DeMo: Decoupling Motion Forecasting into Directional Intentions and Dynamic States

Bozhou Zhang **Nan Song** **Li Zhang***
School of Data Science, Fudan University

https://github.com/fudan-zvg/DeMo

Abstract

Accurate motion forecasting for traffic agents is crucial for ensuring the safety and efficiency of autonomous driving systems in dynamically changing environments. Mainstream methods adopt a one-query-one-trajectory paradigm, where each query corresponds to a unique trajectory for predicting multi-modal trajectories. While straightforward and effective, the absence of detailed representation of future trajectories may yield suboptimal outcomes, given that the agent states dynamically evolve over time. To address this problem, we introduce **DeMo**, a framework that decouples multi-modal trajectory queries into two types: mode queries capturing distinct directional intentions and state queries tracking the agent's dynamic states over time. By leveraging this format, we separately optimize the multi-modality and dynamic evolutionary properties of trajectories. Subsequently, the mode and state queries are integrated to obtain a comprehensive and detailed representation of the trajectories. To achieve these operations, we additionally introduce combined Attention and Mamba techniques for global information aggregation and state sequence modeling, leveraging their respective strengths. Extensive experiments on both the Argoverse 2 and nuScenes benchmarks demonstrate that our DeMo achieves state-of-the-art performance in motion forecasting.

1 Introduction

Motion forecasting [29, 58, 67] empowers self-driving vehicles to anticipate how surrounding agents will move and affect the ego car, providing references and conditions for the ego-action. It is critical for maintaining safety and dependability, enabling vehicles to comprehend the dynamics of driving environments and make calculated decisions. The challenges and complexities of this task arise from various factors, including unpredictable road conditions, varied movement patterns of traffic participants, and the necessity to simultaneously analyze the states of observed agents along with the road maps.

The research community has witnessed significant progress in the representation of driving scenes [18, 36, 44] and the paradigm of trajectory decoding [26, 38, 54, 77]. These methods have achieved substantial advancements in prediction accuracy, primarily following a certain pattern inspired from detection [4, 40], *i.e.*, the one-query-one-trajectory paradigm [38, 54, 59, 77]. This paradigm utilizes several queries to represent different estimated trajectories, as shown in Figure 1 (a), covering the possibilities of distinct motion intentions. Although effective, these approaches can only approximately provide a direction and collect surroundings to generate various trajectory waypoints in a one-shot fashion, overlooking the detailed relationships with scenes. The lack of concrete representation for trajectories and comprehensive spatiotemporal interactions with the surrounding

*Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3383

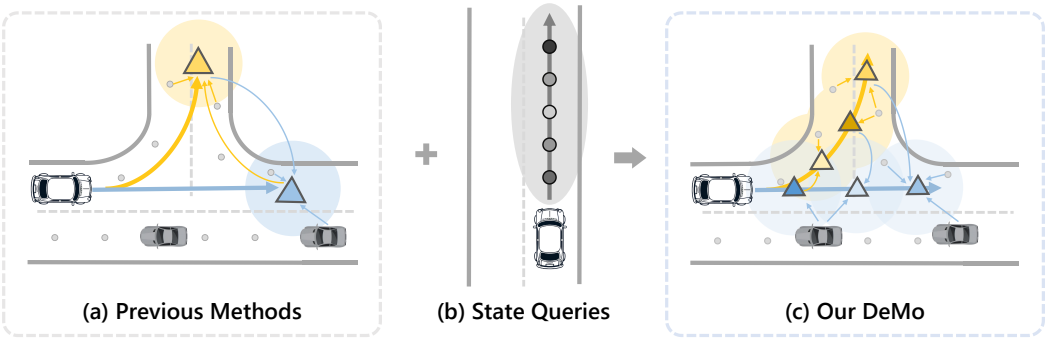

Figure 1: The primary distinction between previous methods and ours lies in the representation of future trajectories. Previous methods, as depicted in (a), use only one mode query for each trajectory. Our approach, illustrated in (c), utilizes decoupled mode queries, as shown in (a), and state queries, as shown in (b), to represent the multi-modal trajectories.

environment and among each other might lead to a decline in accuracy and consistency across varying time steps.

To solve this problem, we propose a novel framework dubbed **DeMo**, which provides a detailed representation of multi-modal trajectories. Specifically, we decouple forecasting queries into two types: besides the original motion mode queries to capture different directional intentions as shown in Figure 1 (a), we introduce the dynamic state queries for future trajectories to track the agent's dynamic states across various time steps, as shown in Figure 1 (b). This approach allows us to achieve a comprehensive query representation within our framework, as illustrated in Figure 1 (c). Mode queries and state queries are processed using the Mode Localization Module and the State Consistency Module, respectively. These modules enable the explicit interactions of queries with the surrounding environments and among each other, by which the directional accuracy and temporal consistency of future trajectories are significantly optimized. Subsequently, two types of queries are integrated by our Hybrid Coupling Module to achieve a comprehensive representation of future trajectories. Due to the sequential nature of trajectory states, Mamba is particularly well-suited for modeling the temporal consistency of dynamic states. Therefore, we utilize a combination of Attention and Mamba in our modules to effectively and efficiently aggregate global information and model state sequences, leveraging the strengths of both techniques.

Our contributions are summarized as follows: **(i)** We propose a motion forecasting framework that decouples multi-modal trajectory queries into mode queries and state queries to represent directional intentions and dynamic states, respectively. **(ii)** We design three modules based on integrated Attention and Mamba to process decoupled mode queries, state queries, and coupled mode and state queries. **(iii)** Extensive experiments on both the Argoverse 2 and nuScenes benchmarks demonstrate that DeMo achieves state-of-the-art performance.

2 Related work

Motion forecasting. In recent advancements in autonomous driving, it is critical to effectively predict the movements of relevant agents by accurately representing scene components. Traditional methods [5, 20, 48] transformed driving scenarios into image formats and used conventional convolutional networks for scene context encoding. However, these techniques often failed to sufficiently capture intricate structural details. This challenge has led to the adoption of vectorized scene representations [26, 60, 75, 79], exemplified by the introduction of VectorNet [18]. Additionally, graph-based structures are also widely utilized to represent the relationships between agents and their environments [14, 21, 30, 31, 36, 50, 69].

