

FIMP: Future Interaction Modeling for Multi-Agent Motion Prediction

Sungmin Woo, Minjung Kim, Donghyeong Kim, Sungjun Jang, Sangyoun Lee*

Abstract—Multi-agent motion prediction is a crucial concern in autonomous driving, yet it remains a challenge owing to the ambiguous intentions of dynamic agents and their intricate interactions. Existing studies have attempted to capture interactions between road entities by using the definite data in history timesteps, as future information is not available and involves high uncertainty. However, without sufficient guidance for capturing future states of interacting agents, they frequently produce unrealistic trajectory overlaps. In this work, we propose Future Interaction modeling for Motion Prediction (FIMP), which captures potential future interactions in an end-to-end manner. FIMP adopts a future decoder that implicitly extracts the potential future information in an intermediate feature-level, and identifies the interacting entity pairs through future affinity learning and top- k filtering strategy. Experiments show that our future interaction modeling improves the performance remarkably, leading to superior performance on the Argoverse motion forecasting benchmark.

I. INTRODUCTION

Accurate motion prediction is one of the essential requirements for safe and robust autonomous driving. Anticipating the near future enables a thorough understanding of the surrounding contexts and serves as the fundamental grounds for automated decision-making. However, it remains a challenge because the ambiguous intentions of dynamic agents involve high uncertainty and their behaviors are considerably affected by environmental constraints, such as kinematic states of neighboring agents, map topology and traffic rules. It is not straightforward to precisely capture all potential interactions between those factors in complex multi-agent scenarios.

Recent deep learning approaches have proposed diverse solutions for interaction modeling. Raster-based methods [1]–[3] rasterize the scene information as a multi-channel image from a top-down view, and encode the local interactions by using off-the-shelf 2D convolutional neural networks (CNNs). Graph-based methods [4]–[6] employ a vectorized representation that organizes data as polylines, and apply graph neural networks (GNNs) to learn the flow of information between nodes. Attention mechanisms are also extensively utilized in numerous methods [4]–[11] to better capture long-term interactions by modeling the relationships between entities (e.g., trajectory waypoints, lane segments) in spatial and temporal aspects.

As the available data given for motion prediction is from past timesteps, the interaction modeling of most approaches is naturally designed to focus on the observed historical information. The interaction is usually computed between

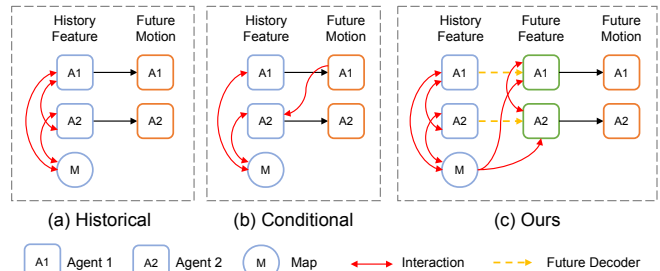


Fig. 1. Overview of interaction modeling in motion prediction. (a) Observed-historical-information-based interaction. (b) Estimated high-level future-states-based conditional prediction. (c) Our feature-level potential future information based interaction.

entity features extracted from the observed data (i.e., history feature) as illustrated in Figure 1 (a). However, this architecture lacks sufficient guidance for modeling interactions in future timesteps as message passing between predicted future motions is not significantly focused and performed. Some works [11]–[14] address this issue by solving the conditional prediction, which is to predict the motion conditioned on the estimated future states of interacting agents (Figure 1 (b)). This intuitive approach can take into account future interactions directly, but there remain three drawbacks. (1) As explicit high-level future information (e.g., future trajectory, goal point) is required, the interaction modeling heavily relies on the pre-estimated future motions of other agents. The accuracy of pre-estimation determines the reliability of considered interaction. (2) The conditional prediction often neglects the mutual influence between interacting agents. It lacks the ability to predict the motions of multi-agents jointly when both behaviors of agents are influenced by each other. (3) Owing to the additional procedure to estimate and refine the motion, the entire prediction process is inefficient and not suitable for real-time applications.

To alleviate these problems, we propose Future Interaction modeling for Motion Prediction (FIMP), which learns to capture a future interaction in an end-to-end manner. Instead of using the pre-estimated high-level future information, FIMP utilizes the features that implicitly contain the potential future information (i.e., future feature). As shown in Figure 1 (c), we decouple future features from the history feature for each agent, enabling to model the mutual future interaction without high-level cues as well as history interaction.

