INTERACTION-AWARE NEIGHBOR REPRESENTATION FOR VEHICLE TRAJECTORY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate trajectory prediction in autonomous driving requires not only expressive model architectures but also informative representations of surrounding vehicles. However, raw kinematic features do not explicitly encode the structure or relative importance of inter-vehicle interactions. We propose a model-agnostic neighbor representation framework that abstracts raw kinematics into discrete longitudinal and lateral interaction states and integrates them into a unified importance score. Slot-semantic weighting and importance-based filtering further refine the representation according to driving context. Evaluated on real-world highway driving datasets across diverse single- and multi-agent prediction models, the proposed representation improves prediction accuracy across most evaluated backbones and metrics. It achieves up to a 28.47% reduction in ADE with negligible forward-pass runtime overhead on an embedded platform.

1 INTRODUCTION

Safe and reliable motion planning in autonomous driving requires accurate prediction of surrounding vehicles' future behavior (Madjid et al., 2026; Messaoud et al., 2021a; Salzmann et al., 2020). In highly interactive scenarios such as highway driving, merging zones, and dense traffic, a vehicle's future trajectory is closely tied to the motion and relative spatial configuration of surrounding vehicles (Madjid et al., 2026; Li et al., 2019; Song et al., 2020; Deo & Trivedi, 2018; Jo et al., 2021). A broad line of work has incorporated neighboring-agent information into trajectory prediction, modeling interactions through social pooling (Deo & Trivedi, 2018; Alahi et al., 2016), graph neural networks (Li et al., 2019; Jo et al., 2021), attention mechanisms (Messaoud et al., 2021a), and transformer-based architectures (Liu et al., 2021b; Ngiam et al., 2022). Beyond advances in prediction backbones and interaction modeling, recent studies highlight the importance of how surrounding agents are encoded as model inputs, showing that raw kinematic inputs and interaction-aware semantic features can yield markedly different interaction cues and prediction performance (Madjid et al., 2026; Messaoud et al., 2021a; Gao et al., 2020; Huang et al., 2025; Hasan et al., 2021).

Many existing methods still represent neighboring agents primarily through raw kinematic features, such as relative position, velocity, and acceleration (Madjid et al., 2026; Liu et al., 2021a). Although these features accurately describe the kinematic state of surrounding vehicles, they do not explicitly encode the semantic content of inter-vehicle interactions. Longitudinal interaction changes gradually as a function of following distance and relative velocity (Treiber et al., 2000; Kesting et al., 2007), while lateral interaction exhibits comparatively sudden state transitions associated with specific events such as lane changes, cut-ins, and merging maneuvers (Meng et al., 2023; Choi & Ahn, 2024). Therefore, leaving both to be inferred from the same raw kinematic features makes it difficult for models to capture their structural differences. Existing interaction-aware representations address this issue only partially, since they are often co-designed with specific backbone architectures and make it difficult to isolate the effect of the representation itself (Zhou et al., 2025; Messaoud et al., 2025; Huang et al., 2022).

This form of representation also differs from the way human drivers naturally interpret the behavior of surrounding vehicles. Rather than perceiving precise numerical values of distance or relative speed, drivers tend to interpret surrounding vehicles qualitatively, such as *approaching rapidly* or *close enough to matter* (Kondoh et al., 2014; KONDOH et al., 2008). In addition, drivers selectively attend to surrounding vehicles perceived as most relevant to the current maneuver, based on driving

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

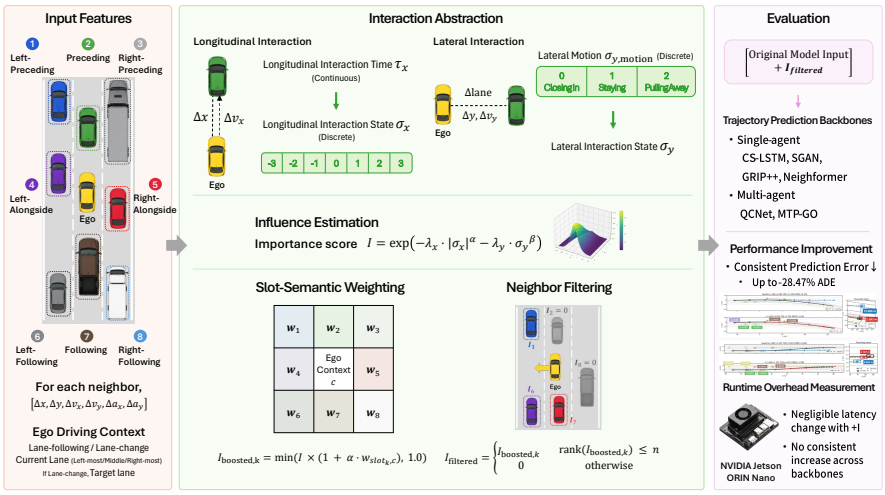


Figure 1: **Overview of the proposed interaction-aware neighbor representation framework.** Raw neighbor kinematics are abstracted into longitudinal and lateral interaction states, integrated into a unified importance score, and refined through slot-semantic weighting and importance-based filtering. The resulting representation can be incorporated into diverse single- and multi-agent trajectory prediction backbones with negligible runtime overhead.

context rather than proximity alone (Stapel et al., 2022; Engström, 2011). However, most existing methods do not explicitly encode either of these aspects at the input level. They typically select neighbors based on simple distance-based criteria and leave the qualitative structure of interactions for the model to learn from data (Huang et al., 2025; Hasan et al., 2021).

To overcome these limitations, we propose a model-agnostic neighbor representation framework for trajectory prediction. It converts raw kinematic inputs into longitudinal and lateral interaction states to compute per-neighbor importance, then applies slot-semantic weighting and importance-based filtering based on the ego vehicle's driving context. The resulting representation provides prediction models with explicit cues on neighbor relevance and driving context that raw kinematic features alone cannot adequately capture.

We evaluate the proposed representation on real-world highway driving datasets across four backbone models. Each backbone is evaluated under two input conditions: raw kinematics only, and raw kinematics augmented with the proposed representation. Experimental results demonstrate consistent improvements across diverse prediction backbones and datasets, while introducing negligible additional inference latency on an embedded platform. The additional inference latency on an embedded platform is also negligible. Our contributions are summarized as follows:

- We propose a model-agnostic framework for representing neighboring vehicle influence, grounded in driver perception.
- We introduce interaction abstractions that convert raw neighbor kinematics into discrete interaction-oriented features.
- We develop a method to estimate neighbor interaction importance, incorporating slot-semantic context, and selectively retain only the most interaction-relevant neighbors.
- We empirically validate that the proposed framework consistently improves trajectory prediction accuracy across diverse model architectures while maintaining practical runtime performance.