Existing methodologies have delved into a variety of frameworks to predict multi-modal future trajectories given the scene features. Initially, prediction techniques were centered on goal-oriented methods [26, 71] or employed probability heatmaps to sample trajectories [20, 21]. However, contemporary strategies, such as MTR [54] and QCNet [77], among others [41, 43, 44, 73], utilize Transformer [61] models to analyze relationships within the scene. Additionally, the introduction

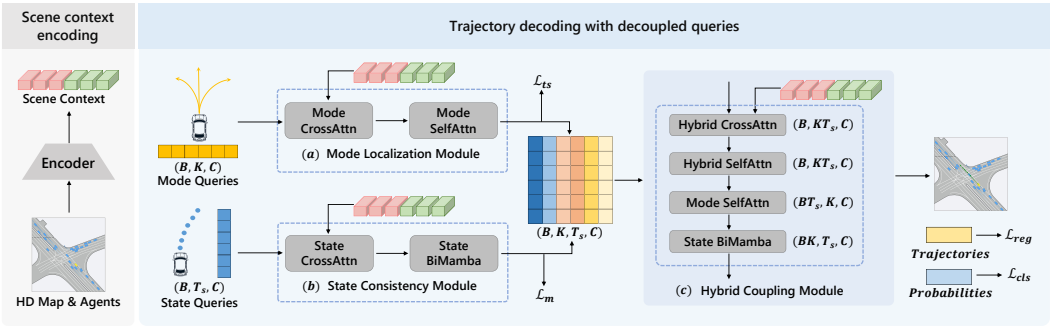

Figure 2: Overview of our DeMo framework: The HD maps and agents are first processed by the encoder to obtain the scene context. The decoding pipeline includes: (a) the Mode Localization Module, which processes mode queries by interacting with the scene context from the encoder and among themselves; (b) the State Consistency Module, which processes state queries; and (c) the Hybrid Coupling Module, which combines these queries to generate the final output. The feature dimension is illustrated using a single-agent setting, where B represents the batch size.

of novel paradigms such as pre-training [7, 8, 34], historical prediction design [46, 59], GPT-style next-token prediction [49, 53], and post-refinement [9, 76] in some techniques has led to remarkable advancements in performance.

Furthermore, the advancements in multi-agent forecasting aim to enhance the applicability of predicted trajectories for various agents in real-world scenarios. Several approaches [22, 55, 79] follow an agent-centric model, where trajectories are forecasted individually for each agent, a process that might be slow. On the other hand, alternative approaches [44, 78] utilize a scene-centric model that allows for simultaneous forecasting across all agents, introducing an innovative approach to trajectory prediction.

Inspired by the progress in object detection and motivated by its significant success [4, 40], mainstream methods [38, 54, 59, 77] have adopted a one-query-one-trajectory paradigm to achieve high performance in motion forecasting benchmarks [6, 58, 67]. These methods use transformers to model the relationship between each trajectory query and its environment but lack detailed representation. We propose decoupled mode queries and state queries for a more detailed and comprehensive representation of multi-modal trajectories.

State space models. Originally developed for modeling dynamic systems with state variables in fields such as control theory, state space models (SSMs) have emerged as promising alternatives to Transformers [61] in sequence modeling, particularly due to their effectiveness in addressing attention complexity and capturing long-term dependencies. As SSMs have evolved [17, 25, 56], a new class termed Mamba [24], which incorporates selection mechanisms and hardware-aware architectures, has recently demonstrated significant promise in long-sequence modeling. Several studies have explored Mamba's substantial potential across a range of fields, including natural language processing [27, 37] and computer vision [28, 35, 74, 80]. Notably, in the vision domain, Mamba has demonstrated superior GPU efficiency and effectiveness compared to Transformers in tasks such as visual representation learning [80], video understanding [35], and human motion generation [74]. Building on these achievements, to the best of our knowledge, this is the first method to combine the strengths of Mamba with mainstream Transformer-based architecture to achieve impressive performance.

3 Methodology

In this section, we present DeMo, which utilizes decoupled mode queries and state queries for directional intentions and dynamic states to predict future trajectories. We also employ a hybrid architecture combining Attention and Mamba, along with two auxiliary losses for feature modeling.

https://doi.org/10.52202/079017-3383

3.1 Problem formulation

Given HD map and agents in the driving scenario, motion forecasting aims to predict the future trajectories for the interested agents. The HD map comprises several polylines of lanes or crossings, while agents are traffic participants like vehicles and pedestrians. To transform these elements into easily processable and learnable inputs, we utilize a popular vectorized representation following [18]. Specifically, the map $\mathrm{M} \in \mathbb{R}^{N_\mathrm{m} \times L \times C_\mathrm{m}}$ is generated by dividing each line into several shorter segments, where N_m, L, and C_m denote the number of map polylines, divided segments and feature channels, respectively. We represent the historical information of agents as $\mathrm{A} \in \mathbb{R}^{N_\mathrm{a} \times T_\mathrm{h} \times C_\mathrm{a}}$, where N_a, T_h, and C_a are the number of agents, historical timestamps, and motion states (e.g., position, heading angle, velocity). Additionally, the future trajectories $A_\mathrm{f} \in \mathbb{R}^{N_\mathrm{aoi} \times T_\mathrm{f} \times 2}$ for agents of interest are estimation objectives, with N_aoi, T_f indicating the number of selected agents and the future timestamps, respectively.

3.2 Scene context encoding

Given the vectorized representations A for agents and M for HD map, we first employ individual encoders to process them separately. Specifically, we use a PointNet-based polyline encoder, as described in [8, 54, 55], to process the map representation M, generating the map features $F_\mathrm{m} \in \mathbb{R}^{N_\mathrm{m} \times C}$. For the agents A, we replace generic Transformer [61] or RNN with several Unidirectional Mamba [24] blocks, which are more efficient and effective for sequence encoding, to aggregate the historical trajectory features $F_\mathrm{a} \in \mathbb{R}^{N_\mathrm{a} \times C}$ up to the current time. Subsequently, the scene context features $F_\mathrm{s} \in \mathbb{R}^{(N_\mathrm{a}+N_\mathrm{m}) \times C}$ are formed by concatenating them and further propagated to a Transformer encoder for intra-interaction learning. The overall process can be formulated as:

$$F_\mathrm{m} = \mathrm{PointNet}(\mathrm{M}), \quad F_\mathrm{a} = \mathrm{UniMamba}(\mathrm{A}), \quad F_\mathrm{s} = \mathrm{Transformer}(\mathrm{Concat}(F_\mathrm{a}, F_\mathrm{m})). \tag{1}$$

3.3 Trajectory decoding with decoupled queries

After obtaining the scene context features, we aim to decode multi-modal future trajectories for each interested agent based on our proposed decoupled queries. As illustrated in Figure 2, the decoder network comprises a State Consistency Module that enhances the consistency and accuracy of dynamic future state queries, a Mode Localization Module for learning distinct motion modes, and a Hybrid Coupling Module to integrate the decoupled queries and generate the final output. The detailed description of these components are provided in the following.

Dynamic state consistency. Considering the recurrence and causality of the future trajectories A_f, we propose to represent them as a series of dynamic states across various time steps, distinct yet interconnected. To preserve precise time information, the state queries $Q_\mathrm{s} \in \mathbb{R}^{N_\mathrm{aoi} \times T_\mathrm{s} \times C}$ are initialized with an MLP module for real-time differences. It is notable that the steps T_s can differ from T_f to balance the effectiveness and efficiency, especially when predicting long-term future trajectories or a higher frequency of future trajectories. The State Consistency Module is then employed to enhance the consistency of the state queries and aggregate the specific scene context, which can be formulated as follows:

$$\begin{aligned} Q_\mathrm{s} &= \mathrm{MLP}([t_1, t_2, \cdots, t_{T_\mathrm{s}}]), \\ Q_\mathrm{s} &= \mathrm{MultiHeadAttn}(\mathrm{Q} = Q_\mathrm{s}, \mathrm{K} = F_\mathrm{s}, \mathrm{V} = F_\mathrm{s}), \\ Q_\mathrm{s} &= \mathrm{BiMamba}(Q_\mathrm{s}). \end{aligned} \tag{2}$$

Specifically, cross-attention is first applied to enable state queries to interact with the scene context, followed by a Mamba block to model sequence relationships with linear-time complexity. Simultaneously, to account for the influences of rear state queries on the front ones, we adopt the bidirectional Mamba [35, 80] for both forward and backward scanning. Additionally, a simple MLP module is utilized to decode the state queries Q_s into a single future trajectory for explicit supervision of time consistency.

