing (PT) module to evaluate its efficacy in our proposed model, denoted as PnS-Baseline. In this setting, the prediction model can only condition future trajectories on the observed trajectories. As shown in Table 5, removing the PT module results in a significant drop in performance, about 5% in both mADE$_5$ and mFDE$_5$ for predicting five trajectories, and approximately 11% in mADE$_{10}$ and 17% in mFDE$_{10}$ for predicting ten trajectories.

PT	mADE$_5$	mFDE$_5$	mADE$_{10}$	mFDE$_{10}$
$\checkmark$	1.49	3.10	1.19	2.24
-	1.57	3.28	1.32	2.63

Table 5. Ablation study on PnS.

Secondly, we analyze the number of predecessors aggregated via Eq. (6) for the predecessor tracing module, as shown in Table 6. With a single predecessor, the model performs slightly worse than with two predecessors because the successor can be influenced by other predecessors as well. However, increasing the number of potential predecessors to three leads to a clear performance drop, as measured by mFDE$_{10}$. This is because the chances of including less influential or even irrelevant predecessors also increase.

k	mADE$_{10}$	mFDE$_{10}$
1	1.19	2.27
2	1.19	2.24
3	1.21	2.32

Table 6. The number of top predecessors.

Thirdly, we explore another distance metric in ϕ for predecessor identification. Namely, we substitute L_2 by L_1 in ϕ. As shown in Table 7, L_1 does not work as well as L_2. It increases, *e.g.* mFDE$_{10}$, by about 3%.

Distance	mADE$_{10}$	mFDE$_{10}$
L_1	1.24	2.43
L_2	1.22	2.36

Table 7. Distance metric for predecessor identification.

Finally, we investigate the sensitivity of the weight parameter λ in the loss function presented in Eq. (10). We vary the value of λ while keeping other settings unchanged to analyze its effect on the prediction performance. We observe that decreasing or increasing this value to 0.1 or 1 leads to slightly worse performance in terms of mADE$_{10}$ and mFDE$_{10}$, as shown in Table 8.

λ	mADE$_{10}$	mFDE$_{10}$
0.1	1.25	2.45
0.5	1.22	2.36
1	1.28	2.53

Table 8. Sensitivity analysis of λ in the objective function.

4.4. Qualitative results

Fig. 3 illustrates the qualitative results obtained by LAformer, LAformer+PnS, and LAformer+HD. In the first row, a vehicle is seen traversing an eight-arm intersection. LAformer, without any scene cues, generates a right turn that is incompatible with lane connections, and drives through areas outside lane boundaries. In contrast, LAformer+PnS follows traces from predecessors and produces more accurate predictions. With the aid of lane centerlines, LAformer+HD accurately captures the vehicle's driving intent. The differences among these three models become more apparent in the second row, as the vehicle drives along on the road. LAformer generates several predictions that are incompatible with the scene, while the predictions of LAformer+PnS and LAformer+HD are more compliant with the scene. We also observe that LAformer+PnS generates more divergent predictions in lateral directions in the third row, as it may follow the traces of predecessors turning into different directions in this traffic situation. Interestingly, in the fourth row, LAformer+PnS predicts a turning modality that is well overlapped with the ground truth trajectory, while LAformer fails to predict the vehicle's intent, and the predictions of LAformer+HD have an offset from the ground truth trajectory. However, we also note that some modalities predicted by LAformer+PnS are not perfectly aligned with the road geometry due to the lack of lane information.