2 Related Works

Interaction-aware trajectory prediction. Interaction-aware trajectory prediction methods differ mainly in how they model interactions among agents. Early approaches used social interaction priors such as Social Force (Helbing & Molnar, 1995), while CS-LSTM (Deo & Trivedi, 2018), Social LSTM (Alahi et al., 2016), and Social GAN (Gupta et al., 2018) modeled interactions

through pooled neighbor states. Graph-based methods such as GRIP++ (Li et al., 2020), Social-STGCNN (Mohamed et al., 2020), and STGAT (Huang et al., 2019) represented spatial-temporal interactions among agents using graph structures. Transformer-based approaches have unified the modeling of agent interactions, temporal dependencies, and scene context within a data-driven framework (Ngiam et al., 2022; Yuan et al., 2021; Zhou et al., 2022; Nayakanti et al., 2023; Shi et al., 2024; Zhang et al., 2024; Zhou et al., 2023; Seff et al., 2023; Thakkar et al., 2025). Trajectron++ (Salzmann et al., 2020), MID (Gu et al., 2022), and LED (Mao et al., 2023), along with diffusion-based generative models, focused on multimodal uncertainty in future trajectories. Despite these advances, many methods still use raw kinematic inputs for surrounding agents, such as relative position, velocity, and acceleration (Rudenko et al., 2020). Interaction relevance is often left to pooling, graph, or attention modules inside the backbone (Schmidt et al., 2022). Using Shapley value-based feature attribution, Makansi et al. (2021) showed that many trajectory prediction models rely more on the target agent's past trajectory than on neighbor cues. Saadatnejad et al. (2022) further pointed out that socially-aware models can lack genuine social understanding. These findings motivate input-level neighbor representations that provide explicit interaction cues independently of the prediction backbone.

Human-Centric Neighbor Representation. Psychophysical car-following models have long described driver behavior as a discrete, threshold-driven process, where drivers remain in a behavioral state and transition only when a perceptual threshold or action point is reached (Ahmed et al., 2021; Saifuzzaman & Zheng, 2014). Studies on TTC, time headway, and risk perception further show that longitudinal risk perception cannot be fully captured by a single continuous metric, but varies with proximity and relative velocity (KONDOH et al., 2008; Lu et al., 2012; Tang et al., 2025). Human drivers also selectively attend to surrounding vehicles according to the driving task and scene context (Lemonnier et al., 2020; Liao et al., 2024b). Several works have incorporated these characteristics into trajectory prediction by modeling driver behavioral tendencies (Liao et al., 2024a) or emulating the selective and context-dependent nature of human visual attention (Liao et al., 2024b). However, they commonly embed human perceptual and behavioral characteristics into the backbone's internal mechanisms, rather than transforming the input representation itself into a human-perceptual form. This backbone-dependent design makes it difficult to independently assess the contribution of the representation itself, and limits its applicability across diverse models. Motivated by these limitations, we propose a neighbor representation that explicitly encodes human perceptual principles at the input level, independently of the backbone, making it broadly applicable across diverse trajectory prediction models.

3 Interaction-Aware Neighbor Influence Representation

3.1 Interaction Abstraction

Conventional trajectory prediction methods have relied on raw kinematic features, leaving models to infer interaction structure from numerical inputs. However, these features do not directly capture how drivers perceive and react to surrounding vehicles. Vehicle interactions also exhibit distinct characteristics along the longitudinal and lateral axes, with each axis governed by different relational structures. We therefore abstract raw kinematic observations into axis-specific interaction descriptors, each defined to reflect driver-oriented interaction cues. An example of this abstraction process is shown in Figure 2.

Longitudinal interaction time τ_x. Longitudinal interaction risk cannot be fully characterized by distance or relative velocity alone. For example, two vehicles at the same distance from the ego vehicle can pose very different risks depending on whether they are approaching or receding. This motivates a temporal distance measure that jointly accounts for distance and relative velocity, as in the Time-to-Collision (TTC) metric. We accordingly define Longitudinal Interaction Time τ_x as a lane-independent longitudinal interaction descriptor as follows:

$$\tau_x = \begin{cases} \dfrac{|x_{nb,\mathrm{rear}} - x_{ego,\mathrm{front}}|}{v_{nb} - v_{ego} \pm \epsilon}, & x_{nb} \geq x_{ego} \\ \dfrac{|x_{ego,\mathrm{rear}} - x_{nb,\mathrm{front}}|}{v_{ego} - v_{nb} \pm \epsilon}, & x_{nb} < x_{ego} \end{cases} \tag{1}$$

where $x_{\{ego,nb\},\mathrm{front}}$ and $x_{\{ego,nb\},\mathrm{rear}}$ denote the longitudinal front and rear boundaries of the ego and neighboring vehicles, v_{ego} and v_{nb} their longitudinal velocities, and ϵ a small constant for nu-

merical stability. τ_x represents the estimated time for a neighboring vehicle to reach the ego vehicle's longitudinal position, assuming that the current relative velocity is maintained. By encoding both distance and relative velocity into a single scalar measure, it quantifies longitudinal interaction for neighboring vehicles regardless of lane. A τ_x close to zero indicates a high longitudinal interaction risk, with its sign distinguishing approaching ($\tau_x < 0$) from receding ($\tau_x > 0$) scenarios.

Longitudinal interaction state σ_x. While τ_x serves as a continuous indicator that effectively reflects longitudinal interaction risk, its direct use presents practical limitations. Since it depends on relative velocity in the denominator, small velocity errors can cause significant fluctuations, making τ_x unstable for direct model input. Furthermore, human drivers perceive longitudinal risk through qualitative and relative categories, such as *approaching fast*, *too close*, or *moving away*, rather than as a continuous value. On the basis of these considerations, we define the Longitudinal Interaction State σ_x by discretizing τ_x into seven states using empirical quantile boundaries, where $\sigma_x \in \{-3, -2, -1, 0, 1, 2, 3\}$. Further details of the discretization procedure are provided in Appendix A.1. This symmetric design allows both the sign and relative magnitude of τ_x to be preserved simultaneously, with states closer to zero corresponding to greater longitudinal interaction risk. The discretization retains the time-to-reach semantics of τ_x, while aligning the representation with how human drivers qualitatively perceive longitudinal risk. Consequently, σ_x serves as an interpretable state variable that jointly encodes the direction and intensity of longitudinal interaction.

Lateral interaction state σ_y. Lateral interactions differ from longitudinal ones in that their relevance is largely tied to lane-relative relations and maneuver events such as lane changes, cut-ins, and merging. Rather than tracking continuous lateral displacement, drivers tend to interpret lateral interactions in terms of intent, namely, whether the neighboring vehicle is moving into their lane, remaining in its lane, or pulling away. We accordingly introduce the lateral motion state $\sigma_{y,\text{motion}}$, which encodes the neighbor's lateral behavior as a discrete state. The lateral interaction state σ_y is then derived by integrating $\sigma_{y,\text{motion}}$ with the lane difference between the neighbor and the ego vehicle $\Delta lane$.

The lateral motion state $\sigma_{y,\text{motion}}$ is derived from the normalized lateral position relative to the lane center $\tilde{y} \in [-1, 1]$ and the lateral relative velocity Δv_y. If $|\tilde{y}| \leq 0.5$, the lateral motion state is assigned as staying. For vehicles with $|\tilde{y}| > 0.5$, which are regarded as occupying the boundary region, the state is determined by whether the lateral motion is directed toward or away from the ego lane. We encode the three lateral motion states as integer values $\sigma_{y,\text{motion}} \in \{0, 1, 2\}$, corresponding to closing in, staying, and pulling away, respectively.