Directional intention localization. Mode queries $Q_{\mathrm{m}} \in \mathbb{R}^{N_{\mathrm{aoi}} \times K \times C}$ represent different motion modes, with each query responsible for decoding one of the K trajectories. We utilize the Mode Localization Module to localize the potential directional intentions, as shown below:

$$\begin{aligned} Q_{\mathrm{m}} &= \mathrm{MultiHeadAttn}(\mathrm{Q} = Q_{\mathrm{m}}, \mathrm{K} = F_{\mathrm{s}}, \mathrm{V} = F_{\mathrm{s}}), \\ Q_{\mathrm{m}} &= \mathrm{MultiHeadAttn}(\mathrm{Q} = Q_{\mathrm{m}}, \mathrm{K} = Q_{\mathrm{m}}, \mathrm{V} = Q_{\mathrm{m}}). \end{aligned} \tag{3}$$

For spatial motion learning, two Multi-Head Attention blocks are employed to enable interactions among mode queries and with the scene context. Additionally, we also employ simple MLPs to decode the future trajectories and probabilities. Similarly, we introduce another auxiliary supervision to endow mode queries with distinct motion intentions.

Hybrid query coupling. To incorporate dynamic states and directional intentions, we simply add Q_{m} and Q_{s} together to form the hybrid spatiotemporal queries $Q_{\mathrm{h}} \in \mathbb{R}^{N_{\mathrm{aoi}} \times K \times T_{\mathrm{s}} \times C}$. Then, the Hybrid Coupling Module is utilized to further process Q_{h} and yield a comprehensive representation for future trajectories, as formulated below:

$$\begin{aligned} Q_{\mathrm{h}} &= \mathrm{MultiHeadAttn}(\mathrm{Q} = Q_{\mathrm{h}}, \mathrm{K} = F_{\mathrm{s}}, \mathrm{V} = F_{\mathrm{s}}), \\ Q_{\mathrm{h}} &= \mathrm{HybridMultiHeadAttn}(\mathrm{Q} = Q_{\mathrm{h}}, \mathrm{K} = Q_{\mathrm{h}}, \mathrm{V} = Q_{\mathrm{h}}), \\ Q_{\mathrm{h}} &= \mathrm{ModeMultiHeadAttn}(\mathrm{Q} = Q_{\mathrm{h}}, \mathrm{K} = Q_{\mathrm{h}}, \mathrm{V} = Q_{\mathrm{h}}), \\ Q_{\mathrm{h}} &= \mathrm{BiMamba}(Q_{\mathrm{h}}). \end{aligned} \tag{4}$$

Besides the Attention and Mamba modules for interaction with the scene context, among modes, and across time states, we additionally introduce a hybrid self-attention layer, which connects queries across both time and modes, boosting the diversity of predicted trajectories. The change in feature dimensions in this module is shown in Figure 2. The final predictions are generated by decoding the output Q_{h} into trajectory positions and probabilities with MLPs.

3.4 Training losses

DeMo is trained with three component losses in an end-to-end manner. Primarily, the regression loss $\mathcal{L}_{\mathrm{reg}}$ and the classification loss $\mathcal{L}_{\mathrm{cls}}$ are employed to supervise the accuracy of predicted trajectories and their associated probability scores. Additionally, we introduce two auxiliary losses, $\mathcal{L}_{\mathrm{ts}}$ and $\mathcal{L}_{\mathrm{m}}$, for intermediate features of time states and motion modes, respectively. The former enhances the coherence and causality of dynamic states across various time steps, while the latter endows mode with distinct directional intentions. The overall loss $\mathcal{L}$ is a combination of these individual losses with equal weights, formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{reg}} + \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{ts}} + \mathcal{L}_{\mathrm{m}}. \tag{5}$$

We adopt the cross-entropy loss for probability score classification and the smooth-L1 loss for trajectory regression tasks. The winner-take-all strategy is employed, optimizing only the best prediction with minimal average prediction error to the ground truth.

4 Experiments

4.1 Experimental settings

Datasets. We evaluate our method's performance using the Argoverse 2 [67] and nuScenes [3] motion forecasting datasets. The Argoverse 2 dataset comprises 250,000 scenarios with a sampling frequency of 10 Hz, each featuring a 5s historical trajectory length and predicting a 6s future ones. The nuScenes dataset contains 1,000 scenes at 2 Hz, predicting the next 6s trajectories with the past 2s history.

https://doi.org/10.52202/079017-3383

Table 1: Performance comparison on *Argoverse 2 single-agent test set* in the official leaderboard. For each metric, the best result is in **bold** and the second best result is underlined. The upper part features a single model, while the lower part employs model ensembling as a trick.

Method	*minFDE*$_1$	*minADE*$_1$	*minFDE*$_6$	*minADE*$_6$	*MR*$_6$	*b-minFDE*$_6$
FRM [47]	5.93	2.37	1.81	0.89	0.29	2.47
HDGT [31]	5.37	2.08	1.60	0.84	0.21	2.24
SIMPL [72]	5.50	2.03	1.43	0.72	0.19	2.05
THOMAS [22]	4.71	1.95	1.51	0.88	0.20	2.16
GoRela [11]	4.62	1.82	1.48	0.76	0.22	2.01
MTR[54]	4.39	1.74	1.44	0.73	0.15	1.98
HPTR [73]	4.61	1.84	1.43	0.73	0.19	2.03
GANet [64]	4.48	1.77	1.34	0.72	0.17	1.96
ProphNet [65]	4.74	1.80	1.33	0.68	0.18	1.88
QCNet [77]	4.30	1.69	1.29	0.65	0.16	1.91
SmartRefine [76]	4.17	1.65	1.23	0.63	0.15	1.86
DeMo (Ours)	**3.74**	**1.49**	**1.17**	**0.61**	**0.13**	**1.84**
QML [57]	4.98	1.84	1.39	0.69	0.19	1.95
TENET [66]	4.69	1.84	1.38	0.70	0.19	1.90
MacFormer [15]	4.69	1.84	1.38	0.70	0.19	1.90
BANet [70]	4.61	1.79	1.36	0.71	0.19	1.92
Gnet [19]	4.40	1.72	1.34	0.69	0.18	1.90
Forecast-MAE [8]	4.15	1.66	1.34	0.69	0.17	1.91
QCNet [77]	3.96	1.56	1.19	0.62	0.14	1.78
DeMo (Ours)	**3.70**	**1.49**	**1.11**	**0.60**	**0.12**	**1.73**

Evaluation metrics. We adopt the common metrics: $minADE$, $minFDE$, MR, and $b\text{-}minFDE$. For multi-agent scenarios, we use $avgMinADE$, $avgMinFDE$, and $actorMR$. The Argoverse 2 dataset is evaluated across six prediction modes, while nuScenes is evaluated across ten prediction modes. We typically follow the evaluation metrics from the official leaderboard, setting K to 1 and 6 for the Argoverse 2 dataset, and K to 5 and 10 for the nuScenes dataset.