Specifically, we derive future features by adopting an intermediate future decoder comprised of a multi-head projection layer and a gated recurrent unit (GRU) [15]. The multi-head projection layer extracts disparate future mode embeddings from the history feature and GRU temporalizes

*Corresponding author.

The authors are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea.

each mode embedding into specific future time zones, which are the small time chunks split from entire prediction time. The acquired zone-wise future feature is then optimized to precisely determine when and where agents will be in the corresponding time period by learning. In addition, pairs of interacting agents are identified without prior knowledge of the explicit future states of agents in our approach. Instead, FIMP learns to extract affinity between future features of agents and selects the agent pairs with top- k high affinities for message passing. This future affinity learning and top- k filtering strategy enables proximity in feature space to represent the potential relationships between future positions. Extensive experiments on the large-scale Argoverse motion forecasting dataset show that our approach captures the future interaction properly and leads to superior performance in multi-agent motion prediction.

II. RELATED WORK

In complex traffic scenarios as in urban areas, diverse interactions occur between multiple entities simultaneously. As the future behaviors of agents are considerably affected by neighboring entities, it is crucial to capture the interactions between them in spatial as well as temporal perspectives. Existing interaction modelings mainly fall into two approaches: observed-historical-information-based interaction and estimated-future-states-based conditional prediction.

History based interaction. Numerous methods attempt to find the interaction by extracting useful information from observed data in history timesteps. LaneGCN [6] models four types of interactions: actor-to-lane, lane-to-lane, lane-to-actor and actor-to-actor. These interactions are captured in series by propagating the spatial information over the lane graph. HiVT [4] finds the interacting entity pairs in each past timestep using observed positions and applies scaled dot-product attention to learn the interactions. However, these interaction modelings based on history features extracted from observed data are not sufficient to purely model the future interaction, which considerably affects the future behaviors of agents. In contrast, our FIMP decouples future feature from history feature to represent potential future information aside from the observed data. By separating agent embedding into future and history, we can model the future interaction as well as history interaction in an end-to-end manner.

Conditional prediction. To consider the future interaction explicitly, some methods [11]–[14], [19] explore the conditional prediction that takes the future states of another agent as an input. CBP [13] takes the ground truth future motion of the query agent and models the behavior changing of a target agent. M2I [12] learns to predict the relationships between agents by classifying them as pairs of influencer and reactor, and produces the reactor’s trajectory conditioned on the estimated influencer’s trajectory. The concurrent work FRM [19] models the future interaction by predicting the lane-level waypoint occupancy explicitly. Then the agents passing the adjacent lanes are regarded as an interacting pair. These approaches can consider potential interactions in the future, but they require access to the ground-truth

motion or should explicitly predict the approximate motion of an influencing agent. Furthermore, the traffic scenarios with mutual interaction are disregarded in most methods. In contrast, our FIMP does not require the prior knowledge of high-level motion cues because implicit future information is used to capture the potential interaction at an intermediate feature-level. We demonstrate that interacting pairs and their connectivity in the future can be well-identified by our future affinity learning and top- k filtering strategy.

III. METHOD

In this section, we first introduce the formulation of multi-head attention in Section III-A and subsequently elaborate on our approach. The architecture of our framework is illustrated in Figure 2.

A. Multi-Head Attention

We exploit multi-head attention (MHA) to model the interaction and temporal dependencies in our method. Following [20], we define an MHA with h heads based on a scaled dot-product attention for input variables X and Y :

$$\text{MHA}(X, Y) = [\text{Attn}_1(X, Y), \dots, \text{Attn}_h(X, Y)]W^O, \quad (1)$$

$$\text{Attn}_i(X, Y) = \text{softmax}\left(\frac{(XW_i^Q)(YW_i^K)^\top}{\sqrt{C/h}}\right)YW_i^V, \quad (2)$$

where projections $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{C \times C/h}$ and $W^O \in \mathbb{R}^{C \times C}$ are parameter matrices, C is the feature dimension and $[\cdot, \cdot]$ indicates concatenation. The multi-head self-attention (MHSA) can be represented as MHA with two identical inputs:

$$\text{MHSA}(X) = \text{MHA}(X, X). \quad (3)$$

B. Input Representation

For input data, we adopt a vectorized representation that involves geometric attributes of entities as a form of vector sets. As the raw vectorized data is not invariant to translation and rotation, we transform the absolute history positions of each agent to be agent-centric where the scene is centered at the current position of a target agent and aligned with its heading. Specifically, we denote the motion history of an agent i as $A_i = \{\Delta p_i^t\}_{t=1}^{T_{\text{history}}}$, where $\Delta p_i^t \in \mathbb{R}^2$ is a 2D motion vector from timestep $t-1$ to t and T_{history} is the number of history timesteps. Then the multi-agent history can be denoted as $A_{\text{input}} = \{A_i\}_{i=1}^N \in \mathbb{R}^{N \times T_{\text{history}} \times 2}$, where N is the number of agents. To capture the interactions, we sample the neighboring agents for each target agent in the current frame $t = T_{\text{history}}$ within a predefined radius. The relative position and heading of sampled agent j in the coordinates centered at target agent i are represented as $\{p_{ij}, \theta_{ij}\}$. We also sample the set of lane segments that are close to the current position of each agent and convert them into corresponding agent-centric coordinates. We represent the lane segment ω surrounding agent i as $\{l_{i\omega}^{\text{start}}, l_{i\omega}^{\text{end}}\}$, where $l_{i\omega}^{\text{start}}$ and $l_{i\omega}^{\text{end}}$ are starting and ending positions of the lane segment in the coordinates centered at agent i .

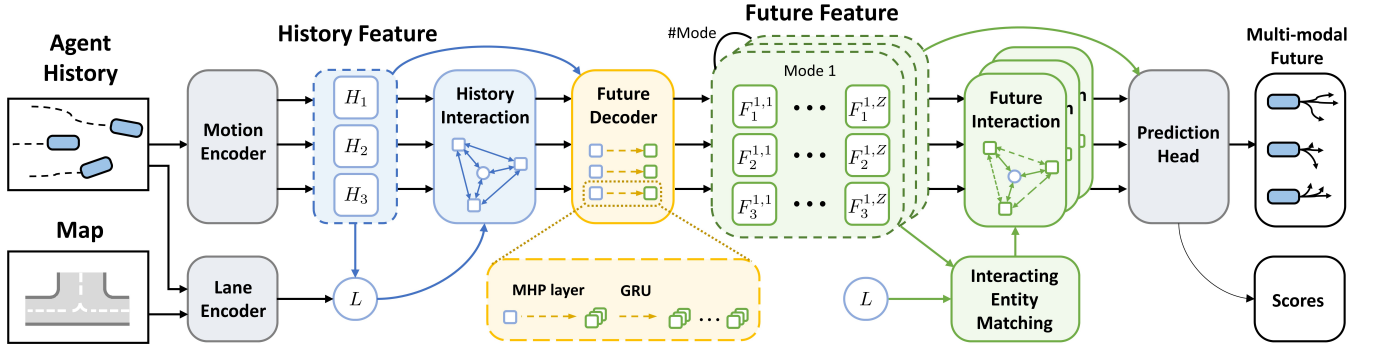


Fig. 2. Architecture of FIMP framework. Our network consists of two parts each for history and future feature learning. The future decoder separates the future feature space from the history, enabling the interaction modeling in respective time zones.

C. History Feature Encoding

Motion encoder. Given the motion history $A_{input} \in \mathbb{R}^{N \times T_{history} \times 2}$, we first encode the motion vector at each timestep by using Multi-Layer Perceptrons (MLPs) to obtain the motion embedding $A_m \in \mathbb{R}^{N \times T_{history} \times C}$:

$$A_m = \text{MLP}(A_{input}). \quad (4)$$

Then, we learn the temporal dependencies across the history sequence of A_m by adopting a temporal attention module. Similar to ViT [21], each layer in the module is comprised of layer norm (LN) operations [22], MHSA block, residual connections [23], and a feed-forward network (MLP). A learnable token with size \mathbb{R}^C is added to the end of input sequence, which reasons the sequence of motions as a whole. Taking A_m as an input for the initial layer, the output A_m^{l+1} from the l^{th} encoder layer is obtained as

$$\hat{A}_m^l = \text{MHSA}(\text{LN}(A_m^l + p)) + (A_m^l + p), \quad (5)$$

$$A_m^{l+1} = \text{MLP}(\text{LN}(\hat{A}_m^l)) + \hat{A}_m^l, \quad (6)$$

where positional embeddings $p \in \mathbb{R}^{(T_{history}+1) \times C}$ is added to the motion embedding. Finally, we adopt the updated learnable token in the last motion embedding as the history feature $H \in \mathbb{R}^{N \times C}$, which represents the motion history information of N agents.