The lateral interaction state σ_y is defined by combining the lateral motion state $\sigma_{y,\text{motion}}$ and the lane difference between the neighbor and the ego vehicle $\Delta lane$ as follows

$$\sigma_y = \sqrt{\sigma_{y,\text{motion}}^2 + \Delta lane^2}. \tag{2}$$

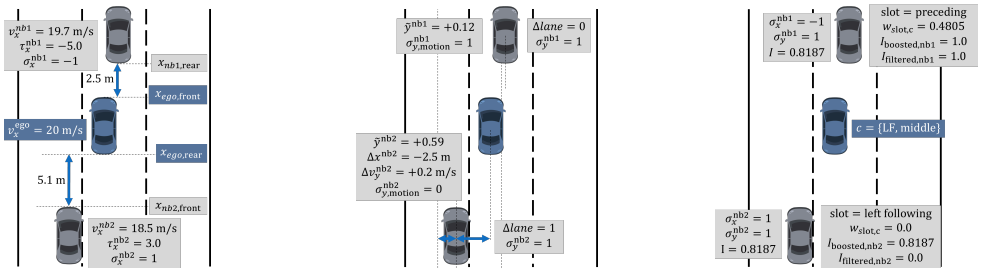
Since $\sigma_{y,\text{motion}}$ is encoded in risk-descending order and $\Delta lane$ approaches zero as the interaction strengthens, a smaller σ_y indicates a more hazardous lateral interaction. This property is subsequently leveraged to quantify each neighbor's influence during interaction importance computation.

3.2 Importance Estimation

While σ_x and σ_y capture longitudinal and lateral risk individually, the overall importance of a neighboring vehicle cannot be determined by either dimension alone. A same-lane vehicle that is longitudinally close may still pose little risk if it is moving away from the ego vehicle in the lateral direction. On the other hand, a vehicle in an adjacent lane can become highly important if it is laterally converging toward the ego lane. We therefore define the interaction importance I as a joint function of both state variables, formulated as follows:

$$I = \exp(-\lambda_x \cdot |\sigma_x|^{\alpha} - \lambda_y \cdot {\sigma_y}^{\beta}) \tag{3}$$

where λ_x and λ_y set the relative weight of longitudinal and lateral interactions, controlling how much each dimension contributes to the overall importance. Since each term penalizes weak interaction along its axis, the importance score remains high only when both dimensions carry significant interaction simultaneously. The exponents α and β control the sensitivity of the importance decay with respect to each interaction state, determining how rapidly the importance decreases as σ_x, σ_y increase. Larger values amplify the contrast between weak and strong interactions, leading



(a) Longitudinal interaction time (b) Lateral interaction features (c) Importance estimation

Figure 2: This figure illustrates the step-by-step construction of the proposed neighbor representation. Figure 2a shows longitudinal interaction abstraction from relative distance and velocity. Figure 2b presents lateral abstraction and importance estimation based on lateral motion state and lane difference. Figure 2c shows slot-semantic weighting and top-1 neighbor filtering in a lane-following context.

to sharper attenuation and enabling the model to emphasize critical interaction regimes. As a result, the proposed function can better capture important interaction scenarios compared to distance-based or single-variable methods. Additional visualizations of Eq. 3 are provided in Appendix A.2.

3.3 Slot-Semantic Weighting

The interaction importance I defined above quantifies the relevance of each neighboring vehicle based on its relative longitudinal and lateral interaction states with respect to the ego vehicle. However, since I is derived from kinematic features, it does not explicitly reflect the semantic role of the slot occupied by each neighboring vehicle. Here, a slot refers to a lane-relative region defined with respect to the ego vehicle, following the lane-based surrounding vehicle representation adopted in prior work (Woo et al., 2019). The local traffic scene is represented as a 3×3 semantic grid aligned to the ego vehicle's lane, where the center cell corresponds to the ego vehicle itself and is excluded from the neighboring slot set. The remaining eight cells form $\mathcal{S} = \{\mathrm{LP}, \mathrm{P}, \mathrm{RP}, \mathrm{LA}, \mathrm{RA}, \mathrm{LF}, \mathrm{F}, \mathrm{RF}\}$, and unoccupied slots are masked during the weight computation. In real-world driving, the influence of a surrounding vehicle depends not only on physical proximity and relative motion, but also on how its slot position relates to the ego vehicle's current maneuver. Thus, two vehicles exhibiting identical kinematic interactions may carry different semantic significance depending on whether they occupy the target lane, the current lane, or a less maneuver-relevant region. To account for this, we derive a slot-semantically adjusted importance score by incorporating a slot-conditioned weight:

$$I_{\text{boosted}} = \min\left(I \times \left(1 + \gamma \cdot w_{\text{slot},\,c}\right),\ 1.0\right), \tag{4}$$

where $w_{\text{slot},c}$ denotes the slot-conditioned importance weight for the neighboring vehicle located in slot $\in \mathcal{S}$ under the driving context $c \in \mathcal{C}$, and γ is a hyperparameter controlling the degree of importance amplification. The context set $\mathcal{C}$ covers Lane Following (LF) and Lane Change (LC) contexts, including lane-position-dependent LF cases and pre-/post-LC cases defined by the ego vehicle's lane-change direction and lane-boundary crossing.

The slot-conditioned weight $w_{\text{slot},c}$ is estimated by a context-specific MLP-based weighting network trained independently of the main prediction model. Since ground-truth slot importance labels are unavailable, an auxiliary trajectory prediction objective is used for training. After training, the validation-set mean of the learned slot weights is used as the fixed $w_{\text{slot},c}$ in the proposed representation. Full training details are provided in Appendix A.4.

3.4 Neighbor Filtering

Human drivers selectively attend to surrounding vehicles relevant to the current maneuver (Stapel et al., 2022), an idea echoed in interaction-aware trajectory prediction models that weight influential neighboring vehicles when forecasting future motion (Messaoud et al., 2021b). Retaining interaction importance scores for all neighbors may therefore inject noise from low-relevance vehicles into the model. We perform neighbor filtering based on I_{boosted} to guide the prediction module toward interaction-relevant neighbors. Figure 3 illustrates how slot-semantic weighting reshapes neighbor importance and how top-n filtering retains the most relevant neighbors. Neighbors are ranked in

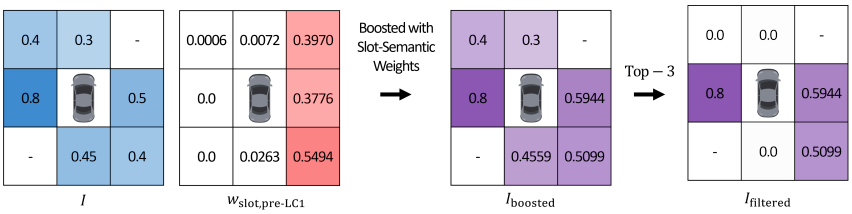


Figure 3: Example of Slot-semantic Weighting (c=pre-LC1) and neighbor filtering (n=3)

descending order of $I_{\text{boosted},k}$, and only the top n neighbors retain their nonzero importance scores:

$$I_{\text{filtered},k} = \begin{cases} I_{\text{boosted},k} & \text{if } \operatorname{rank}(I_{\text{boosted},k}) \leq n \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\operatorname{rank}(I_{\text{boosted},k})$ denotes the rank of neighbor k among all neighbors, and n specifies the maximum number of neighbors with active interaction importance. Since filtering is applied only to the importance scores and not to the agents themselves, the kinematic features of all observed neighbors are preserved as inputs to the prediction model. In this way, I_{filtered} serves as a sparse representation of interaction importance.