Implementation details. Our models are trained for 60 epochs using the AdamW [42] optimizer, with a batch size of 16 per GPU. The training is conducted end-to-end with a learning rate of 0.003 and a weight decay of 0.01. We adopt an agent-centric coordinate system and sample scene elements within a 150-meter radius of the agents of interest. All experiments are conducted on 8 NVIDIA GeForce RTX 3090 GPUs. Additional details and further experiments are provided in Appendix A and Appendix B.

4.2 Comparison with state of the art

We first compare our method, DeMo with several models on the Argoverse 2 [67] motion forecasting benchmark for the single-agent setting as demonstrated in Table 1. To ensure a comprehensive and fair comparison, we separately evaluate the performance of different methods with and without the model ensembling technique. It is shown that DeMo has significantly outperformed all previous approaches, including the state-of-the-art model QCNet [77] and its post-refinement enhancement SmartRefine [76]. Concretely, our method distinctly surpasses other methods across all metrics, particularly in terms of $minFDE_1$ and $minADE_1$, where it demonstrates performance improvements of 13.02% and 11.83% relative to QCNet, respectively. After using ensembling techniques similar to other entries, DeMo surpasses all methods on all metrics by a large margin. Then, we compare the performance of DeMo on the nuScenes [3] motion forecasting benchmark, with the results of the test split presented in Table 2. Our method is also superior to others over all metrics except $minADE_5$.

Table 2: Performance comparison on *nuScenes test set* in the official leaderboard. The "-" symbol means the corresponding metric is unknown.

Method	$minFDE_1$	$minADE_5$	$minADE_{10}$	MR_5	MR_{10}
Trajectron++ [51]	9.52	1.88	1.51	0.70	0.57
LaPred [33]	8.37	1.47	1.12	0.53	0.46
P2T [13]	10.50	1.45	1.16	0.64	0.46
GOHOME [21]	6.99	1.42	1.15	0.57	0.47
CASPNet [52]	-	1.41	1.19	0.60	0.43
Autobot [23]	8.19	1.37	1.03	0.62	0.44
THOMAS [22]	6.71	1.33	1.04	0.55	0.42
PGP [14]	7.17	1.27	0.94	0.52	**0.34**
LAformer [39]	6.95	**1.19**	1.19	0.48	0.48
DeMo (Ours)	**6.60**	1.22	**0.89**	**0.43**	**0.34**

Table 3: Performance comparison on *Argoverse 2 multi-agent test set* in the official leaderboard.

Method	$avgMinFDE_1$	$avgMinADE_1$	$avgMinFDE_6$	$avgMinADE_6$	$actorMR_6$
FJMP [50]	4.00	1.52	1.89	0.81	0.23
Forecast-MAE [8]	3.33	1.30	1.55	0.69	0.19
FFINet [32]	3.18	1.24	1.77	0.77	0.24
Gnet [19]	3.05	1.23	1.46	0.69	0.19
DeMo (Ours)	**2.78**	**1.12**	**1.24**	**0.58**	**0.16**

4.3 Multi-agent quantitative results

In multi-agent environments, it is essential for predictors to simultaneously forecast the future paths of all relevant agents to comprehensively understand the driving situation. To validate the efficacy of our model, DeMo, we conduct tests on the Argoverse 2 multi-agent dataset [67]. The results, presented in Table 3, show that despite lacking the specialized multi-agent forecasting features found in models such as [44, 55, 78], our model surpasses recent advancements across all evaluated metrics due to our novel designs.

4.4 Ablation study

Effects of components. Table 4 demonstrates the effectiveness of each component in our method. We show the baseline in the first row, which is similar to previous methods [54, 77] and utilizes mode queries to generate multi-modal future trajectories. Then, we directly adopt state queries in the second row (ID-2) to decode the trajectories. A performance decline is observed due to the surplus queries, which impose a burden on the model and make it difficult to distinguish the meanings of different types. In the third row (ID-3), we introduce two auxiliary losses, resulting in a slight improvement compared to the first row. Although the model can identify what each query represents, it demonstrates only moderate performance due to the limited information. In the fourth row (ID-4), we incorporate the three aggregation modules in Figure 2 but remove auxiliary losses, leading to significant performance enhancements. Finally, in the fifth row (ID-5), our DeMo integrates all these techniques and achieves outstanding performance.

Effects of state sequence modeling with Mamba. Mamba excels at sequence modeling, so we utilize Bidirectional Mamba [35, 80] to enhance the consistency of states across different time steps. To demonstrate its effectiveness, we compare Bidirectional Mamba with several other modules, including Unidirectional Mamba [24], Attention, Conv1d, and GRU [10]. As illustrated in the left part of Table 5, our Bidirectional Mamba configuration outperforms the others due to its specialized design for sequence modeling, compared to Attention, and its capability to perform both forward and backward scans, unlike Unidirectional Mamba.

https://doi.org/10.52202/079017-3383

Table 4: Ablation study on the core components of DeMo on the *Argoverse 2 single-agent validation set*. "Decpl. Query" indicates decoupled query paradigm. "Agg. Module" indicates three aggregation modules. "Aux. Loss" indicates two auxiliary losses.

ID	State Query	Decpl. Query	Agg. Module	Aux. Loss	$minFDE_1$	$minADE_1$	$minFDE_6$	$minADE_6$	MR_6	$b\text{-}minFDE_6$
1					4.489	1.792	1.414	0.750	0.184	2.067
2	✓				4.494	1.800	1.505	0.777	0.208	2.138
3	✓	✓		✓	4.385	1.746	1.405	0.761	0.180	2.051
4	✓	✓	✓		4.247	1.695	1.319	0.687	0.166	1.961
5	✓	✓	✓	✓	**3.917**	**1.609**	**1.268**	**0.674**	**0.152**	**1.918**

Table 5: Ablation study on (a) (left) the sequence modeling choices and (b) (right) the effects of aggregation modules and auxiliary losses. For (a), "Uni-MB" and "Bi-MB" represent Unidirectional Mamba and Bidirectional Mamba. For (b), "H.C." indicates Hybrid Coupling Module. "S.C." indicates State Consistency Module. "M.L." indicates Mode Localization Module.

	$minFDE_6$	$minADE_6$	MR_6
None	1.307	0.692	0.161
GRU	1.842	0.923	0.274
Conv1d	1.304	0.693	0.161
Attn	1.289	0.687	0.159
Uni-MB	1.288	0.690	0.156
Bi-MB	**1.268**	**0.674**	**0.152**

	$minFDE_6$	$minADE_6$	MR_6
w/o $\mathcal{L}_{\text{ts}}$	1.290	0.715	0.161
w/o $\mathcal{L}_{\text{m}}$	1.289	0.687	0.159
w/o H.C.	1.324	0.704	0.164
w/o S.C.	1.317	0.697	0.162
w/o M.L.	1.297	0.693	0.158
All	**1.268**	**0.674**	**0.152**

Effects of auxiliary losses and aggregation modules. We conduct an ablation study to assess the impacts of auxiliary losses and aggregation modules. As illustrated in the right part of Table 5, removing any of these losses or modules leads to a performance decline in the model. Notably, the aggregation modules have a greater impact than the auxiliary losses. This is attributed to the critical role of learning information from the scene context and from each other, which is essential for decoupling queries to represent distinct meanings.