History interaction. We extract the agent-map and agent-agent interactions in the history to better understand the observed scene. For target agent i , we first encode the surrounding lane ω in the agent-centric coordinates by using MLPs to obtain the lane embedding $L_{i\omega} \in \mathbb{R}^C$:

$$L_{i\omega} = \text{MLP}([l_{i\omega}^{start}, l_{i\omega}^{end} - l_{i\omega}^{start}, attr_{\omega}]), \quad (7)$$

where $attr_{\omega}$ is the lane attributes (e.g., turning direction, lane type). The obtained lane embedding $L_{i\omega}$ is then incorporated into target agent i by MHA to obtain lane-aware agent feature H_i^L :

$$H_i^L = \text{MHA}(H_i, \{L_{i\omega} \mid \omega \in N_L(i)\}), \quad (8)$$

where $N_L(i)$ is the set of neighboring lanes respective to agent i . To capture the agent-agent interaction, we first project the feature H_j of neighboring agent j into the coordinate of target agent i by encoding the relation between their agent-centric coordinates:

$$H_j^L = \text{MLP}(H_j^L) + \text{MLP}([p_{ij}, \cos(\theta_{ij}), \sin(\theta_{ij})]). \quad (9)$$

The projected feature H_{ij}^L can be learned by the relative position p_{ij} and heading θ_{ij} . Then we incorporate the neighboring agents' feature by MHA to obtain interaction-aware history feature \tilde{H}_i :

$$\tilde{H}_i = \text{MHA}(H_i^L, \{H_{ij}^L \mid j \in N_A(i)\}), \quad (10)$$

where $N_A(i)$ is the set of neighboring agents respective to agent i .

D. Future Decoder

Our approach adopts an intermediate future decoder that decouples the future feature from history feature. It aims to derive features that represent the potential future information aside from historical information, enabling feature-level future interaction modeling before predicting motions. The decoder comprises two primary components: a multi-head projection (MHP) layer for multi-modal prediction and GRU for temporalization. The MHP layer generates diverse future mode embeddings from the history feature \tilde{H} using different MLPs for each mode. The m^{th} mode embedding $F^m \in \mathbb{R}^{N \times C}$ can be computed by MLP_m as

$$F^m = \text{MLP}_m(\tilde{H}), \quad m \in \{1, 2, \dots, M\}, \quad (11)$$

where M is the number of modes. Then GRU is used to temporalize the mode embedding into several *future time zones*, with each zone containing a predefined number of timesteps. When the number of future timesteps to be predicted and time zones are T and Z respectively, the number of future timesteps in each zone becomes T/Z . We divide the future period into these sparse time zones instead of dense timesteps because it is more effective to map uncertain future information, and the interaction tends to occur over consecutive timesteps. As illustrated in Figure 3, a GRU takes a history feature H obtained from the motion encoder as an initial hidden state h_0 and uses the identical mode embedding F^m as the inputs for all time zones, different from conventional GRU that takes temporal inputs. By repeating non-temporal mode embedding as a input sequence, we can temporalize the embedding into zone-wise future feature $F^{m,z} \in \mathbb{R}^{N \times C}$, where $z \in \{1, 2, \dots, Z\}$ is an index of time zones.

Learning. After interactions are captured within respective zones of future features, another GRU is used to temporalize the zone into timesteps. Similar to GRU in the future decoder,

