RUFI: Reducing Uncertainty in behavior prediction with Future Information

Seokha Moon[1], Sejeong Lee[1], Hyun Woo[1],
Kyuhwan Yeon[2], Hayoung Kim[2], Seong-Gyun Jeong[2], and Jinkyu Kim[1]
[1]Department of Computer Science and Engineering, Korea University
[2]42dot Inc.
Correspondence: jinkyukim@korea.ac.kr

Abstract

Autonomous driving has shown significant progress in recent years, but accurately predicting the movements of surrounding traffic agents remains a challenge for ensuring safety. Previous studies have focused on behavior prediction using large-scale data with diverse information like lane and agent information. However, these studies only use historical information, leading to uncertainty in predicting interactions between agents, which can result in collisions or incorrect trajectory predictions. To address this, we propose a novel method that uses future information during training to reduce uncertainty. Our approach leverages a Teacher-Student technique and attention-based model to reflect agent interaction. To bridge the gap in future information between the student and teacher models, we introduce Lane-guided Attention Module(LAM) that predicts trajectory using only local information in the student model. Our proposed model achieves state-of-the-art performance on the Argoverse motion forecasting dataset, demonstrating that future data, which was previously used only for supervision, can be effectively incorporated into the training process. This study is the first attempt to use future information during training for a behavioral prediction task, and provides a valuable contribution to this field.

1. Introduction

Predicting behavior is a crucial and challenging task for autonomous vehicles as it involves avoiding collisions with pedestrians, cyclists, and other vehicles. The complexity of this task may depend on several critical factors, including the need to interact with multiple agents, uncertainty regarding intentions, and challenges in fusing and utilizing multimodal information. Plus, fusion and utilization of multimodal information (such as map data, agent location, and multiple time steps) also present significant challenges.

Behavior prediction models have been developed based on large-scale data [4, 8, 12] that provide sufficient information for driving, such as map, agent poses over multiple time steps. For instance, vectorized methods have been utilized to incorporate map information in some studies [5, 9], while transformer-based methods have been used to fuse multimodal data [6, 10]. Furthermore, researchers have explored the interaction modeling between agents [11] and employed knowledge distillation techniques for behavior prediction [1, 3]. These models often produce a per-agent parametric distribution over multiple trajectories, where these per-agent trajectories may overlap in space, violating physical space occupancy constraints.

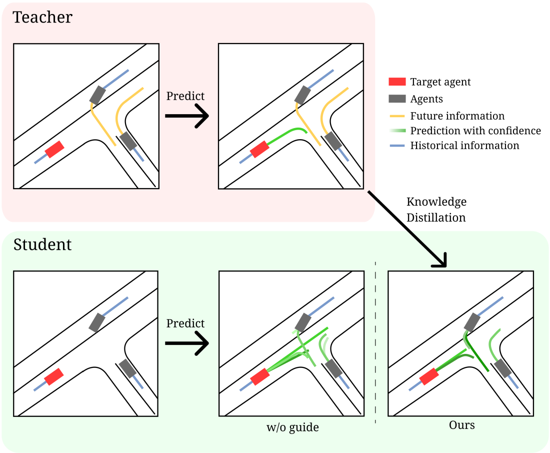

Figure 1. RUFI utilizes future information as direct input data during training to effectively capture interactions between the target agent and other agents. To achieve this, RUFI employs a Teacher model that directly utilizes the future information of other agents to perceive interactions, and uses the interaction information of the Teacher model as a guide to prevent unreasonable trajectories and collisions in the student model.

To optimize multi-agent trajectory uncertainty jointly, we advocate for utilizing a Teacher-Student learning technique where we train a Teacher model that takes agents' future positions as input and learns the interaction information between agents, which is then transferred to the Student model using attention-based knowledge distillation