Effects of state queries. We conduct an ablation study on the number of state queries, as shown in the left part of Table 6. In our default setting, we use 60 state queries to represent the future states at 60 timestamps. As we gradually reduce the number of state queries, we observe a performance decline due to the increasing ambiguity of the state query meanings.

Effects of the depth of Attention and Mamba blocks. A suitable depth configuration of Attention and Mamba units is crucial for achieving an optimal balance between efficiency and performance. As depicted in the right part of Table 6, we conduct an ablation study focusing on the layer depth. It is observed that the best results are obtained with Attention units at a depth of three and Mamba units at a depth of two.

Table 6: Ablation study on (a) (left) state queries and (b) (right) the depth of Attention and Mamba layers.

Queries	$minFDE_6$	$minADE_6$	MR_6
10	1.312	0.704	0.160
20	1.294	0.688	0.157
30	1.290	0.692	0.155
60	**1.268**	**0.674**	**0.152**

Attn	M.B.	$minFDE_6$	$minADE_6$	MR_6
1	1	1.309	0.708	0.160
2	2	1.288	0.691	0.157
3	2	**1.268**	**0.674**	**0.152**
3	3	1.276	0.675	0.154

Effects of the depth of Mamba blocks in the encoder. We add ablation studies on the Mamba for encoding agent historical information in the encoder of our DeMo. As shown in Table 7, the left part shows different modules for encoding the historical information of agents. Our goal is to aggregate historical information up to the present time, making Unidirectional Mamba the most suitable choice.

The right part presents an ablation study concerning the number of Mamba blocks, indicating that three layers yield the optimal performance.

Table 7: Ablation study on (a) (left) sequence modeling choices and (b) (right) depth of Mamba blocks in agent historical information encoding.

	$minFDE_6$	$minADE_6$	MR_6	Num	$minFDE_6$	$minADE_6$	MR_6
GRU	1.344	0.726	0.170	1	1.312	0.701	0.162
Bi-MB	1.280	0.684	0.154	2	1.283	0.681	0.155
Uni-MB	**1.268**	**0.674**	**0.152**	3	**1.268**	**0.674**	**0.152**

4.5 An analysis to improve the measurement of query decoupling

We measure the outputs of state queries and mode queries with $minADE$ and $minFDE$, as shown in Table 8. We can see that the $minADE_1$ and $minFDE_1$ of the trajectories from state query outputs are better than those from mode query outputs. This means state dynamics are encoded in state queries. Additionally, there are six output trajectories from mode queries, indicating that directional information is predominantly stored in mode queries. The final outputs take advantage of the strengths of both.

Table 8: An analysis to improve the measurement of query decoupling.

	$minFDE_1$	$minADE_1$	$minFDE_6$	$minADE_6$
state query outputs	3.84	1.52	-	-
mode query outputs	4.12	1.63	1.31	0.67
final outputs	3.93	1.54	1.24	0.64

4.6 Efficiency analysis and qualitative results

Balancing performance, inference speed, and model size is important for the model deployment. We compare our DeMo with two recent, representative models: the state-of-the-art QCNet [77] and its enhancement through post-refinement, SmartRefine [76]. The size of our model is 5.9M, in contrast to 7.7M for QCNet and 8.0M for SmartRefine. Despite its smaller size, our model demonstrates superior performance by a significant margin, as detailed in Table 1. As for inference speed, we compare DeMo with QCNet, both end-to-end methods. Measurements are conducted on the Argoverse 2 single-agent validation set using an NVIDIA GeForce RTX 3090 GPU, with a maintained batch size of one. The average inference speed of DeMo is only 38ms, which is approximately 2.5 times faster than QCNet's 94ms. This demonstrates that our method is not only superior to QCNet but also more efficient.

In Figure 3, we present qualitative results of our network. The results of the baseline model, which lacks the decoupled query paradigm, are shown in panel (a), while the results of our DeMo are shown in panel (b). From the first two rows, it is evident that by explicitly optimizing the dynamic states of future trajectories, our model predicts trajectories that are more accurate and closer to the ground truth. From the third row, it is apparent that our model can better capture potential directional intentions. Additional qualitative results and failure cases are detailed in Appendix D and E.

4.7 Computational cost compared to other methods

We provide a comparison of computational cost with resent representative methods in Table 9. The experiments are conducted on Argoverse 2 [67] dataset using 8 NVIDIA GeForce RTX 3090 GPUs.

https://doi.org/10.52202/079017-3383

Table 9: Computational cost compared to other methods.

Method	FLOPs	Training time	Memory	Parameter	Batch size
SIMPL [72]	19.7 GFLOPs	8h	14G	1.9M	16
QCNet [77]	53.4 GFLOPs	45h	16G	7.7M	4
DeMo (Ours)	22.8 GFLOPs	9h	12G	5.9M	16

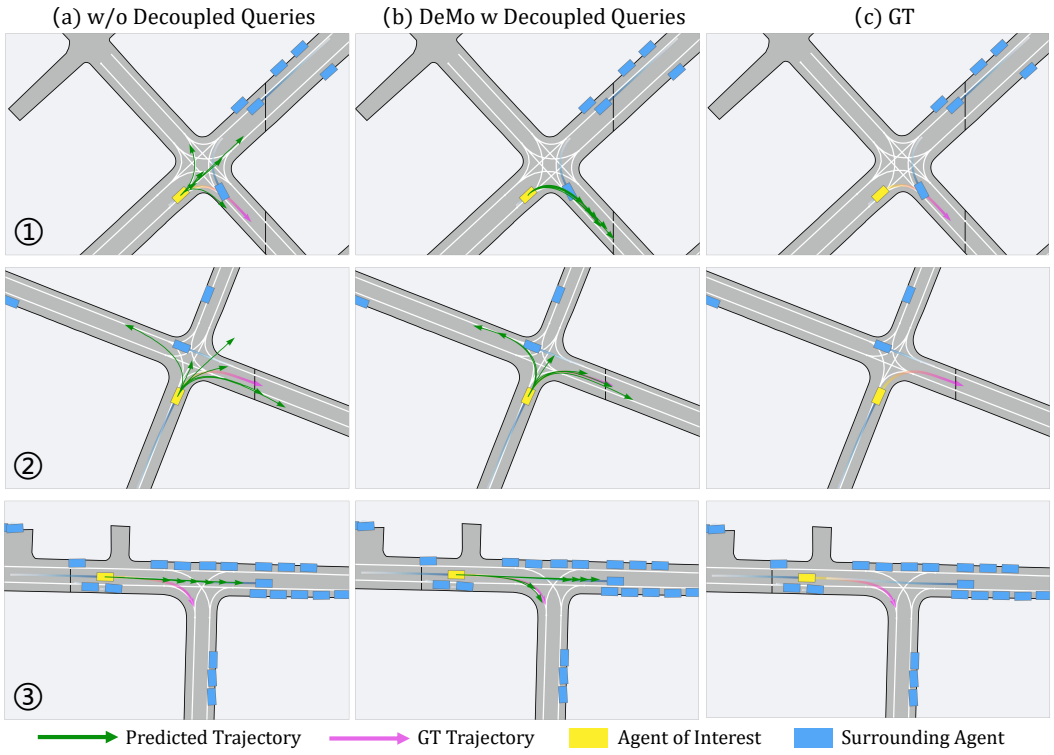

Figure 3: Qualitative results on the Argoverse 2 single-agent validation set. Panel (a) illustrates the results of the baseline model without decoupled queries; Panel (b) illustrates the results of our DeMo, which employs decoupled queries; and Panel (c) represents the ground truth.