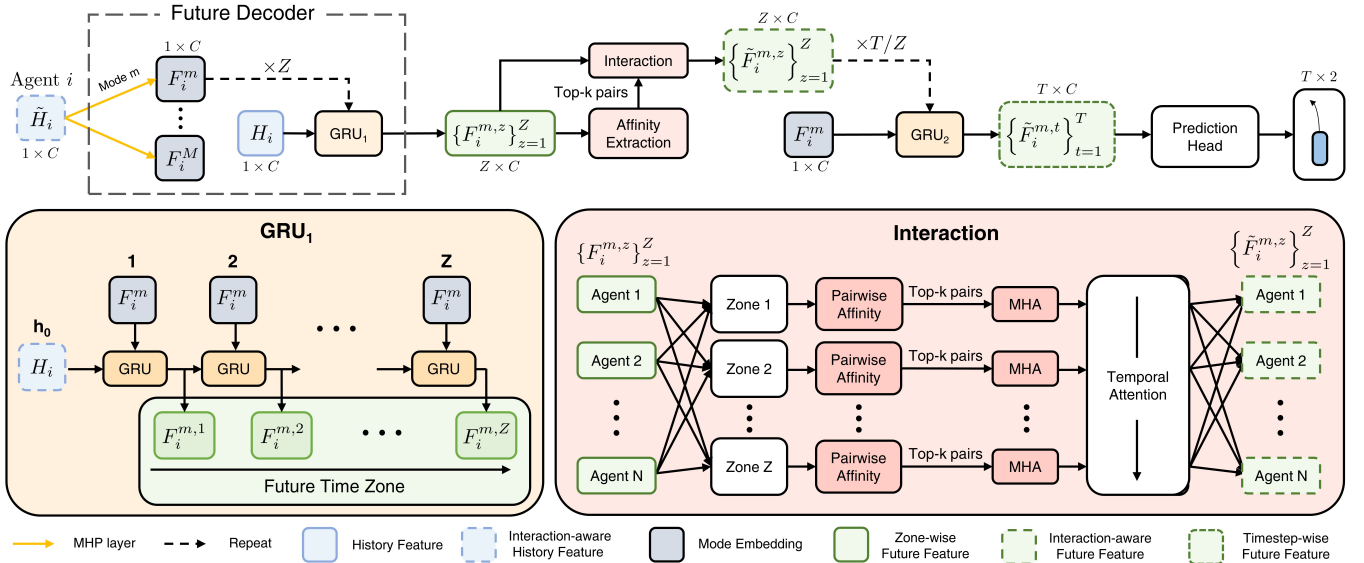


Fig. 3. Network structure from future decoder to prediction head. For brevity, the process to predict the motions on a single mode of agent i is illustrated. GRU_2 works similar to GRU_1 , but interaction-aware zone-wise future feature $\tilde{F}^{m,z}$ is only repeated for timesteps that it involves.

mode embedding is used as the initial hidden state and future features of each zone become the inputs for timesteps that it covers. The obtained timestep-wise future feature $\tilde{F}^{m,t}$ with $t \in \{1, 2, \dots, T\}$ is then used to predict the position at the corresponding timestep. In this learning, future feature of each time zone is optimized to represent the potential states in a particular period over the involved future timesteps.

E. Future Interaction Modeling

In this section, we describe our approach for identifying and modeling potential future interactions among entities.

Agent-lane interaction. As the map topology is time-invariant, agent-lane interaction is less difficult to capture than agent-agent interaction. We thus roughly incorporate the lane information within the potential future trajectories of agents, similar to history interaction modeling. We borrow the lane embedding $l_{i\omega}$ computed in Eq. 7 and apply MHA to future feature F^m as in Eq. 8, but with varied sampling regions to cover the lanes around future motions. The region is defined roughly with a bigger radius considering the heading of an agent, and the attention module learns to selectively incorporate the interacting lanes.

Agent-agent interaction via affinity learning. We identify the interacting agent pairs by learning the affinity between implicit future features. It aims to extract high affinity from agent pairs where message passing is required owing to potential interaction. As the encoded information in $F^{m,z}$ is based on each agent-centric coordinate, it is necessary to transform the future features of all agents into the same feature space before computing affinity. We therefore project them into an autonomous vehicle (AV)'s coordinates as the reference feature space. Similar to Eq. 9 in history interaction modeling, the projected future feature $F_{\alpha i}^{m,z} \in \mathbb{R}^C$ of agent i in the coordinate of AV α can be obtained as

$$F_{\alpha i}^{m,z} = \text{MLP}(F_i^{m,z}) + \text{MLP}([p_{\alpha i}, \cos(\theta_{\alpha i}), \sin(\theta_{\alpha i})]). \quad (12)$$

Here, the affinity matrix between projected future features $F_{\alpha i}^{m,z}$ is computed based on feature distance and used to determine which agent pair to model the future interaction from. For each target agent, only the agents with top- k high affinities are selected as the interacting pairs and can perform the message passing. The interaction-aware future feature $\tilde{F}_i^{m,z}$ is obtained by MHA as follows:

$$\tilde{F}_i^{m,z} = \text{MHA}(F_i^{m,z}, \{F_{ij}^{m,z} \mid j \in \text{Top}_k^{m,z}(i)\}), \quad (13)$$

where $\text{Top}_k^{m,z}(i)$ indicates the indices of interacting agents. In this process, $F_{\alpha i}^{m,z}$ is learned to represent the future states of agents in the reference coordinates so that the relationships between agents' future positions can be indicated by the proximity in reference feature space \mathbb{R}^C . The optimization of affinity learning enables interacting agents to perform message passing with high affinities, leading to better prediction. This top- k filtering strategy is ideal for interaction modeling as only a few agents will be in interaction while others are noises. After learning, we empirically verify in Section IV-C that our affinity learning with top- k filtering enables proper identification of interacting agents in future timesteps, compared to conventional interacting agents matching strategies of existing methods.