4.5. Discussion

To avoid false positive predecessor identification, for example, predecessors are too far away from the successor or the angles from the successor to the predecessors are too sharp to make a maneuver, we have tested setting a maximum distance threshold and angle difference. This ensures that the identified predecessor agent is spatially close
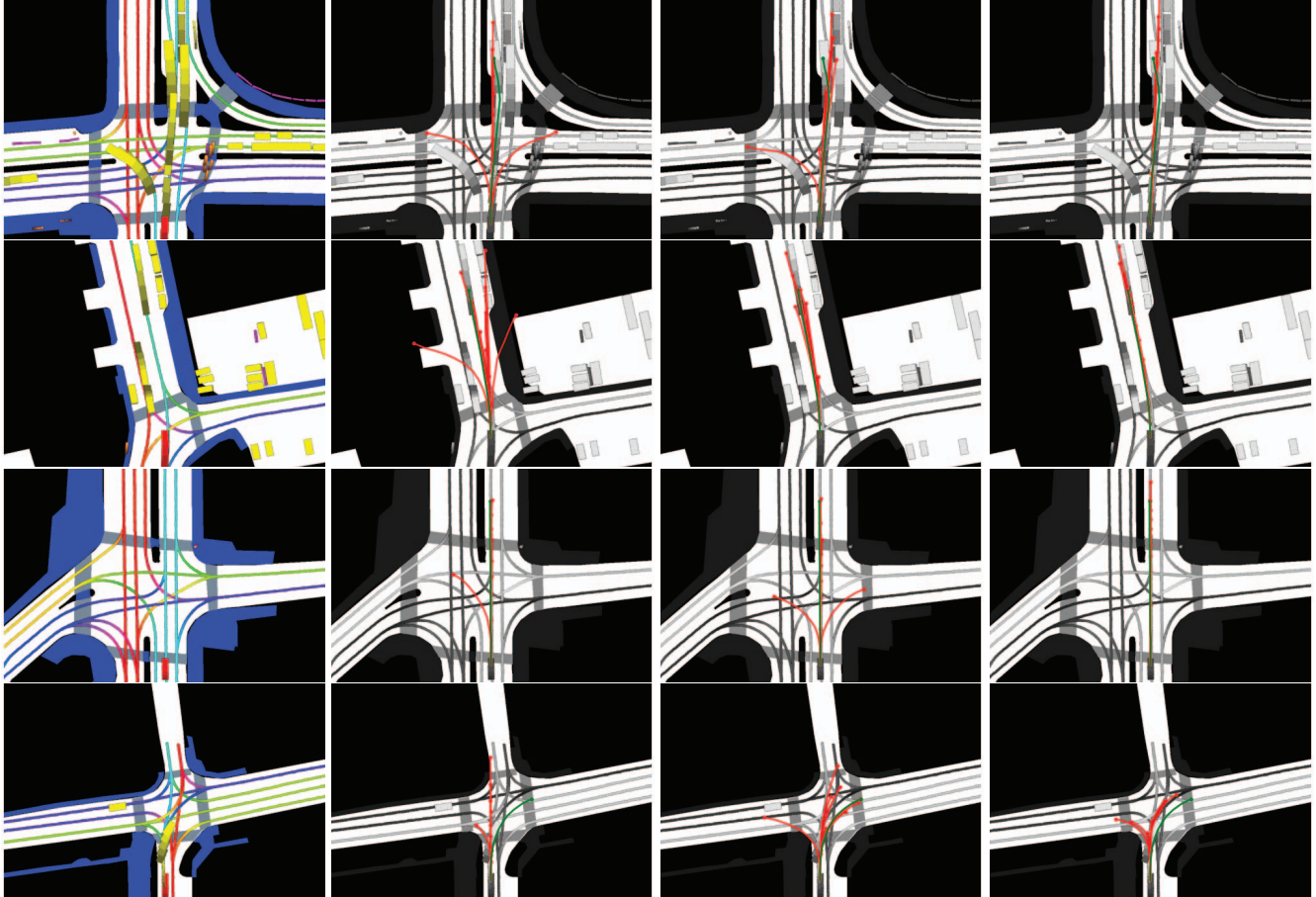
Figure 3. The qualitative comparison of the prediction results on the nuScenes [3] validation set. From left to right columns: **a)** traffic situation, **b)** LAformer without the PnS and HD map information, **c)** LAformer+PnS that uses the predecessor tracing module to guide the predictions, **d)** LAformer+HD that uses the HD map information to guide the predictions. The predictions are in red and the corresponding ground truth trajectories are in green.

to the successor and driving in a feasible direction. However, as a limitation in some cases, the successor may not find any predecessor that meets the criteria, resulting in a lack of observed predecessor information. In such cases, the prediction rolls back to conditioning on the past trajectories with an empty predecessor. The empirical studies on the nuScenes dataset indicate that even with a few those cases, this did not lead to a significant performance difference (more details in supplementary material). Overall, the prediction model achieves evidently better performance by including the PnS with the predecessor tracing module. We leave exploration of more sophisticated solutions, such as knowledge distillation [56], to address the cases of no HD map and no predecessors for future work.

5. Conclusion

In this paper, we propose a probabilistic approach named Predecessor-and-Successor (PnS) to trace the influence of agents (predecessors) identified from the neighboring agents within the same scene on the target agent (successor). The moving patterns derived from the traced prede-cessors serve as informative priors to guide the prediction of the successor's movement. We use a probabilistic predecessor tracing module to select several highly influential predecessors to account for the stochasticity of the successor's future behavior. Compared to memory-based models, our method requires no extra effort to collect all the trajectory data beforehand. Our simple but effective PnS is integrated into a graph-based predictor for pedestrian trajectory prediction on the ETH/UCY datasets, which achieves a new state-of-the-art performance. Furthermore, we demonstrate the effectiveness of PnS by replacing the HD-map extraction module with PnS in a transformer-based predictor for vehicle trajectory prediction on the nuScenes dataset. Our method largely mitigates the performance degradation when the map information is removed. It also achieves comparable performance when it is compared with a Local Aware module that exploits the historical trajectories traversed in the same location.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[2] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Learning pedestrian group representations for multi-modal trajectory prediction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 270–289. Springer, 2022.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9491–9497. IEEE, 2020.