5 Conclusion

In this paper, we introduce DeMo, which redefines the motion forecasting task by decoupling it into expressions of directional intentions and dynamic states. We utilize state queries to model various states across different time, and mode queries to capture the agent's motion intentions. Our approach incorporates three aggregation modules, combining Attention and Mamba for effective modeling. Comprehensive experiments, covering both single-agent and multi-agent scenarios, indicate that DeMo outperforms the current state-of-the-art methods. This highlights its potential as a promising approach for achieving safe and reliable motion forecasting in the rapidly advancing field of autonomous driving.

Limitations and future work. The proposed framework adopts a decoupled query paradigm, which may lead to heavier models due to the need to predict longer trajectories. Our current model design does not sufficiently take model efficiency into account. In the future, we plan to use sparse states for modeling trajectories, thereby making the framework more deployment-friendly.

https://doi.org/10.52202/079017-3383

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No. 62106050 and 62376060), Natural Science Foundation of Shanghai (Grant No. 22ZR1407500).

References

[1] G. Aydemir, A. K. Akan, and F. Güney. Adapt: Efficient multi-agent trajectory prediction with adaptation. In *ICCV*, 2023. 15

[2] P. Bhattacharyya, C. Huang, and K. Czarnecki. Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving. In *CoRL*, 2023. 15

[3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 5, 6

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 3

[5] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2020. 2

[6] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 3, 15

[7] H. Chen, J. Wang, K. Shao, F. Liu, J. Hao, C. Guan, G. Chen, and P.-A. Heng. Traj-mae: Masked autoencoders for trajectory prediction. In *ICCV*, 2023. 3

[8] J. Cheng, X. Mei, and M. Liu. Forecast-mae: Self-supervised pre-training for motion forecasting with masked autoencoders. In *ICCV*, 2023. 3, 4, 6, 7

[9] S. Choi, J. Kim, J. Yun, and J. W. Choi. R-pred: Two-stage motion prediction via tube-query attention-based trajectory refinement. In *ICCV*, 2023. 3, 15

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint*, 2014. 7

[11] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. In *ICRA*, 2023. 6

[12] F. Da and Y. Zhang. Path-aware graph attention for hd maps in motion prediction. In *ICRA*, 2022. 15

[13] N. Deo and M. M. Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint*, 2020. 7

[14] N. Deo, E. Wolff, and O. Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *CoRL*, 2022. 2, 7

[15] C. Feng, H. Zhou, H. Lin, Z. Zhang, Z. Xu, C. Zhang, B. Zhou, and S. Shen. Macformer: Map-agent coupled transformer for real-time and robust trajectory prediction. *IEEE RA-L*, 2023. 6

[16] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. In *ECCV*, 2024. 15

[17] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Re. Hungry hungry hippos: Towards language modeling with state space models. In *ICLR*, 2023. 3

[18] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*, 2020. 1, 2, 4

[19] X. Gao, X. Jia, Y. Li, and H. Xiong. Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks. *IEEE RA-L*, 2023. 6, 7

[20] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde. Home: Heatmap output for future motion estimation. In *IEEE ITSC*, 2021. 2

[21] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. In *ICRA*, 2022. 2, 7

[22] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde. THOMAS: Trajectory heatmap output with learned multi-agent sampling. In *ICLR*, 2022. 3, 6, 7

[23] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *ICLR*, 2022. 7

[24] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint*, 2023. 3, 4, 7

[25] A. Gu, K. Goel, and C. Re. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022. 3

[26] J. Gu, C. Sun, and H. Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *ICCV*, 2021. 1, 2, 15

[27] W. He, K. Han, Y. Tang, C. Wang, Y. Yang, T. Guo, and Y. Wang. Densemamba: State space models with dense hidden connection for efficient large language models. *arXiv preprint*, 2024. 3

 https://doi.org/10.52202/079017-3383

[28] V. T. Hu, S. A. Baumann, M. Gui, O. Grebenkova, P. Ma, J. Fischer, and B. Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *ECCV*, 2024. 3

[29] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE T-IV*, 2022. 1

[30] X. Jia, L. Sun, H. Zhao, M. Tomizuka, and W. Zhan. Multi-agent trajectory prediction by combining egocentric and allocentric views. In *CoRL*, 2022. 2

[31] X. Jia, P. Wu, L. Chen, Y. Liu, H. Li, and J. Yan. Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *IEEE TPAMI*, 2023. 2, 6

[32] M. Kang, S. Wang, S. Zhou, K. Ye, J. Jiang, and N. Zheng. Ffinet: Future feedback interaction network for motion forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 7

[33] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi. Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents. In *CVPR*, 2021. 7

[34] Z. Lan, Y. Jiang, Y. Mu, C. Chen, and S. E. Li. Sept: Towards efficient scene representation learning for motion prediction. In *ICLR*, 2024. 3, 16

[35] K. Li, X. Li, Y. Wang, Y. He, Y. Wang, L. Wang, and Y. Qiao. Videomamba: State space model for efficient video understanding. In *ECCV*, 2024. 3, 4, 7

[36] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 1, 2, 15

[37] O. Lieber, B. Lenz, H. Bata, G. Cohen, J. Osin, I. Dalmedigos, E. Safahi, S. Meirom, Y. Belinkov, S. Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint*, 2024. 3

[38] L. Lin, X. Lin, T. Lin, L. Huang, R. Xiong, and Y. Wang. Eda: Evolving and distinct anchors for multimodal motion prediction. In *AAAI*, 2024. 1, 3

[39] M. Liu, H. Cheng, L. Chen, H. Broszio, J. Li, R. Zhao, M. Sester, and M. Y. Yang. Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. *arXiv preprint*, 2023. 7

[40] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In *ICLR*, 2022. 1, 3

[41] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 2, 15

[42] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[43] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *ICRA*, 2023. 2

[44] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. J. Weiss, B. Sapp, Z. Chen, and J. Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *ICLR*, 2022. 1, 2, 3, 7

[45] Z. Pang, D. Ramanan, M. Li, and Y.-X. Wang. Streaming motion forecasting for autonomous driving. In *IROS*, 2023. 14

[46] D. Park, J. Jeong, S.-H. Yoon, J. Jeong, and K.-J. Yoon. T4p: Test-time training of trajectory prediction via masked autoencoder and actor-specific token memory. In *CVPR*, 2024. 3

[47] D. Park, H. Ryu, Y. Yang, J. Cho, J. Kim, and K.-J. Yoon. Leveraging future relationship reasoning for vehicle trajectory prediction. In *ICLR*, 2023. 6, 15

[48] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *CVPR*, 2020. 2

[49] J. Philion, X. B. Peng, and S. Fidler. Trajeglish: Learning the language of driving scenarios. In *ICLR*, 2024. 3

[50] L. Rowe, M. Ethier, E.-H. Dykhne, and K. Czarnecki. Fjmp: Factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In *CVPR*, 2023. 2, 7