4 Experiment

4.1 Experimental Setup

Datasets and Preprocessing. We evaluate the proposed method on the highD (Krajewski et al., 2018) and exiD (Moers et al., 2022) datasets, both collected via overhead drone cameras on real highway segments. highD covers standard highway driving scenarios, while exiD includes more complex road geometries and traffic contexts, such as on-ramps, off-ramps, and merging lanes. Data is downsampled to 3 Hz, and each model observes 2 seconds of history to predict vehicle trajectories over the next 5 seconds.

Input Configuration. Since the backbones employ different input and interaction representations, we use model-specific adapters to incorporate the proposed interaction importance into each model. The baseline and $+I$ conditions follow the same model-specific data and training configurations, with the $+I$ condition additionally incorporating the filtered interaction importance into the corresponding neighbor, actor, or interaction representation. We denote I_{filtered} as I for brevity throughout the experiments. The fixed hyperparameters used to compute the proposed representation are provided in Appendix A.3.

Baseline Models. To verify that the proposed representation is model-agnostic, we evaluate it across both single-agent and multi-agent prediction paradigms. For single-agent baselines, we cover sequential, graph-based, GAN-based generative, and transformer-based models, including CS-LSTM (Deo & Trivedi, 2018), GRIP++ (Li et al., 2020), SGAN (Gupta et al., 2018), and a custom NeighFormer backbone. Furthermore, we extend our evaluation to multi-agent prediction models, incorporating QCNet (Zhou et al., 2023) and MTP-GO (Westny et al., 2023).

Evaluation Metrics. All models within each prediction category (single-agent and multi-agent) are trained and evaluated on identical train/validation/test splits under strictly consistent input configurations. Performance is averaged over five independent runs with different random seeds. We employ ADE, FDE, and RMSE for single-agent models, and minADE@6, minFDE@6, minSADE@6, and minSFDE@6 for multi-agent models.

4.2 Cross-Model Prediction Performance

The cross-model evaluation results for single- and multi-agent predictions on the highD and exiD datasets are summarized in Tables 1 and 2. Overall, incorporating I improves prediction performance across most evaluated models and metrics, supporting the general applicability of the proposed interaction representation across different prediction architectures.

Table 1: Single-agent prediction performance on highD and exiD

Model	Condition	highD			exiD		
		ADE ↓	FDE ↓	RMSE ↓	ADE ↓	FDE ↓	RMSE ↓
CS-LSTM	Baseline	0.1823	0.4666	0.2360	0.1818	0.5039	0.2445
	+I	0.1810 (−0.76%)	0.4649 (−0.37%)	0.2344 (−0.65%)	0.1816 (−0.11%)	0.5037 (−0.04%)	0.2444 (−0.04%)
GRIP++	Baseline	0.4146	1.1388	0.5608	0.6958	1.8700	0.9413
	+I	0.3874 (−6.57%)	1.1049 (−2.97%)	0.5271 (−6.02%)	0.4977 (−28.47%)	1.2935 (−30.83%)	0.6915 (−26.53%)
SGAN	Baseline	0.8168	2.0031	1.0327	0.8021	2.0084	1.0256
	+I	0.7640 (−6.46%)	1.8956 (−5.37%)	0.9721 (−5.87%)	0.7997 (−0.31%)	2.0032 (−0.26%)	1.0226 (−0.30%)
Neighformer	Baseline	0.2195	0.7627	0.3282	0.2329	0.6028	0.2993
	+I	0.2186 (−0.41%)	0.7591 (−0.48%)	0.3267 (−0.47%)	0.1978 (−15.10%)	0.5492 (−8.89%)	0.2621 (−12.42%)

Table 2: Multi-agent prediction performance on highD and exiD

Model	Condition	highD				exiD			
		minADE@6 ↓	minFDE@6 ↓	minSADE@6 ↓	minSFDE@6 ↓	minADE@6 ↓	minFDE@6 ↓	minSADE@6 ↓	minSFDE@6 ↓
QCNet	Baseline	0.1582	0.3961	0.3273	0.9907	0.1872	0.5070	0.3440	1.0366
	+I	0.1552 (−1.88%)	0.3796 (−4.16%)	0.3090 (−5.60%)	0.9176 (−7.37%)	0.1823 (−2.63%)	0.4909 (−3.17%)	0.3374 (−1.92%)	1.0145 (−2.13%)
MTP-GO	Baseline	0.1666	0.4377	0.2601	0.8405	0.2718	0.6294	0.3669	0.9427
	+I	0.1639 (−1.63%)	0.4376 (−0.04%)	0.2487 (−4.37%)	0.8113 (−3.48%)	0.2738 (+0.74%)	0.6348 (+0.86%)	0.3608 (−1.66%)	0.9277 (−1.59%)

For single-agent prediction, GRIP++ reduced ADE by 6.57% on highD and 28.47% on exiD, while Neighformer achieved a 15.10% ADE reduction on exiD. Notably, these larger gains are observed in architectures that explicitly model inter-agent relations through graph-based interaction modeling or attention-based feature fusion. This suggests that models with stronger interaction modeling capabilities may be better able to exploit the additional neighbor-importance information provided by I, particularly in more complex merging and ramp scenarios.

The multi-agent results show a similar trend. QCNet improved all evaluation metrics on both datasets, while MTP-GO consistently reduced the scene-level minSADE and minSFDE metrics. These results indicate that the proposed representation is not limited to single-agent prediction and can also provide complementary interaction information for joint multi-agent forecasting.

Case Study in an Intersection Scenario. We further conduct a case study on an intersection scenario from the inD dataset (Bock et al., 2020) to examine the applicability of the proposed representation beyond highway environments. As shown in Fig. 5, Neighformer with the proposed representation produces a trajectory that closely follows the ground truth, suggesting its applicability to intersection scenarios.

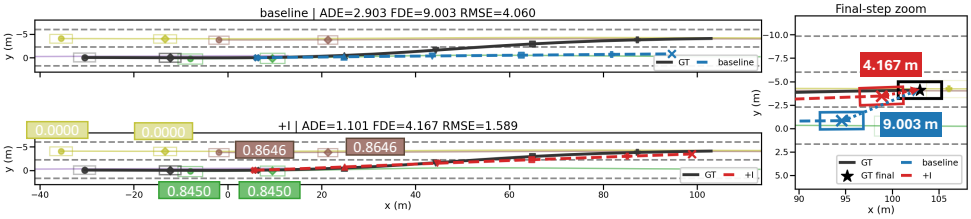
4.3 Ablation Study

We conduct ablation studies using the Neighformer backbone to examine the contribution of individual components in the proposed neighbor representation. The experiments are designed to answer two questions: (1) how the longitudinal and lateral interaction abstractions contribute to trajectory prediction, and (2) whether slot-semantic weighting and neighbor filtering further improve the use of the interaction importance score. Each configuration is trained with three different random seeds.