As the interaction is considered at each time zone independently, we further capture the temporal information by applying temporal attention module in the same way as described in Eqs. 5 and 6.

F. Multi-Modal Motion Prediction

Multi-modality. As the future behaviors of agents are not deterministic, the network should predict the multi-modal motions to deal with environmental uncertainties. Our FIMP thus generates multiple mode embeddings in the future decoder and extracts timestep-wise future features $\tilde{F}^{m,t}$ for each mode. An MLP in the prediction head takes these features to output the final forecasting as a form of Laplace distribution with location $\mu_i^t \in \mathbb{R}^2$ and scale $b_i^t \in \mathbb{R}^2$,

which represent the state of an agent at future timestep t . Following [7], the displacement error d_i^m at the endpoint is predicted for each mode and is converted to the confidence score with softmax function.

Training objective. We train our model end-to-end with the regression loss \mathcal{L}_{reg} and the classification loss \mathcal{L}_{cls} . The winner-takes-all (WTA) strategy is used in regression tasks to avoid penalizing diverse plausible predictions. The best prediction $(\hat{\mu}_i^t, \hat{b}_i^t)$ among M modes is selected for each agent by calculating the average displacement error along timesteps. Then we use the negative log-likelihood as

$$\mathcal{L}_{reg} = -\frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \log P(g_i^t | \hat{\mu}_i^t, \hat{b}_i^t), \quad (14)$$

where g_i^t is the ground-truth location of agent i at future timestep t and $P(\cdot|\cdot)$ is a probability density function of Laplace. For classification loss \mathcal{L}_{cls} , we adopt smooth L_1 loss between ground-truth displacement and prediction. The final loss is the sum of regression and classification losses with equal weights.

IV. EXPERIMENTS

A. Experimental Setup

Dataset. We evaluate our approach on the large-scale Argoverse motion forecasting dataset [24], which comprises approximately 320K real-world driving scenarios. Each data comprises the trajectories of agents and an HD map. The training and validation set contains 5-second scenarios sampled at 10 Hz, where the first 2-second trajectories are given as an input and remaining 3-seconds are labeled as future trajectories to be predicted. For the test set, only the first 2-second observations are given.

Metrics. As motion prediction is multi-modal by nature, we adopt the widely used metrics for multi-modal evaluation, including minimum Final Displacement Error (minFDE) and minimum Average Displacement Error (minADE). minFDE is the L2 distance between the best predicted trajectory and ground-truth trajectory at the final future timestep, while minADE is the error averaged over all future timesteps. Argoverse benchmark allows up to $M = 6$ predictions for each agent and we predict six trajectories following the baseline.

Implementation details. We train our model in an end-to-end manner by using an AdamW [25] optimizer for 64 epochs with two Titan RTX GPUs. We use a batch size of 32 and an initial learning rate of 0.0005, which decays with a cosine annealing scheduler [26]. The agent-lane and agent-agent interaction modules are comprised of 1 and 3 layers respectively, while the temporal attention module consists of 4 layers. Each MHA block has 8 heads and the feature dimension C is 128. The local radii for neighboring lane and agent sampling used in history interactions are both 50m, and the radius for neighboring lane sampling used in future interaction is 100m.

TABLE I
RESULTS ON THE ARGOVERSE VALIDATION AND TEST SET.

Model	Validation set		Test set	
	minFDE	minADE	minFDE	minADE
LaneGCN [6]	1.08	0.71	1.36	0.87
mmTrans [9]	1.15	0.71	1.34	0.84
TPCN [7]	1.15	0.73	1.24	0.82
DenseTNT [18]	1.05	0.73	1.28	0.88
GOHOME [27]	1.26	-	1.45	0.94
PAGA [28]	1.02	0.69	1.21	0.80
THOMAS [17]	1.22	-	1.44	0.94
AutoBot [29]	1.10	0.73	1.37	0.88
Scene Transformer [10]	-	-	1.23	0.80
LTP [8]	1.07	0.78	1.29	0.83
HiVT [4]	0.96	0.66	1.17	0.77
FRM [19]	0.99	0.68	1.27	0.82
FIMP (ours)	0.92	0.64	1.13	0.76

B. Evaluation

Quantitative results. We compare our FIMP with state-of-the-art methods that have recently been applied on the Argoverse dataset. The results on validation and test sets are reported chronologically in Table I. FIMP achieves the best performance on the validation set by a clear margin in terms of minFDE and minADE. The results on the test set also outperform the related works. For all metrics on both sets, our future interaction modeling improves the performance remarkably, reducing the large proportions of prediction errors. As our future interaction modeling is compatible with other state-of-the-art methods, we believe that the result can be further improved by adopting a stronger baseline.