[51] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 7

[52] M. Schäfer, K. Zhao, M. Bühren, and A. Kummert. Context-aware scene prediction network (caspnet). In *IEEE ITSC*, 2022. 7

[53] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, K. S. Refaat, R. Al-Rfou, and B. Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *ICCV*, 2023. 3

[54] S. Shi, L. Jiang, D. Dai, and B. Schiele. Motion transformer with global intention localization and local movement refinement. *NeurIPS*, 2022. 1, 2, 3, 4, 6, 7

[55] S. Shi, L. Jiang, D. Dai, and B. Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE TPAMI*, 2024. 3, 4, 7

[56] J. T. Smith, A. Warrington, and S. Linderman. Simplified state space layers for sequence modeling. In *ICLR*, 2023. 3

[57] T. Su, X. Wang, and X. Yang. Qml for argoverse 2 motion forecasting challenge. *arXiv preprint*, 2022. 6

[58] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1, 3, 15

[59] X. Tang, M. Kan, S. Shan, Z. Ji, J. Bai, and X. Chen. Hpnet: Dynamic trajectory forecasting with historical prediction attention. In *CVPR*, 2024. 1, 3, 14, 15

[60] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *ICRA*, 2022. 2

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2, 3, 4

[62] J. Wang, T. Ye, Z. Gu, and J. Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *CVPR*, 2022. 15

[63] M. Wang, X. Ren, R. Jin, M. Li, X. Zhang, C. Yu, M. Wang, and W. Yang. Futurenet-lof: Joint trajectory prediction and lane occupancy field prediction with future context encoding. *arXiv preprint*, 2024. 16

[64] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, and W. Yang. Ganet: Goal area network for motion forecasting. In *ICRA*, 2023. 6

[65] X. Wang, T. Su, F. Da, and X. Yang. Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *CVPR*, 2023. 6

[66] Y. Wang, H. Zhou, Z. Zhang, C. Feng, H. Lin, C. Gao, Y. Tang, Z. Zhao, S. Zhang, J. Guo, et al. Tenet: Transformer encoding network for effective temporal flow on motion prediction. *arXiv preprint*, 2022. 6

[67] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS*, 2021. 1, 3, 5, 6, 7, 9

[68] M. Ye, T. Cao, and Q. Chen. Tpcn: Temporal point cloud networks for motion forecasting. In *CVPR*, 2021. 15

[69] W. Zeng, M. Liang, R. Liao, and R. Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *IROS*, 2021. 2, 15

[70] C. Zhang, H. Sun, C. Chen, and Y. Guo. Banet: Motion forecasting with boundary aware network. *arXiv preprint*, 2022. 6

[71] L. Zhang, P. Li, J. Chen, and S. Shen. Trajectory prediction with graph-based dual-scale context fusion. In *IROS*, 2022. 2, 15

[72] L. Zhang, P. Li, S. Liu, and S. Shen. Simpl: A simple and efficient multi-agent motion prediction baseline for autonomous driving. *IEEE RA-L*, 2024. 6, 10, 15

[73] Z. Zhang, A. Liniger, C. Sakaridis, F. Yu, and L. V. Gool. Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding. *NeurIPS*, 2023. 2, 6

[74] Z. Zhang, A. Liu, I. Reid, R. Hartley, B. Zhuang, and H. Tang. Motion mamba: Efficient and long sequence motion generation with hierarchical and bidirectional selective ssm. In *ECCV*, 2024. 3

[75] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, et al. Tnt: Target-driven trajectory prediction. In *CoRL*, 2021. 2, 15

[76] Y. Zhou, H. Shao, L. Wang, S. L. Waslander, H. Li, and Y. Liu. Smartrefine: A scenario-adaptive refinement framework for efficient motion prediction. In *CVPR*, 2024. 3, 6, 9

[77] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang. Query-centric trajectory prediction. In *CVPR*, 2023. 1, 2, 3, 6, 7, 9, 10

[78] Z. Zhou, Z. Wen, J. Wang, Y.-H. Li, and Y.-K. Huang. Qcnext: A next-generation framework for joint multi-agent trajectory prediction. *arXiv preprint*, 2023. 3, 7, 16

[79] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. H. Lu. Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, 2022. 2, 3, 15

[80] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *ICML*, 2024. 3, 4, 7

 https://doi.org/10.52202/079017-3383

Appendix

A Experimental settings

A.1 Baseline model

Based on the streaming nature of driving data, our baseline model employs a lightweight transformer-based architecture capable of historical prediction [45, 59]. This model predicts future trajectories at intervals of 3s, 4s, and 5s for the Argoverse 2 dataset, and at 1s, 1.5s, and 2s for the Argoverse 1 dataset. The final prediction is the ultimate outcome of these trajectories. Features from the encoder and decoder in the historical prediction phase are aggregated with those from the encoder and decoder in the subsequent prediction phase. The aggregation module comprises two cross-attention layers. The historical prediction component is not used for the nuScenes dataset due to the limited history steps. In ablation studies, we remove the historical prediction component from the baseline model to make the experiments more efficient.

A.2 Single-agent evaluation metrics

We employ established metrics to evaluate our models, including minimum Average Displacement Error ($minADE_k$), minimum Final Displacement Error ($minFDE_k$), Miss Rate (MR_k), and Brier minimum Final Displacement Error ($b\text{-}minFDE_k$). The $minADE_k$ metric calculates the L_2 distance between the ground-truth trajectory and the best K predicted trajectories, averaged over all future time steps. The $minFDE_k$ metric measures the discrepancy between the endpoints of the predicted trajectories and the ground truth. The MR_k metric represents the proportion of scenes where $minFDE_k$ exceeds 2 meters. To provide a more nuanced evaluation of uncertainty, $b\text{-}minFDE_k$ incorporates $(1-\pi)^2$ into the final displacement error, where π indicates the probability score assigned by the model to the best-predicted trajectory.

A.3 Multi-agent evaluation metrics

Following the official settings for the Argoverse 2 multi-agent test set, we use metrics like Average Minimum Final Displacement Error ($avgMinFDE$), Average Minimum Average Displacement Error ($avgMinADE$), and Actor Miss Rate ($actorMR$). The $avgMinFDE$ metric calculates the average of the lowest Final Displacement Errors (FDEs) for all scored actors in a scenario, reflecting prediction accuracy. Similarly, $avgMinADE$ is the average of the lowest Average Displacement Errors (ADEs) for all scored actors, showing overall movement accuracy. The $actorMR$ measures the proportion of actors missed across the evaluation set, as previously described.

A.4 More implementation details

In addition to the details in Section 4.1, we set the dropout rates at 0.2 for single-agent settings and 0.1 for multi-agent settings. We employ a cosine learning rate schedule with a warm-up phase of 10 epochs. For normalization, we use nn.LayerNorm, and for activation, we use nn.GELU. No data augmentation techniques are used. For details on the number of layers in each component, please refer to Table 10.