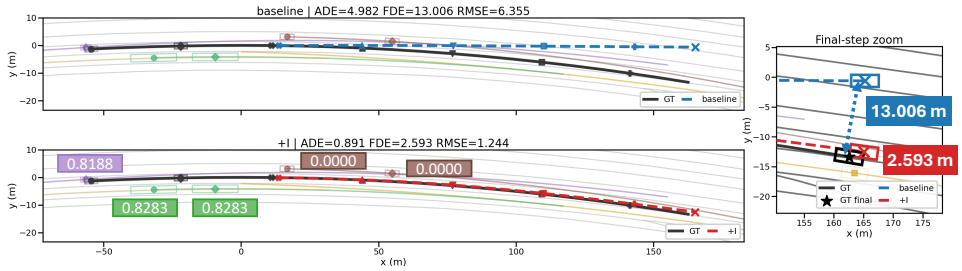
Interaction Representation. We first evaluate the contribution of the proposed longitudinal and lateral interaction representations. Starting from the baseline, we individually add the longitudinal interaction state σ_x, the lateral interaction state σ_y, their direct combination, and the proposed joint importance score I.

As shown in Table 3, both interaction states provide useful information when introduced individually. Adding σ_x reduces ADE from 0.2848 to 0.2670, corresponding to a 6.2% improvement, while σ_y achieves a larger 8.6% reduction to 0.2603. Similar trends are observed for FDE and RMSE, indicating that both longitudinal and lateral interaction abstractions capture complementary cues that are not fully represented by raw kinematics alone.

However, directly concatenating σ_x and σ_y does not lead to further improvement. Instead, ADE increases to 0.3022, which is 6.1% worse than the kinematics-only baseline. This suggests that



(a) highD



(b) exiD

Figure 4: **Qualitative trajectory prediction results of GRIP++.** The baseline (top) and the proposed representation (bottom) are compared. Black lines indicate ground truth. Blue and red dashed lines indicate predictions.

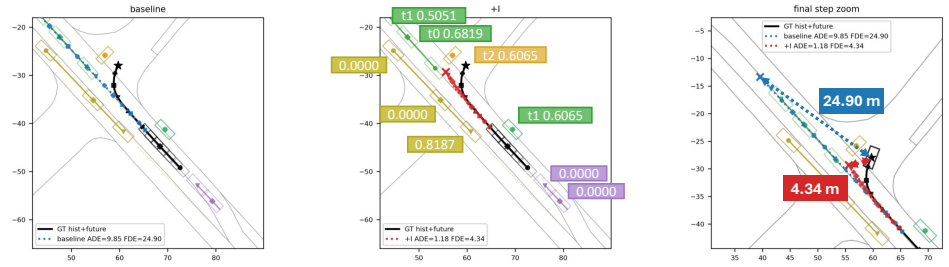


Figure 5: **Trajectory prediction results of Neighformer on the inD dataset.** A case study in an intersection scenario illustrates the applicability of the proposed representation beyond highway driving environments.

simply exposing the model to both interaction states does not guarantee effective fusion and may introduce redundant or conflicting interaction cues.

In contrast, the proposed importance score I, which integrates the longitudinal and lateral interaction states into a unified interaction measure, achieves the best performance across all metrics. It reduces ADE to 0.2394, yielding a 15.9% improvement over the baseline, and achieves the lowest FDE and RMSE of 0.5878 and 0.3018, respectively. These results indicate that the benefit of the proposed representation does not arise merely from adding additional interaction features. Rather, explicitly structuring their joint contribution into a single importance representation is critical for effective neighbor modeling.

Importance Refinement. We further examine the contribution of the two refinement components applied to the interaction importance I: slot-semantic weighting (Slot) and neighbor filtering (NF). We evaluate all four combinations by independently enabling or disabling each component, including I, I+Slot, I+NF, and I+Slot+NF. Applying both mechanisms jointly achieves the best performance, reducing ADE from 0.2394 to 0.2138 and RMSE from 0.3018 to 0.2817. These results suggest that slot-semantic weighting and neighbor filtering work effectively together in improving the use of interaction importance.

4.4 Runtime Efficiency

Beyond prediction accuracy, inference latency is important for deploying trajectory prediction models in autonomous driving and embedded environments. We therefore measured inference latency on an NVIDIA Jetson Orin Nano using single-sample inference (batch size 1) with automatic mixed

Table 3: Ablation study of the proposed interaction representation using Neighformer.

Interaction Representation				Importance Refinement			
Configuration	ADE ↓	FDE ↓	RMSE ↓	Configuration	ADE ↓	FDE ↓	RMSE ↓
Baseline	0.2848	0.6755	0.3541	I	0.2394	0.5878	0.3018
$+\sigma_x$	0.2670	0.6619	0.3380	I + Slot	0.2496	0.6298	0.3187
$+\sigma_y$	0.2603	0.6401	0.3285	I + NF	0.2766	0.6789	0.3493
$+\sigma_x, \sigma_y$	0.3022	0.7015	0.3723	I + Slot + NF	**0.2138**	**0.5839**	**0.2817**
$+I$	**0.2394**	**0.5878**	**0.3018**				

Table 4: Inference time comparison on NVIDIA Jetson Orin Nano

Model	Baseline (ms)			$+I$ (ms)		
	Avg.	Min	Max	Avg.	Min	Max
CS-LSTM	4.55	3.64	8.42	4.48	3.60	7.69
GRIP++	56.58	43.66	375.85	50.31	43.26	230.14
SGAN	10.96	10.47	16.53	10.91	10.52	15.19
Neighformer	28.37	25.16	137.38	29.67	23.08	208.72
QCNet	359.41	324.49	861.06	363.59	329.42	869.17
MTP-GO	334.76	310.00	553.09	335.69	312.95	534.55

precision (bfloat16). Single-agent models were evaluated after 1,000 warm-up iterations over 10,000 measurement iterations, while multi-agent models used 100 warm-up and 1,000 measurement iterations.

Table 4 compares the inference latency of the baseline and the proposed representation with I. Since the proposed method only extends the input representation without modifying the backbone architecture, it introduces little additional computation. Across the evaluated models, adding I resulted in only minor latency variations, with no consistent increase in inference time. These results indicate that the proposed representation provides improved interaction information with negligible forward-pass runtime overhead.

5 Conclusion

We proposed a neighbor-influence representation that models surrounding vehicles from an interaction-aware perspective. Using longitudinal and lateral interaction states, the method represents each neighbor's interaction relevance and movement intent in a discrete and interpretable manner. In addition, it combines interaction importance, slot-semantic weighting, and neighbor filtering to selectively attend to neighbors that matter most in the current scene. Experiments on highD and exiD demonstrate improved prediction performance across diverse single- and multi-agent backbones with negligible forward-pass runtime overhead, indicating that explicitly representing neighbor influence provides complementary interaction information beyond raw kinematics without redesigning the prediction backbone.