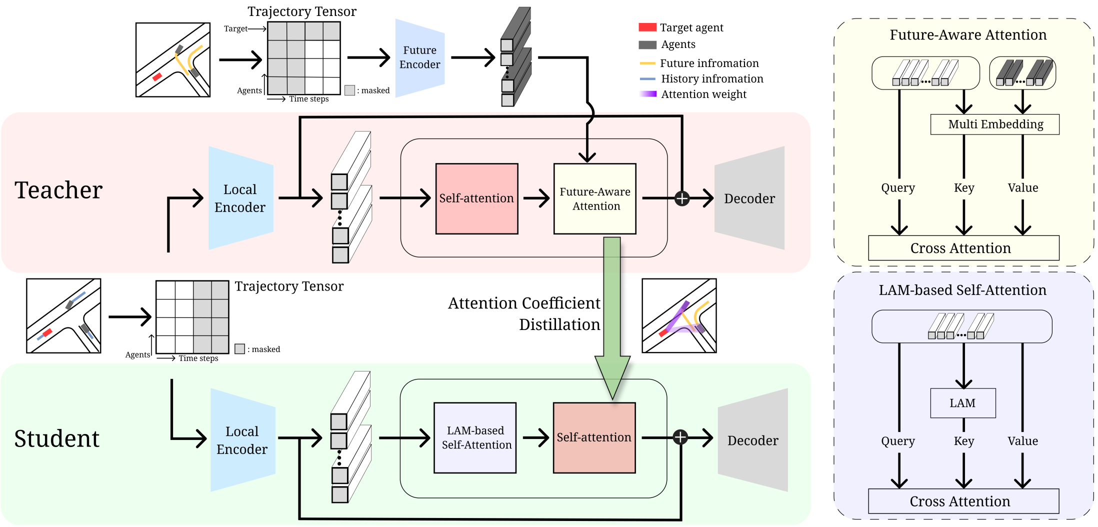

Figure 2. An overview of our proposed method called RUFI. Our model is built upon a student-teacher learning model where (top) the teacher model can leverage the ground-truth future observations of other agents while (bottom) the student model cannot use them. The knowledge is transferred from the teacher model to the student model via attention-based knowledge distillation, i.e., which agents should be *attended* to and be *interacted* with for the final verdict.

techniques [17]. In specific, our proposed model consists of two steps: (i) Teacher model is trained to predict trajectories and accurate interaction between agents using other agents' future positions and local information such as historical information and map data. (ii) Student model predicts future trajectories using local information and the more accurately learned interactions between each agent from the Teacher model as a guide.

We further introduce Lane-guided Attention Module(LAM) to refine each agent's predicted trajectory by extracting relevant lanes and assigning confidence using a confidence calculation equation. The Student model utilizes the refined feature from the LAM to better mimic the interactions between agents learned by the Teacher model, which improves trajectory prediction accuracy.

We demonstrate how to use future information during training and how knowledge distillation techniques using such information can improve the accuracy of behavior prediction models. We introduce Lane-guided Attention Module that effectively utilizes agent features and predicted future information to understand interactions and reduce uncertainty. Our proposed method achieved state-of-the-art results on the Argoverse [8] challenges.

2. Method

As shown in Figure 2, our model utilizes student-teacher learning where (i) the teacher model ($\mathcal{T}$) can leverage other agents' ground-truth future trajectories, i.e., a model as an oracle predicts the target agent's future poses given other agents' past, current, and future poses. This accelerates the learning process by simplifying agent-wise interactions and the multi-modality of output space. Given other agents' future poses, a model $\mathcal{T}$ is trained to attend to other agents for the final verdict selectively. Such attentional knowledge is transferred to (ii) the student model ($\mathcal{S}$) by forcing it to mimic the teacher model's attention distributions.

Teacher Network $\mathcal{T}$. Our teacher network $\mathcal{T}$ first utilizes the standard self-attention module to encode agent-agent interactions, yielding the same dimensional latent representation $\mathbf{s}'_i \in \mathbb{R}^d$. Further, we use another self-attention module to augment the future trajectory Y^i, yielding the agent-wise latent vector $\mathbf{a}'_i \in \mathbb{R}^d$. Specifically, as shown in Figure 2 (b), we use the following query, key, and value with the learnable parameters W'_Q, W'_K, and W'_V:

$$Q'_i = W'_Q \mathbf{s}'_i, \;\; K'_{ij} = W'_K \phi(\mathbf{s}'_j + \mathbf{o}_j), \;\; V'_{ij} = W'_V \phi(\mathbf{s}'_j + \mathbf{o}_j) \tag{1}$$

where ϕ is an MLP layer and $\mathbf{o}_j$ is the future trajectory. Our teacher network uses the future trajectory only to generate attention distributions, preventing the direct use of future information in predicting the target agent's future trajectory.