Among considered studies, the latest concurrent method FRM [19] also aims to capture the future interaction by explicitly predicting the lane-level waypoint occupancy first and subsequently performing conditional prediction, but shows inferior performance. This is because the occupancy along lane axis is predicted without considering possible interactions, and the final forecasting is heavily reliant upon the accuracy of this occupancy estimation. Furthermore, the approach dependent on lane topology cannot be used in the map where lane information is not available or poorly constructed, and 2-phase motion prediction inference is not suitable for real-time applications. In contrast, our FIMP extracts implicit future information before computing motions in an end-to-end manner, which is optimized to determine when and where agents will be in the future. Based on our experiments, it appears that using this intermediate feature-level information to represent potential future states is adequate for identifying interacting agents and their connectivity, without encountering issues with pre-estimation errors.

Qualitative results. We present the qualitative results of FIMP compared to related models in Figure 4. For LaneGCN [6] and HiVT [4], we use the pretrained models provided by authors. We can observe that our FIMP captures the potential interaction in the future and predicts the plausible trajectories for both agents without any conflicts. In contrast, other historical-information-based methods lack capability of capturing the future motions of an interacting agent, resulting in unrealistic trajectory overlaps. We provide more comparison and analysis on diverse traffic scenarios in the supplementary material.

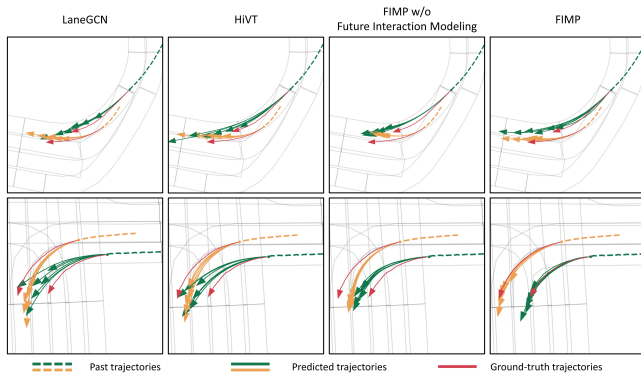


Fig. 4. Qualitative comparison of models in the scenarios where future interaction modeling is essential. The trajectories of interacting agents are shown in green and orange while ground-truth trajectories are in red.

TABLE II
ABLATION STUDY ON INTERACTION MODELING.

	History		Future		minFDE	minADE
	A-L	A-A	A-L	A-A		
1	✓				1.17	0.74
2	✓	✓			0.97	0.67
3	✓			✓	0.93	0.65
4	✓	✓		✓	0.92	0.65
5	✓	✓	✓	✓	0.92	0.64

C. Ablation Study

We conduct an ablation study to demonstrate the effectiveness of each component in FIMP. All ablation results are based on the Argoverse validation set.

Importance of future interaction. We present the contributions of history and future interaction modelings in Table II. The first two rows show that capturing the interactions in agent-lane and agent-agent pairs during the observed timesteps can considerably improve the performance of motion prediction. This finding is consistent with prior research in this field. However, a comparison between rows 2 and 3 reveals that it is much more significant to consider the potential future interactions to predict the agents’ future motions, and our proposed FIMP captures such interactions effectively. It is also observed in row 4 that the agent-agent interaction in the history timesteps becomes less significant when the network is able to model the future interaction. Furthermore, incorporating agent-lane future interaction modeling can enhance the performance of minADE as presented in row 5.

Interacting pair matching strategy. In Table III, we investigate three distinct methods for discerning interacting agents in future timesteps: local-region-based matching, nearest-order-based matching, and high-future-affinity-based matching. The first two methods operate on the assumption that agent pairs satisfying the matching criteria at the current time step will interact in the future, akin to historical interaction modeling. However, the results reveal that these conventional methods struggle to accurately identify interacting pairs in the future, as the relationships between agents in the future do not consistently align with those at the current timestep. In contrast, our future-affinity-based matching converge to better results by identifying the top- k interacting agents in

TABLE III
ABLATION STUDY ON STRATEGIES OF MATCHING INTERACTING AGENTS IN THE FUTURE.