[5] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[6] Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Scept: Scene-consistent, policy-based trajectory predictions for planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17103–17112, 2022.

[7] Hao Cheng, Mengmeng Liu, Lin Chen, Hellward Broszio, Monika Sester, and Michael Ying Yang. GATraj: A graph- and attention-based multi-agent trajectory prediction model. *arXiv preprint arXiv:2209.07857*, 2022.

[8] Luigi Filippo Chiara, Pasquale Coscia, Sourav Das, Simone Calderara, Rita Cucchiara, and Lamberto Ballan. Goal-driven self-attentive recurrent networks for trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2518–2527, 2022.

[9] Nachiket Deo and Mohan M Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint arXiv:2001.00735*, 2020.

[10] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *Conference on Robot Learning*, pages 203–212. PMLR, 2022.

[11] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020.

[12] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Thomas: Trajectory heatmap output with learned multi-agent sampling. In *International Conference on Learning Representations*, 2021.

[13] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. In *2022 International Conference on Robotics and Automation*, pages 9107–9114. IEEE, 2022.

[14] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[15] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022.

[16] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[17] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8454–8462, 2019.

[18] Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *2022 International Conference on Robotics and Automation*, pages 2605–2611. IEEE, 2022.

[19] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *arXiv preprint arXiv:2008.10587*, 2020.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[21] John Lambert and James Hays. Trust, but verify: Cross-modality fusion for hd map change detection. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[22] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.

[23] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[24] Lihuan Li, Maurice Pagnucco, and Yang Song. Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2231–2241, 2022.

[25] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, pages 541–556. Springer, 2020.

[26] Mengmeng Liu, Hao Cheng, Lin Chen, Hellward Broszio, Jiangtao Li, Runjiang Zhao, Monika Sester, and Michael Ying Yang. Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. *arXiv preprint arXiv:2302.13933*, 2023.

[27] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021.

[28] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019.

[29] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. *arXiv preprint arXiv:2303.10895*, 2023.

[30] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Multiple trajectory prediction of moving agents with memory augmented networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[31] Mancheng Meng, Ziyan Wu, Terrence Chen, Xiran Cai, Xiang Zhou, Fan Yang, and Dinggang Shen. Forecasting human trajectory from scene history. *Advances in Neural Information Processing Systems*, 35:24920–24933, 2022.

[32] Sriram Narayanan, Ramin Moslemi, Francesco Pittaluga, Buyu Liu, and Manmohan Chandraker. Divide-and-conquer for lane-aware diverse trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15799–15808, 2021.

[33] Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. Trajectory prediction with latent belief energy-based model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11814–11824, 2021.

[34] Daehee Park, Hobin Ryu, Yunseo Yang, Jegyeong Cho, Jiwon Kim, and Kuk-Jin Yoon. Leveraging future relationship reasoning for vehicle trajectory prediction. In *International Conference on Learning Representations*, 2023.

[35] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[36] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020.

[37] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020.

[38] Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 660–669, 2020.

[39] Jianhua Sun, Yuxuan Li, Hao-Shu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13250–13259, 2021.

[40] Qiao Sun, Xin Huang, Junru Gu, Brian C Williams, and Hang Zhao. M2i: From factored marginal trajectory prediction to interactive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6543–6552, 2022.

[41] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation*, pages 7814–7821. IEEE, 2022.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[43] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022.

[44] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17134–17142, 2022.

[45] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[46] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[47] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2022.

[48] Yanyu Xu, Zhixin Piao, and Shenghua Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5275–5284, 2018.

[49] Hao Xue, Du Q Huynh, and Mark Reynolds. Poppl: Pedestrian trajectory prediction by lstm with automatic route class clustering. *IEEE transactions on neural networks and learning systems*, 32(1):77–90, 2020.

[50] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020.

[51] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.

[52] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 532–539. IEEE, 2021.

[53] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud 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.

[54] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12085–12094, 2019.

[55] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021.

[56] Yiqi Zhong, Zhenyang Ni, Siheng Chen, and Ulrich Neumann. Aware of the history: Trajectory forecasting with the local behavior data. In *European Conference on Computer Vision*, pages 393–409. Springer, 2022.

[57] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8823–8833, 2022.