Lane-guided Attention Module (LAM). As shown in Figure 3, our proposed Lane-guided Attention Module (LAM) first predicts the future trajectory $\bar{Y}^i$ for each agent $i \in [1, n]$ using their respective features as input. Given this, the lane segments around each agent's predicted future paths are extracted, and a corresponding confidence value is assigned to each extracted lane segment using the following equation:

$$C_{i\xi} = \sum_{k=1}^{6} \begin{cases} \Pi^k, & \text{if} \;\; \exists \hat{p}_t^k \,|\, \left|\hat{p}_t^k - p_{\xi,0}\right| < \mathcal{D} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

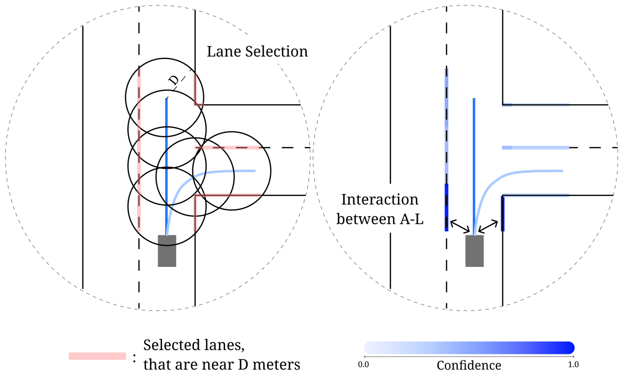

Figure 3. An overview of Lane-guided Attention Module (LAM), which predicts future trajectories of agents using their features and selects lanes within $\mathcal{D}$ meters of the predicted trajectories.

where $\hat{p}_t^k$ represents the position of the i-th agent's k-th mode of the t-th step in the predicted path, $p_{\xi,0}$ represents the starting position of lane ξ, and Π^k represents the confidence of the k-th path in the i-th agent's predicted path. The lanes to be passed by each agent are obtained within $\mathcal{D}$ meter of the predicted trajectory, and each lane reflects the confidence of the trajectory. So, we can extract features for the lane segment related to the i-th agent as follows:

$$F_{i\xi} = \phi_{lane}\left(\left[R_i^T\left(p_{\xi,1}-p_{\xi,0}\right), R_i^T\left(p_{\xi,0}-p_i^{T_c}\right), C_{i\xi}, a_\xi\right]\right), \tag{3}$$

where ϕ_{lane} is the MLP layer. We define a 2x2 rotation matrix for conversion to the central coordinates of the i-th agent as $R_i \in \mathbb{R}^{2\times 2}$. The starting and ending positions of lane segment ξ can be represented as $p_{\xi,0} \in \mathbb{R}^2$ and $p_{\xi,1} \in \mathbb{R}^2$, respectively. The semantic attributes of lane segment ξ are denoted as a_ξ. Then the confidence-weighted lane features are combined with the agent features using cross-attention.

$$Q_i = W_Q F_i,\ K_{i\xi} = W_K F_{i\xi},\ V_{i\xi} = W_V F_{i\xi} \tag{4}$$

Student Network $\mathcal{S}$. Similar to our teacher network, our student network $\mathcal{S}$ utilizes the self-attention module to encode the agent-wise latent vectors $\mathbf{s}_i$ given lane-augmented observation features $\mathbf{l}_i$ for $i \in [1, n]$. To reduce the feature gap between the models' future information, LAM-based Self-Attention is used, which uses the output of LAM as a key instead of the self-attention used by the teacher, as shown in Fig.2. Therefore, the LAM-based Self-Attention of the student model uses agents' features obtained using local information of the target agent as query and value, and uses the predicted future information $\bar{F}_j$ as a key like Eq.(5).

$$\tilde{Q}_i = W_{\tilde{Q}} F_i,\ \tilde{K}_{ij} = W_{\tilde{K}} \bar{F}_j,\ \tilde{V}_{ij} = W_{\tilde{V}} F_j \tag{5}$$

Attention-based Knowledge Distillation. To distill the attention coefficients in the attention module, we employ the knowledge distillation technique used in MINILM [14]. Formally, we use the following distillation loss for attention coefficients in the last interaction layer of the student and teacher models:

$$\mathcal{L}_{\text{distill}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{D}_{KL}\left(\mathcal{A}_i \,||\, \mathbb{A}_i\right) \tag{6}$$

where N is the number of agents. $\mathcal{A}_i$ and $\mathbb{A}_i$ are the attention coefficients of the student model and the teacher model, respectively.

Loss Function. Our model is trained end-to-end by minimizing the following loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{distill}} \tag{7}$$

where $\mathcal{L}_{\text{traj}}$ is the regression loss and $\mathcal{L}_{\text{conf}}$ is the confidence loss for each trajectory. We provide details in the supplemental material.

Table 1. Ablation experiment to see the effect of two modules.

Method	minADE(↓)	minFDE(↓)	MR(↓)
HiVT [15]	0.6868	1.030	0.1024
Ours	**0.6758**	**1.008**	**0.0987**
Ours (w/o LAM)	0.6830	1.013	0.1005
Ours (w/o distillation)	0.6804	1.014	0.0995

Table 2. Performance comparison on Argoverse leaderboard result. †reproduced.