However, the current formulation uses predefined interaction discretization and fixed importance parameters to maintain a simple and interpretable representation. Future work will explore data-adaptive interaction modeling while retaining these properties, and extend the evaluation to broader road geometries, heterogeneous traffic participants, and downstream planning tasks.

AI Use Disclosure

In this work, we did not use generative AI tools for any of the tasks requiring mandatory disclosure. We used generative AI tools to edit the manuscript for readability and language clarity, and to assist with formatting. All AI-assisted content was reviewed and verified by the authors. We take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REFERENCES

Hafiz Usman Ahmed, Ying Huang, and Pan Lu. A review of car-following models and modeling tools for human and autonomous-ready driving behaviors in micro-simulation. *Smart Cities*, 4 (1):314–335, 2021. ISSN 2624-6511. doi: 10.3390/smartcities4010019. URL https://www.mdpi.com/2624-6511/4/1/19.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1929–1934, 2020. doi: 10.1109/IV47402.2020.9304839.

Wansik Choi and Changsun Ahn. Data-based vehicle trajectory prediction model for lane-change maneuver. *Int. J. Control Autom. Syst.*, 22(5):1654–1665, May 2024.

Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1468–1476, 2018.

Johan Engström. *Understanding attention selection in driving: From limited capacity to adaptive behaviour*. Chalmers Tekniska Hogskola (Sweden), 2011.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11522–11530, 2020. doi: 10.1109/CVPR42600.2020.01154.

Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17092–17101, 2022. doi: 10.1109/CVPR52688.2022.01660.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number CONF, 2018.

Mohamed Hasan, Albert Solernou, Evangelos Paschalidis, He Wang, Gustav Markkula, and Richard Romano. Maneuver-aware pooling for vehicle trajectory prediction. *arXiv preprint arXiv:2104.14079*, 2021.

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

Shiji Huang, Lei Ye, Min Chen, Wenhai Luo, Dihong Wang, Chenqi Xu, and Deyuan Liang. Interpretable interaction modeling for trajectory prediction via agent selection and physical coefficient. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15422–15429, 2025. doi: 10.1109/IROS60139.2025.11246119.

Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 7(3):652–674, 2022. doi: 10.1109/TIV.2022.3167103.

Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6271–6280, 2019. doi: 10.1109/ICCV.2019.00637.

Eunsan Jo, Myoungho Sunwoo, and Minchul Lee. Vehicle trajectory prediction using hierarchical graph neural network for considering interaction among multimodal maneuvers. *Sensors*, 21 (16), 2021. ISSN 1424-8220. doi: 10.3390/s21165354. URL https://www.mdpi.com/1424-8220/21/16/5354.

Arne Kesting, Martin Treiber, and Dirk Helbing. General lane-changing model mobil for car-following models. *Transportation Research Record*, 1999(1):86–94, 2007.

Takayuki KONDOH, Tomohiro YAMAMURA, Satoshi KITAZAKI, Nobuyuki KUGE, and Erwin Roeland BOER. Identification of visual cues and quantification of drivers' perception of proximity risk to the lead vehicle in car-following situations. *Journal of Mechanical Systems for Transportation and Logistics*, 1(2):170–180, 2008. doi: 10.1299/jmtl.1.170.

Takayuki Kondoh, Nobuhiro Furuyama, Toshiya Hirose, and Toichi Sawada. Direct evidence of the inverse of ttc hypothesis for driver's perception in car-closing situations. *International Journal of Automotive Engineering*, 5(4):121–128, 2014. doi: 10.20485/jsaeijae.5.4_121.

Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125, 2018. doi: 10.1109/ITSC.2018.8569552.

Sophie Lemonnier, Lara Désiré, Roland Brémond, and Thierry Baccino. Drivers' visual attention: A field study at intersections. *Transportation Research Part F: Traffic Psychology and Behaviour*, 69:206–221, 2020. ISSN 1369-8478. doi: https://doi.org/10.1016/j.trf.2020.01.012. URL https://www.sciencedirect.com/science/article/pii/S1369847819301597.

Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3960–3966, 2019. doi: 10.1109/ITSC.2019.8917228.

Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving. *arXiv preprint arXiv:1907.07792*, 2020.

Haicheng Liao, Zhenning Li, Huanming Shen, Wenxuan Zeng, Dongping Liao, Guofa Li, Shengbo Eben Li, and Chengzhong Xu. Bat: behavior-aware human-like trajectory prediction for autonomous driving. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024a. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i9.28900. URL https://doi.org/10.1609/aaai.v38i9.28900.

Haicheng Liao, Shangqian Liu, Yongkang Li, Zhenning Li, Chengyue Wang, Yunjian Li, Shengbo Eben Li, and Chengzhong Xu. Human observation-inspired trajectory prediction for autonomous driving in mixed-autonomy traffic environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14212–14219, 2024b. doi: 10.1109/ICRA57147.2024.10611104.

Jianbang Liu, Xinyu Mao, Yuqi Fang, Delong Zhu, and Max Q.-H. Meng. A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 978–985, 2021a. doi: 10.1109/ROBIO54168.2021.9739407.

Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. *Computer Vision and Pattern Recognition*, 2021b.

Guangquan Lu, Bo Cheng, Qingfeng Lin, and Yunpeng Wang. Quantitative indicator of homeostatic risk perception in car following. *Safety Science*, 50(9):1898–1905, 2012. ISSN 0925-7535. doi: https://doi.org/10.1016/j.ssci.2012.05.007. URL https://www.sciencedirect.com/science/article/pii/S0925753512001270.

Nadya Abdel Madjid, Abdulrahman Ahmad, Murad Mebrahtu, Yousef Babaa, Abdelmoamen Nasser, Sumbal Malik, Bilal Hassan, Naoufel Werghi, Jorge Dias, and Majid Khonji. Trajectory prediction for autonomous driving: Progress, limitations, and future directions. *Information Fusion*, 126:103588, 2026. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2025.103588. URL https://www.sciencedirect.com/science/article/pii/S1566253525006608.

Osama Makansi, Julius Von Kügelgen, Francesco Locatello, Peter Gehler, Dominik Janzing, Thomas Brox, and Bernhard Schölkopf. You mostly walk alone: Analyzing feature attribution in trajectory prediction. *arXiv preprint arXiv:2110.05304*, 2021.

Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5517–5526, 2023.

Xianwei Meng, Jinjun Tang, Fang Yang, and Zhe Wang. Lane-changing trajectory prediction based on multi-task learning. *Transportation Safety and Environment*, 5(4):tdac073, 09 2023. ISSN 2631-4428. doi: 10.1093/tse/tdac073. URL https://doi.org/10.1093/tse/tdac073.

Kaouther Messaoud, Nachiket Deo, Mohan M. Trivedi, and Fawzi Nashashibi. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 165–170, 2021a. doi: 10.1109/IV48863.2021.9576054.

Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, and Fawzi Nashashibi. Attention based vehicle trajectory prediction. *IEEE Transactions on Intelligent Vehicles*, 6(1):175–185, 2021b. doi: 10.1109/TIV.2020.2991952.