Interacting agent	top- k	minFDE	minADE
Local region ($r=50$)	✗	0.96	0.66
Local region ($r=100$)	✗	0.95	0.66
Nearest order	5	0.96	0.67
Nearest order	10	0.96	0.66
High future affinity	5	0.93	0.65
High future affinity	10	0.92	0.64
High future affinity	20	0.93	0.64

TABLE IV
ABLATION STUDY ON THE NUMBER OF FUTURE TIME ZONES AND INFERENCE LATENCY.

Z	#timesteps	minFDE	minADE	Latency (ms/agent)
3	10	0.94	0.65	17
5	6	0.92	0.64	24
6	5	0.92	0.65	28
10	3	0.94	0.65	37

the future properly. The choice of k is not too sensitive but $k = 10$ leads to best performance.

Number of future time zones Z and latency. As it is ineffective to consider the interaction across all future timesteps, our intermediate future decoder derives information for sparse time zones instead of dense timesteps. We divide the $T = 30$ future timesteps into Z time zones, capturing future interaction within each respective zone. To evaluate the impact of the number of time zones, we conduct an ablation study by varying Z as shown in Table IV. The results indicate that setting $Z = 3$, with 10 timesteps in each zone, leads to worse performance compared to $Z = 5$ or 6 due to an aggregation of too many timesteps into a single zone. This aggregation makes it challenging to extract precise position information for the final prediction head. Conversely, setting $Z = 10$ also results in poor performance, as the interaction is considered too densely along timesteps, allowing for redundant message passing. In addition, we measure the inference latency on the validation set using a batch size of 1 and a Titan RTX GPU. As our approach can make predictions for all agents simultaneously using a single forward, FIMP with $Z = 5$ successfully predicts 6 possible trajectories of an agent at an average speed of 24ms while also capturing potential future interactions with other agents.

V. CONCLUSION

In this paper, we present a novel future interaction modeling for multi-agent motion prediction. Our model FIMP captures the potential interaction in an end-to-end manner by adopting a future decoder that derives the implicit future information aside from observed data. Experiments demonstrate that the interacting agent pairs and their relationships in the future can be effectively identified by learning future affinities and using top- k filtering strategy. FIMP achieves superior performance on the Argoverse motion forecasting dataset with real-time inference, and future work can combine our future interaction modeling with other state-of-the-art methods to further build a strong framework.

REFERENCES

- [1] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.
- [2] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.
- [3] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 500–507.
- [4] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8823–8833.
- [5] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectormet: 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.
- [6] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.
- [7] M. Ye, T. Cao, and Q. Chen, "Tpcn: Temporal point cloud networks for motion forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 318–11 327.
- [8] J. Wang, T. Ye, Z. Gu, and J. Chen, "Ltp: Lane-based trajectory prediction for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 134–17 142.
- [9] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.
- [10] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.
- [11] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.
- [12] Q. Sun, X. Huang, J. Gu, B. C. Williams, and H. Zhao, "M2i: From factored marginal trajectory prediction to interactive prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6543–6552.
- [13] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying driver interactions via conditional behavior prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3473–3479.
- [14] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.
- [15] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.
- [16] J. Hong, B. Sapp, and J. 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*, 2019, pp. 8454–8462.
- [17] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Thomas: Trajectory heatmap output with learned multi-agent sampling," *arXiv preprint arXiv:2110.06607*, 2021.
- [18] J. Gu, C. Sun, and H. Zhao, "Densent: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.
- [19] D. Park, H. Ryu, Y. Yang, J. Cho, J. Kim, and K.-J. Yoon, "Leveraging future relationship reasoning for vehicle trajectory prediction," in *International Conference on Learning Representations*, 2023.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [22] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [23] 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.
- [24] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.
- [25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [26] —, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.
- [27] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Gohome: Graph-oriented heatmap output for future motion estimation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9107–9114.
- [28] F. Da and Y. Zhang, "Path-aware graph attention for hd maps in motion prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6430–6436.
- [29] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal, "Latent variable sequential set transformers for joint multi-agent motion prediction," *arXiv preprint arXiv:2104.00563*, 2021.