Method	minADE(↓)	minFDE(↓)	MR(↓)
THOMAS [13]	0.9423	1.4388	**0.1038**
DenseTNT [16]	0.8817	1.2815	0.1258
LaneGCN [9]	0.8703	1.3622	0.1620
DSP [7]	0.8194	1.2186	0.1303
HiVT† [15]	0.7995	1.2321	0.1357
Multipath++ [2]	0.7897	1.2144	0.1324
Ours	**0.7867**	**1.2028**	0.1319

3. Experiments

Setup. We use the publicly-available Argoverse dataset [8], which consists of 324,557 scenario samples, each lasting 5 seconds (2 seconds for past observation and 3 for future) and sampled at a rate of 10Hz, for use in training and validation. We also use the standard metrics, including minADE, minFDE, and MR. More details are provided in the supplemental material.

Ablation Study. We first perform ablation studies on the Argoverse validation set using HiVT [15] as the base model. As reported in Table 1, our model achieved performance improvement in all metrics compared to the baseline model.

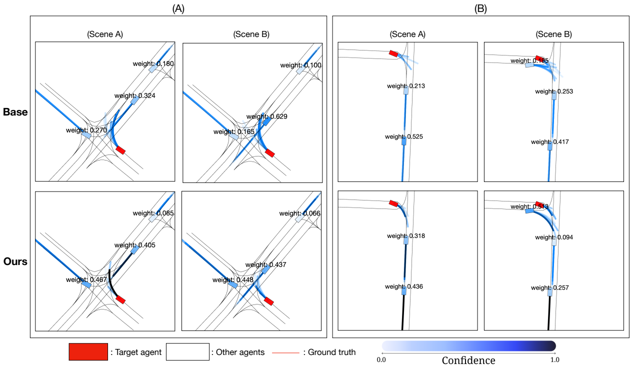

Figure 4. Examples of how our model (and base model [15]) reacts when (A) an agent is moved or (B) removed.

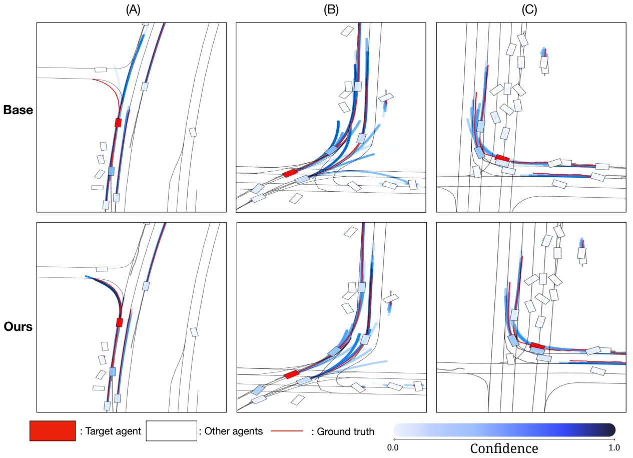

Figure 5. Examples of trajectory prediction outputs for the base model (HiVT [15]) and ours.

Furthermore, our two main modules, LAM and distillation loss, are effective to notably improve the overall performance.

Quantitative Analysis. As shown in Table 2, we compared the performance of our model with other state-of-the-art (SOTA) models on the validation set and test set used in the Argoverse motion forecasting task. Our model clearly outperforms the current SOTA models regarding minADE and minFDE, which may confirm the effect of leveraging attentional knowledge. In the supplemental material, we further provide a more detailed quantitative analysis in terms of different driving scenarios (e.g., go-straight, right-turn, and left-turn).

Qualitative Analysis. As shown in Figure 4, we provide examples where target agent responds to tasks such as agent movement, addition, and deletion. Scenario A shows how trajectory changes when an agent's position changes at the intersection, while Scenarios B demonstrates how well interactions between agents are predicted and how reasonable the trajectory changes are when there is an agent that affects the driving path. Further, in Figure 5, we provide diverse examples of results of trajectory prediction for various situations. Our model is able to more accurately capture direction changes, such as left turns. Further, in scenarios with many agents (where accurate predictions are challenging), our model reasonably predicts trajectories, avoiding collisions.

4. Conclusion

Accurately predicting the future movements of surrounding traffic agents remains challenging for fully-autonomous driving. In this paper, we advocate for incorporating future positional information during training by leveraging a Teacher-Student technique and an attention-based knowledge distillation, further reducing trajectory uncertainty. Also, our proposed Lane-guided Attention Module (LAM) further help incorporate attention-based distilled knowledge. Our model generally outperforms state-of-the-art models on the Argoverse dataset, effectively removing uncertainty and leading to improved interaction predictions.