Kaouther Messaoud, Matthieu Cord, and Alexandre Alahi. Towards generalizable trajectory prediction using dual-level representation learning and adaptive prompting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27564–27574, 2025. doi: 10.1109/CVPR52734.2025.02567.

Tobias Moers, Lennart Vater, Robert Krajewski, Julian Bock, Adrian Zlocki, and Lutz Eckstein. The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 958–964, 2022. doi: 10.1109/IV51971.2022.9827305.

Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14424–14432, 2020.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987, 2023. doi: 10.1109/ICRA48891.2023.10160609.

Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David J Weiss, Benjamin Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Wm3EA5OlHsG.

Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: a survey. *Int. J. Rob. Res.*, 39(8):895–935, 2020.

Saeed Saadatnejad, Mohammadhossein Bahari, Pedram Khorsandi, Mohammad Saneian, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Are socially-aware trajectory prediction models really socially-aware? *Transportation Research Part C: Emerging Technologies*, 141:103705, 2022. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2022.103705. URL https://www.sciencedirect.com/science/article/pii/S0968090X22001437.

Mohammad Saifuzzaman and Zuduo Zheng. Incorporating human-factors in car-following models: A review of recent developments and research needs. *Transportation Research Part C: Emerging Technologies*, 48:379–403, 2014. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2014.09.008. URL https://www.sciencedirect.com/science/article/pii/S0968090X14002551.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 683–700, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58523-5.

Julian Schmidt, Julian Jordan, Franz Gritschneder, and Klaus Dietmayer. Crat-pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7799–7805, 2022. doi: 10.1109/ICRA46639.2022.9811637.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8545–8556, 2023. doi: 10.1109/ICCV51070.2023.00788.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3955–3971, 2024. doi: 10.1109/TPAMI.2024.3352811.

Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *The European Conference on Computer Vision (ECCV)*, August 2020.

Jork Stapel, Mounir El Hassnaoui, and Riender Happee. Measuring driver perception: Combining eye-tracking and automated road scene perception. *Hum. Factors*, 64(4):714–731, June 2022.

Renjing Tang, Guangquan Lu, Miaomiao Liu, Mingyue Zhu, and Pengrui Li. A unified driving behavior model based on psychological safety space. *Transportation Research Part F: Traffic Psychology and Behaviour*, 109:439–457, 2025. ISSN 1369-8478. doi: https://doi.org/10.1016/j.trf.2024.12.024. URL https://www.sciencedirect.com/science/article/pii/S1369847824003656.

Neerja Thakkar, Tara Sadjadpour, Jathushan Rajasegeran, Shiry Ginosar, and Jitendra Malik. Poly-autoregressive prediction for modeling interactions. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12402–12412, 2025. doi: 10.1109/CVPR52734.2025.01157.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

Theodor Westny, Joel Oskarsson, Björn Olofsson, and Erik Frisk. Mtp-go: Graph-based probabilistic multi-agent trajectory prediction with neural odes. *IEEE Transactions on Intelligent Vehicles*, 8(9):4223–4236, 2023. doi: 10.1109/TIV.2023.3282308.

Hanwool Woo, Hirokazu Madokoro, Kazuhito Sato, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama. Advanced adaptive cruise control based on operation characteristic estimation and trajectory prediction. *Applied Sciences*, 9(22), 2019. ISSN 2076-3417. doi: 10.3390/app9224875. URL https://www.mdpi.com/2076-3417/9/22/4875.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Lu Zhang, Peiliang Li, Sikang Liu, and Shaojie Shen. Simpl: A simple and efficient multi-agent motion prediction baseline for autonomous driving. *IEEE Robotics and Automation Letters*, 9(4):3767–3774, 2024. doi: 10.1109/LRA.2024.3370039.

Yang Zhou, Hao Shao, Letian Wang, Steven L. Waslander, Hongsheng Li, and Yu Liu. Smartpretrain: Model-agnostic and dataset-agnostic representation learning for motion prediction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Bmzv2Gch9v.

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 (CVPR)*, 2022.

Zikang Zhou, Jianping Wang, Yung–Hui Li, and Yu–Kai Huang. Query-centric trajectory prediction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17863–17873, 2023. doi: 10.1109/CVPR52729.2023.01713.

APPENDIX

A ADDITIONAL DETAILS OF THE PROPOSED REPRESENTATION

A.1 INTERACTION ABSTRACTION DETAILS

Longitudinal interaction state discretization. Table 5 summarizes the discretization boundaries for the longitudinal interaction state σ_x under different numbers of bins. The boundaries were obtained from the empirical distribution of valid longitudinal interaction time samples in the training data. τ_x values were collected from all observed neighboring vehicles whose neighbor mask was active, while padded or unoccupied neighbor slots were excluded. For an M-bin setting, the internal boundaries were computed at percentile levels $\frac{100}{M}, \frac{200}{M}, \ldots, \frac{100(M-1)}{M}$. This produces equal-frequency intervals over the empirical distribution of τ_x. The outer boundaries were set to $-\infty$ and ∞ to cover all possible values during inference.

The 7-bin setting ($\sigma_x \in \{-3, \ldots, 3\}$) is used in all experiments. In this setting, the resulting intervals are mapped in order to $\sigma_x \in \{-3, -2, -1, 0, 1, 2, 3\}$. This assignment preserves the sign of τ_x, where negative and positive states correspond to approaching and receding interactions, respectively. States closer to zero represent smaller absolute interaction time and therefore stronger longitudinal interaction.

Table 5: Discretization settings for σ_x under different numbers of bins.

Number of bins	Discrete states	Boundary values
3	$\{-1, 0, 1\}$	$[-\infty,\ -7.5696,\ 6.4434,\ \infty]$
5	$\{-2, -1, 0, 1, 2\}$	$[-\infty,\ -16.3463,\ -4.3708,\ 3.4246,\ 15.5837,\ \infty]$
7	$\{-3, -2, -1, 0, 1, 2, 3\}$	$[-\infty,\ -22.0410,\ -10.2790,\ -3.1883,\ 2.2974,\ 9.1881,\ 21.6504,\ \infty]$
9	$\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$	$[-\infty,\ -26.4327,\ -14.5688,\ -7.5696,\ -2.5658,\ 1.6850,\ 6.4434,\ 13.7152,\ 26.2945,\ \infty]$

Lateral motion state assignment. For vehicles with $|\tilde{y}| > 0.5$, the lateral motion state is determined by the signs of Δy and Δv_y. For example, consider a neighboring vehicle to the right of the ego with $\Delta y < 0$ and $\Delta v_y > 0$. The positive lateral velocity indicates movement toward the ego lane, and with $|\tilde{y}| = 0.6$, the vehicle is in the boundary region, indicating it is closing in. If instead $\Delta v_y < 0$ at the same position, the vehicle is moving away from the ego lane and is therefore classified as pulling away. The assignment follows a risk-descending order, where closing in ($\sigma_{y,\text{motion}} = 0$) is the most hazardous, staying ($\sigma_{y,\text{motion}} = 1$) is intermediate, and pulling away ($\sigma_{y,\text{motion}} = 2$) is the least risky.