References

[1] A. Monti et al. How many observations are enough? knowledge distillation for trajectory forecasting. In *CVPR*, pages 6553–6562, 2022. 1

[2] B. Varadarajan et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *ICRA*, pages 7814–7821, 2022. 3

[3] D. A. Su et al. Narrowing the coordinate-frame gap in behavior prediction models: Distillation for efficient and accurate scene-centric motion forecasting. In *ICRA*, pages 653–659. IEEE, 2022. 1

[4] H. Caesar et al. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 1

[5] J. Gao et al. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*, pages 11525–11533, 2020. 1

[6] J. Ngiam et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 1

[7] L. Zhang et al. Trajectory prediction with graph-based dual-scale context fusion. In *IROS*, pages 11374–11381. IEEE, 2022. 3

[8] M. Chang et al. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, pages 8748–8757, 2019. 1, 2, 3

[9] M. Liang et al. Learning lane graph representations for motion forecasting. In *ECCV*, pages 541–556. Springer, 2020. 1, 3

[10] N. Nayakanti et al. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022. 1

[11] Q. Sun et al. M2i: From factored marginal trajectory prediction to interactive prediction. In *CVPR*, pages 6543–6552, 2022. 1

[12] S. Ettinger et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *ICCV*, pages 9710–9719, 2021. 1

[13] T. Gilles et al. Thomas: Trajectory heatmap output with learned multi-agent sampling. *arXiv preprint arXiv:2110.06607*, 2021. 3

[14] W. Wang et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *NeurIPS*, 33:5776–5788, 2020. 3

[15] Z. Zhou et al. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, pages 8823–8833, 2022. 3, 4

[16] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *ICCV*, pages 15303–15312, 2021. 3

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

Supplemental Materials to
RUFI: Reducing Uncertainty in behavior prediction with Future Information

Seokha Moon[1], Sejeong Lee[1], Hyun Woo[1],
Kyuhwan Yeon[2], Hayoung Kim[2], Seong-Gyun Jeong[2], and Jinkyu Kim[1]
[1]Department of Computer Science and Engineering, Korea University
[2]42dot Inc.
Correspondence: jinkyukim@korea.ac.kr

1. Related Work

Trajectory prediction. Recently, extensive research has been introduced for agents' trajectory prediction. However, challenges remain, such as (i) dealing with trajectory uncertainty related to human intentions and (ii) the effective integration of multimodal data. Notable approaches to resolve (i) may include optimizing Gaussian position uncertainty [1, 3, 16] and utilizing CNN (or GNN)-based whole scene (rasterized or vectorized) representation. To resolve (ii), transformer-based architectures [6, 9, 17] have been increasingly chosen as the method for fusing multimodal data. Moreover, as it becomes increasingly important to consider interactions with other agents or map information for more accurate predictions, studies have proposed using vectorized maps. Some use vectorized maps to model the interaction between agents' historical trajectories and road components [5, 8, 10, 14]. Gao *et al.* [5] uses a hierarchical GNN to model these interactions, while Sun *et al.* [10] predicts the interaction type between agents through distance-based rules and leverages this feature to model agents' interaction. In our study, we omit the process of predicting the interaction type and allow the target agent to identify the interaction by itself using the information of other agents' future trajectories in the teacher model.

Knowledge Distillation. Knowledge distillation is a widely used method in various fields of computer vision and natural language processing, aimed at transferring knowledge from high-performing models with a large number of parameters to smaller models with fewer parameters [13, 15]. In recent years, knowledge distillation has been extensively studied and applied in the field of autonomous vehicle application. For example, Su *et al.* [4] proposes a model that is not affected by the number of agents through knowledge transfer from an agent-centric model (teacher) that has high performance but increases computational cost geometrically with the number of agents to a scene-centric model (student). Monti *et al.* [2] proposes an approach that exploits only a few observation inputs to increase predictive performance and eliminate noise probability in the detection phase. In this study, we propose a method of transferring the knowledge of a teacher model, which can better understand the interaction between the target agent and other agents by referencing the future trajectory of other agents, to a student model. This approach allows predicting the interactions between agents based not only on history information but also on distilled features. And this allows for the efficient use of computation resources and improves the performance of the student model.

2. Method Details

Attention-based Knowledge Distillation. The Future-Aware attention, which adds future features, is used in the last interaction layer of the teacher model, and the LAM-based Self-Attention, which enables the student model to capture approximate future information of other agents, is used before the last self-attention module to mimic the interaction of the teacher model. To distill the attention coefficients in the attention module, we employ the knowledge distillation technique used in MINILM [11]. Therefore, we use the following distillation loss for attention coefficients in the last interaction layer of the student and teacher models:

$$\mathcal{L}_{\text{distill}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{D}_{KL}\left(\mathcal{A}_i \,||\, \mathbb{A}_i\right) \tag{1}$$

where N is the number of agents. $\mathcal{A}_i$ and $\mathbb{A}_i$ are the attention coefficients of the student model and the teacher model, respectively.

Loss Function. We use the regression loss $\mathcal{L}_{\text{LAM}}$ for the trajectory used in LAM and the regression loss $\mathcal{L}_{final}$ at the final output using the negative log likelihood function for the Laplace distribution:

$$\mathcal{L}_{LAM}(p_i, \hat{p}_i) = \mathcal{L}_{final}(p_i, \hat{p}_i) = \frac{1}{\sqrt{2b^2}} \sum_{i=1}^{n} |p_i - \hat{p}_i| \, e^{-\frac{|p_i - \hat{p}_i|}{b}} \tag{2}$$

$$\mathcal{L}_{\text{traj}} = \mathcal{L}_{\text{LAM}} + \mathcal{L}_{\text{final}} \tag{3}$$

where p_i denotes the actual future position of the i-th agent, $\hat{p}_i$ denotes the predicted position of the i-th agent, and b is the scale parameter. We also compute the loss $\mathcal{L}_{\text{conf}}$ for the confidence of each trajectory using cross entropy. Finally, we combine the distillation loss, regression loss, and confidence loss to obtain the overall loss function used in our method.

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{distill}} \tag{4}$$

3. Experiments

Dataset. The Argoverse dataset [7] is a valuable resource for training and evaluating behavior prediction models for autonomous vehicles. It includes 3D tracking data and high-definition maps from Pittsburgh and Miami in the United States. The dataset consists of 324,557 scenario samples, each lasting 5 seconds and sampled at a rate of 10Hz, for use in training and validation. The motion forecasting task in the Argoverse dataset involves predicting the future trajectories of agents over a 3-second time horizon, based on their past trajectories over a 2-second time span. The dataset provides 205,942 samples for training, 39,272 samples for validation, and 78,143 samples for testing, making it a comprehensive dataset for behavior prediction research in autonomous vehicles.

Metrics. Our model is evaluated using standard metrics for behavior prediction, which includes minimum Final Displacement Error (minFDE), minimum Average Displacement Error (minADE), and Miss Rate (MR). The minFDE metric measures the final displacement error between the ground truth trajectory's end position and the best predicted end position from K=6 joint samples. The minADE metric, on the other hand, measures the average displacement error between the ground truth trajectory and the best predicted sample out of K=6 joint samples. The MR refers to the percentage of scenarios where the distance between the ground truth trajectory's endpoint and the best predicted endpoint is above diameter threshold.

Additional Quantitative Analysis. To show Quantitative results. We compare the performance of our model with other SOTA models in the validation set and test set used in the Argoverse motion forecasting task. In Table 1, we compared the performance of our model with the base model, HiVT [12], for different maneuvers such as straight, right turn, and left turn, as well as for all agents, including the target agent and other agents that may have missing data. Our model showed an improvement in performance in all situations and for all agents. When compared to the target agent only, we achieved a performance improvement of 1.35%, 2.25%, and 4.34% in the minADE, minFDE, and MR metrics, respectively. When compared to all agents, we achieved a performance improvement of 3.38%, 7.82%, and 7.96% in the same metrics. These results demonstrate that RUFI can extract robust results by leveraging interaction information with other agents even with some information loss for the agent being predicted. We also represent performance of the teacher model in Table 1 to demonstrate the importance of accurately predicting the trajectories of other agents and predicting the interactions between the target agent and other agents when predicting the trajectory of the target agent. When using information about the future trajectories of other agents, excluding the target agent, our proposed teacher model achieved performance improvements of 7.24%, 8.63%, and 9.73% compared to RUFI. Therefore, we can demonstrate the importance of accurate trajectory prediction for other agents and predicting their interactions with the target agent.

method	Maneuver	Target	minADE(↓)	minFDE(↓)	MR(↓)
HiVT [12] (base)	All	Target agent	0.687	1.030	0.1024
	Straight	Target agent	0.615	0.857	0.0691
	Right-turn	Target agent	1.039	1.925	0.2700
	Left-turn	Target agent	1.045	1.860	0.2674
	All	All	1.070	2.071	0.3229
Ours	All	Target agent	**0.676**	**1.008**	**0.0987**
	Straight	Target agent	**0.605**	**0.839**	**0.0669**
	Right-turn	Target agent	**1.024**	**1.881**	**0.2537**
	Left-turn	Target agent	**1.029**	**1.821**	**0.2581**
	All	All	**1.035**	**1.909**	**0.3072**
Teacher	All	Target agent	0.627	0.921	0.0891
	All	All	0.956	1.755	0.2910

Table 1. Performance comparison on **Argoverse validation set**. Ours is the performance in the student model, and the performance of the teacher model is also presented.