A.2 IMPORTANCE FUNCTION DETAILS

Figure 6 illustrates the shape of the interaction importance function (Eq. 3). The exponents α and β determine how rapidly the importance score decreases as the longitudinal and lateral interaction states become weaker; larger values produce sharper contrast between high- and low-relevance neighbors. The decay weights λ_x and λ_y control the relative dominance of the two axes, allowing the function to emphasize longitudinal-dominant, balanced, or lateral-dominant interaction regimes.

A.3 REPRESENTATION HYPERPARAMETERS

Table 6 summarizes all fixed hyperparameters used to compute the proposed interaction representation across all datasets and backbone models.

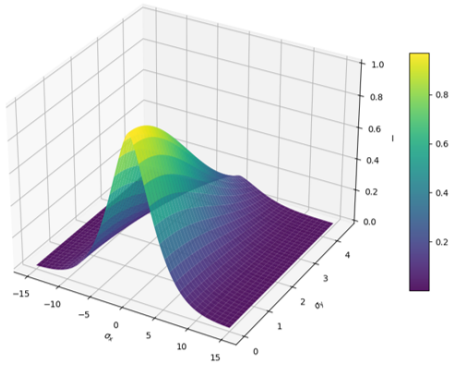


Figure 6: Importance function visualization ($\lambda_x = 0.1,\ \lambda_y = 0.1,\ \alpha = 1.5,\ \beta = 2.0$).

Table 6: Hyperparameters used for the proposed representation.

Component	Value	Description
ϵ	0.3	Numerical stability constant for longitudinal interaction time (Eq. 1).
λ_x	0.1	Longitudinal decay weight in the importance function (Eq. 3).
λ_y	0.1	Lateral decay weight in the importance function (Eq. 3).
α	1.5	Sensitivity exponent for the longitudinal interaction state (Eq. 3).
β	2.0	Sensitivity exponent for the lateral interaction state (Eq. 3).
γ	0.3	Slot-semantic amplification factor (Eq. 4).
n	2	Number of neighbors retained after importance-based filtering (Eq. 5).

A.4 Slot-Semantic Weighting Details

A.4.1 Context Definition

The driving context set $\mathcal{C}$ consists of lane-following and lane-change contexts. For lane-following samples, the context is determined by the ego vehicle's lane position, resulting in three cases: leftmost, middle, and rightmost lane-following. For lane-change samples, we consider four lane-change groups: leftmost-to-right, middle-to-right, middle-to-left, and rightmost-to-left. Each lane-change group is further divided into pre-LC and post-LC stages according to whether the current frame occurs before or after the ego vehicle crosses the lane boundary.

A.4.2 Slot-Conditioned Weight Estimation

The slot-conditioned weight $w_{\text{slot},c}$ is estimated by a context-specific MLP-based weighting network trained independently of the main prediction model. Each training sample is first assigned to one of the predefined driving contexts $c \in \mathcal{C}$ using the ego maneuver label and lane information. For each occupied slot, the network takes as input the slot-position encoding, the corresponding neighbor's kinematic features, and the ego state features. It produces an unnormalized score for each occupied slot, and the scores are converted into normalized slot weights using a masked softmax. Unoccupied slots are excluded from the softmax normalization by applying the occupancy mask.

Since ground-truth slot importance labels are unavailable, the weighting network is trained using an auxiliary trajectory prediction objective. The normalized slot weights are used to aggregate the occupied neighbors' kinematic features, and the aggregated neighbor feature is combined with the ego feature to predict the future ego trajectory. The network is optimized by minimizing the auxiliary prediction loss:

$$\mathcal{L}_{aux} = \frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{p}}_t^{ego} - \mathbf{p}_t^{ego}\|_2^2 .$$

After training, the fixed slot-conditioned weight is computed as the validation-set mean of the learned normalized slot weights:

$$w_{\text{slot},c} = \frac{1}{|\mathcal{D}^{val}_{c,\text{slot}}|} \sum_{i \in \mathcal{D}^{val}_{c,\text{slot}}} \tilde{w}^{(i)}_{\text{slot},c},$$

where $\mathcal{D}^{val}_{c,\text{slot}}$ denotes the set of validation samples in context c where slot is occupied, and $\tilde{w}^{(i)}_{\text{slot},c}$ denotes the normalized slot weight predicted by the auxiliary weighting network.

A.4.3 Learned Slot-Semantic Weights

Tables 7 and 8 report the learned slot-semantic weights for lane-following and lane-change contexts, respectively.

Table 7: Learned slot-semantic weights for lane-following contexts.

Ego lane context	Preceding	Following	Left-preceding	Left-alongside	Left-following	Right-preceding	Right-alongside	Right-following
Leftmost lane	0.4255	0.0336	0.0000	0.0000	0.0000	0.4574	0.0119	0.1190
Middle lane	0.4805	0.0002	0.0000	0.0000	0.0000	0.3803	0.0234	0.1839
Rightmost lane	0.4784	0.0373	0.3344	0.0343	0.2050	0.0000	0.0000	0.0000

Table 8: Learned slot-semantic weights for pre-/post-lane-change contexts.

Lane-change group	Preceding	Following	Left-preceding	Left-alongside	Left-following	Right-preceding	Right-alongside	Right-following
pre-LC0: leftmost → right	0.0001	0.0000	0.0000	0.0000	0.0000	0.6253	0.2663	0.3117
pre-LC1: middle → right	0.0072	0.0263	0.0006	0.0000	0.0000	0.3970	0.3776	0.5494
pre-LC2: middle → left	0.0183	0.1326	0.6745	0.5179	0.2365	0.0000	0.0000	0.0000
pre-LC3: rightmost → left	0.0381	0.0233	0.5755	0.3548	0.4799	0.0000	0.0000	0.0000
post-LC0: leftmost → right	0.0460	0.3983	0.0000	0.0023	0.0762	0.2338	0.2022	0.3281
post-LC1: middle → right	0.1036	0.0851	0.4832	0.0540	0.3810	0.0013	0.0000	0.0002
post-LC2: middle → left	0.6018	0.3591	0.0115	0.0013	0.0099	0.1709	0.0069	0.0014
post-LC3: rightmost → left	0.2618	0.0000	0.0036	0.0000	0.0000	0.6545	0.2032	0.1449

B Model-Specific Input Adaptation

The evaluated models employ different input formats and interaction modeling mechanisms. We therefore construct model-specific adapters to incorporate the proposed interaction importance into each backbone while preserving its prediction architecture.

For CS-LSTM, the interaction importance is appended to the neighbor features represented in the social pooling grid. For GRIP++, it is included in the neighbor features used to construct the ego-neighbor interaction graph. SGAN incorporates the interaction importance into the neighbor features processed by its social pooling module. For NeighFormer, the importance is appended to each neighbor token before the neighbor projection layer.

For QCNet and MTP-GO, the interaction importance is represented pairwise as $I_{i,j}$, where each predicted agent i serves as the reference agent with respect to a neighboring agent j. QCNet incorporates $I_{i,j}$ into its agent-to-agent relation representation, whereas MTP-GO includes it in the corresponding edge attributes.

For each backbone, the baseline and $+I$ conditions follow the same training and evaluation protocol, with only the input interface adapted as necessary to incorporate the proposed interaction importance.