Additional Qualitative Analysis. As a qualitative analysis, we present visualized results for various scenarios in the Argoverse validation dataset in Figure 1 and Figure 2. While there were many positive examples, we propose interpretations for some special cases. Firstly, in Figure 1, we examine how the target agent responds to tasks such as agent movement, addition, and deletion. Scenario A shows how trajectory changes when an agent's position changes at the intersection, and Scenarios B and C demonstrate how well interactions between agents are predicted and how reasonable the trajectory changes are when there is an agent that affects the driving path. Secondly, in Figure 2, we show the results of trajectory prediction for various situations. We were able to capture direction changes, such as left turns, more accurately, and in scenarios with many agents where

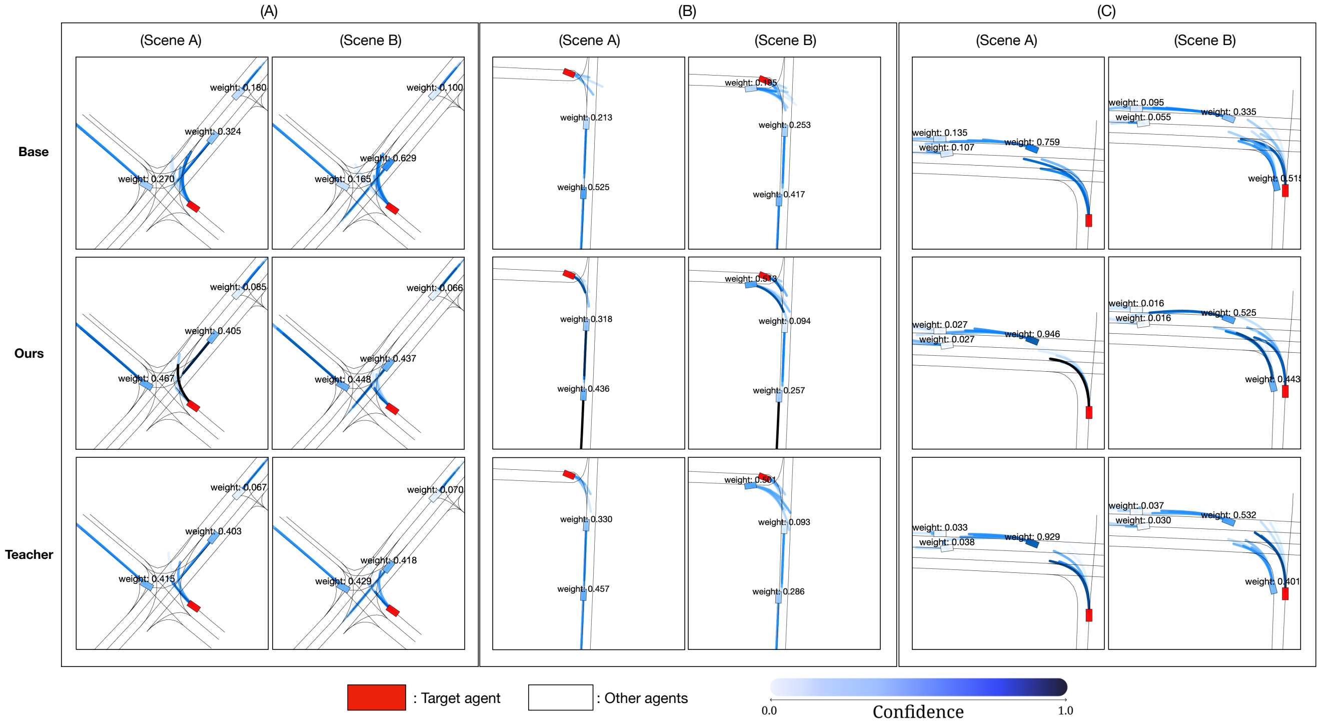

Figure 1. Qualitative results of RUFI on the Argoverse validation dataset. We demonstrate how RUFI reflects interactions by changing the position of other agent in Scenario A and showing how agent trajectories change when agent is removed or added in Scenarios B and C.

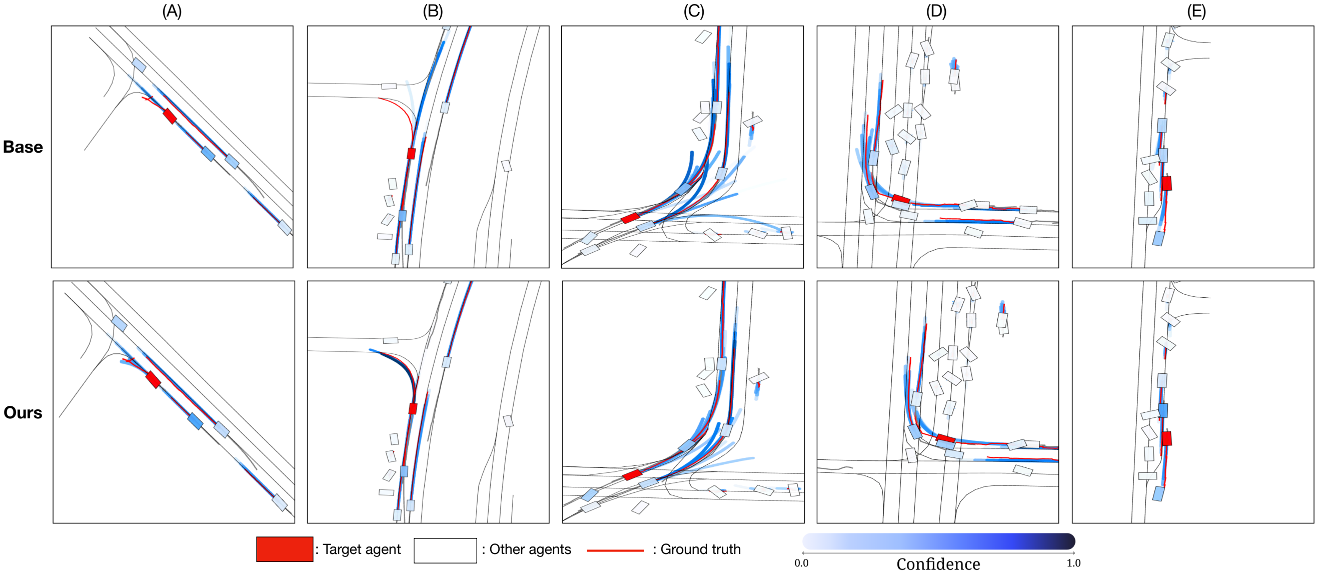

Figure 2. Present the results of RUFI on the Argoverse validation dataset. A and B scenarios demonstrate the model's ability to capture direction changes, while C and D scenarios show reasonable and collision-free trajectory predictions in complex situations with many other agents. E scenario demonstrates that even on straight paths, more accurate predictions are possible through interaction with other agents.

predictions can be difficult, trajectories were predicted more reasonably, avoiding collisions. In addition, by considering interactions with other agents who have similar trajectories even in straightforward sections, we could predict more accurately even in straightforward situations.

References

[1] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 1

[2] A. Monti et al. How many observations are enough? knowledge distillation for trajectory forecasting. In *CVPR*, pages 6553–6562, 2022. 1

[3] B. Varadarajan et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *ICRA*, pages 7814–7821, 2022. 1

[4] D. A. Su et al. Narrowing the coordinate-frame gap in behavior prediction models: Distillation for efficient and accurate scene-centric motion forecasting. In *ICRA*, pages 653–659. IEEE, 2022. 1

[5] J. Gao et al. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*, pages 11525–11533, 2020. 1

[6] J. Ngiam et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 1

[7] M. Chang et al. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, pages 8748–8757, 2019. 2

[8] M. Liang et al. Learning lane graph representations for motion forecasting. In *ECCV*, pages 541–556. Springer, 2020. 1

[9] N. Nayakanti et al. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022. 1

[10] Q. Sun et al. M2i: From factored marginal trajectory prediction to interactive prediction. In *CVPR*, pages 6543–6552, 2022. 1

[11] W. Wang et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *NeurIPS*, 33:5776–5788, 2020. 1

[12] Z. Zhou et al. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, pages 8823–8833, 2022. 2

[13] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. 1

[14] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *ICCV*, pages 15303–15312, 2021. 1

[15] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR, 2019. 1

[16] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco 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*, pages 683–700. Springer, 2020. 1

[17] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr-a: 1st place solution for 2022 waymo open dataset challenge–motion prediction. *arXiv preprint arXiv:2209.10033*, 2022. 1