A.5 A precise formulation of the auxiliary losses

We use an MLP to decode state queries into a single future trajectory Y_f and calculate the loss with ground truth Y_{gt} to obtain $\mathcal{L}_{\text{ts}}$:

$$\mathcal{L}_{\text{ts}} = \text{SmoothL1}(Y_f, Y_{gt}). \tag{6}$$

We use MLPs to decode the future trajectories Y_f and probabilities P_f. So $\mathcal{L}_{\text{m}}$ is shown below:

$$\begin{aligned} Y_{best}, P_{best} &= \text{SelectBest}(Y_f, Y_{gt}), \\ \mathcal{L}_{\text{m}} &= \text{SmoothL1}(Y_{best}, Y_{gt}) + \text{CrossEntropy}(P_f, P_{best}). \end{aligned} \tag{7}$$

Table 10: Number of layers in each component.

Enc/Dec	Name	Num-AV1&AV2	Num-nuScenes
Enc	Agent Encoding Mamba	4	2
	Scene Context Transformer	5	4
Dec	State Consistency Module Attention	2	2
	State Consistency Module Mamba	2	2
	Mode Localization Module Attention	3	2
	Hybrid Coupling Module Attention	3	2
	Hybrid Coupling Module Mamba	2	2

B More experiments

B.1 Performance comparison on the Argoverse 1 dataset

To fully demonstrate the effectiveness of our DeMo, we compare it with several recent models on the Argoverse 1 [6] dataset. The results from the validation split are shown in Table 11, indicating that our model achieves impressive performance.

Table 11: Performance comparison on *Argoverse 1 validation set*.

Method	$minADE_6$	$minFDE_6$	MR_6
LTP [62]	0.78	1.07	-
LaneRCNN [69]	0.77	1.19	0.08
TPCN [68]	0.73	1.15	0.11
DenseTNT [26]	0.73	1.05	0.10
TNT [75]	0.73	1.29	0.09
mmTransformer [41]	0.71	1.15	0.11
LaneGCN [36]	0.71	1.08	-
SSL-Lanes [2]	0.70	1.01	0.09
PAGA [12]	0.69	1.02	-
DSP [71]	0.69	0.98	0.09
FRM [47]	0.68	0.99	-
ADAPT [1]	0.67	0.95	0.08
SIMPL [72]	0.66	0.95	0.08
HiVT [79]	0.66	0.96	0.09
R-Pred [9]	0.66	0.95	0.09
HPNet [59]	0.64	**0.87**	**0.07**
DeMo (Ours)	**0.59**	0.90	**0.07**

B.2 Performance on the Argoverse 2 leaderboard

We provide a performance comparison of the top methods on the Argoverse 2 leaderboard as of September 2024. The results for the single-agent setting are shown in Table 12, and the results for the multi-agent setting are shown in Table 13.

B.3 Results on Waymo open motion dataset

We provide results on WOMD [58] in Table 14 using the settings in UniTraj [16], as shown below. The results of other methods are also from UniTraj.

B.4 Model ensembling

In our approach, we use model ensembling, an essential technique to enhance the accuracy of final predictions. We train six sub-models with various random seeds and training epochs, resulting in 36

https://doi.org/10.52202/079017-3383

Table 12: Performance comparison of the single-agent setting on the Argoverse 2 dataset in the official leaderboard. Unreleased works are marked with the symbol "*".

Method	*b-minFDE*$_6$	Rank
LOF [63]	1.63	1
iDLab-SEPT++ (SEPT++) *	1.65	2
EACON (JMaC) *	1.67	3
PolarMotion_E (PolarMotion) *	1.71	4
DeMo (Ours)	1.73	5
iDLab-SEPT (SEPT) [34]	1.74	6
xPnC (X-MotionFormer) *	1.74	7
GACRND-XLAB (XPredFormer) *	1.76	8

Table 13: Performance comparison of the multi-agent setting on the Argoverse 2 dataset in the official leaderboard. Unreleased works are marked with the symbol "*".

Method	*avgBrierMinFDE*$_6$	Rank
EACON (JMaC) *	1.62	1
QCNet-AV2 (QCNeXt) [78]	1.65	2
Lite-QCNet *	1.67	3
LOF [63]	1.68	4
iDLab-SEPT [34]	1.80	5
DeMo (Ours)	1.93	6
FAW-Prediction *	1.93	7
berste (OGD_test2) *	1.95	8

Table 14: Results on WOMD.

Method	*minFDE*$_6$	*minADE*$_6$
MTR	1.78	0.78
Wayformer	1.46	0.65
AutoBot	1.65	0.73
DeMo (Ours)	1.59	0.75

predicted future trajectories for each agent. We then apply k-means clustering with six cluster centers to process these trajectories. For each cluster group, we compute the average trajectory within the group to determine the final trajectories. We present the results, both with and without the model ensembling technique, on the Argoverse 2 single-agent test set in Table 1.

C Mamba introduction

Mamba is inspired by a continuous system that maps a 1-D function or sequence $x(t) \in \mathbb{R}$ to $y(t) \in \mathbb{R}$, utilizing a hidden state $h(t) \in \mathbb{R}^N$. In this system, $\mathbf{A} \in \mathbb{R}^{N\times N}$ serves as the evolution parameter, while $\mathbf{B} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times N}$ act as the projection parameters.

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}h(t). \end{aligned} \tag{8}$$

Mamba represents the discrete version of a continuous system and includes a timescale parameter $\mathbf{\Delta}$ to convert the continuous parameters $\mathbf{A}$ and $\mathbf{B}$ into discrete counterparts $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$. The most commonly used method for this transformation is zero-order hold (ZOH). After discretizing $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$, the discretized form of Equation (8) using a step size of $\mathbf{\Delta}$ can be reformulated as:

$$
\begin{aligned}
\overline{\mathbf{A}} &= \exp(\mathbf{\Delta A}), \\
\overline{\mathbf{B}} &= (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta B}. \\
h_t &= \overline{\mathbf{A}} h_{t-1} + \overline{\mathbf{B}} x_t, \\
y_t &= \mathbf{C} h_t.
\end{aligned} \tag{9}
$$

Finally, the models compute the output through a global convolution that utilizes a structured convolutional kernel $\overline{\mathbf{K}} \in \mathbb{R}^M$, where M represents the length of the input sequence $\mathbf{x}$.

$$
\begin{aligned}
\overline{\mathbf{K}} &= (\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{A}\mathbf{B}}, \ldots, \mathbf{C}\overline{\mathbf{A}}^{M-1}\overline{\mathbf{B}}), \\
\mathbf{y} &= \mathbf{x} * \overline{\mathbf{K}},
\end{aligned} \tag{10}
$$

D More qualitative results

We provide more qualitative results of our DeMo in Figure 5.

E Failure cases

Although our DeMo has demonstrated exceptional performance on motion forecasting benchmarks, it is not without its failures. We analyze these typical cases and present qualitative results to give readers insight into the scenarios where our model might underperform. This analysis is intended to support future efforts in developing an algorithm that is both more robust and powerful, as illustrated in Figure 4. In the first row, the vehicle will turn into an alley, reflecting a kind of subjective driving behavior. However, the model predicts that the vehicle will just keep going straight. To improve predictions in cases like this, we could enhance how the model interacts with additional information about what the vehicle intends to do, such as adding visual cues, such as turn signals. In the second row, the agent needs to navigate through a complex intersection to reach one of the roads; however, the model fails to accurately predict this driving behavior. This inaccuracy may be caused by a lack of comprehensive understanding of the complex map topology and the unbalanced distribution of driving data, addressing the issue of data balance is necessary to solve this problem.

https://doi.org/10.52202/079017-3383

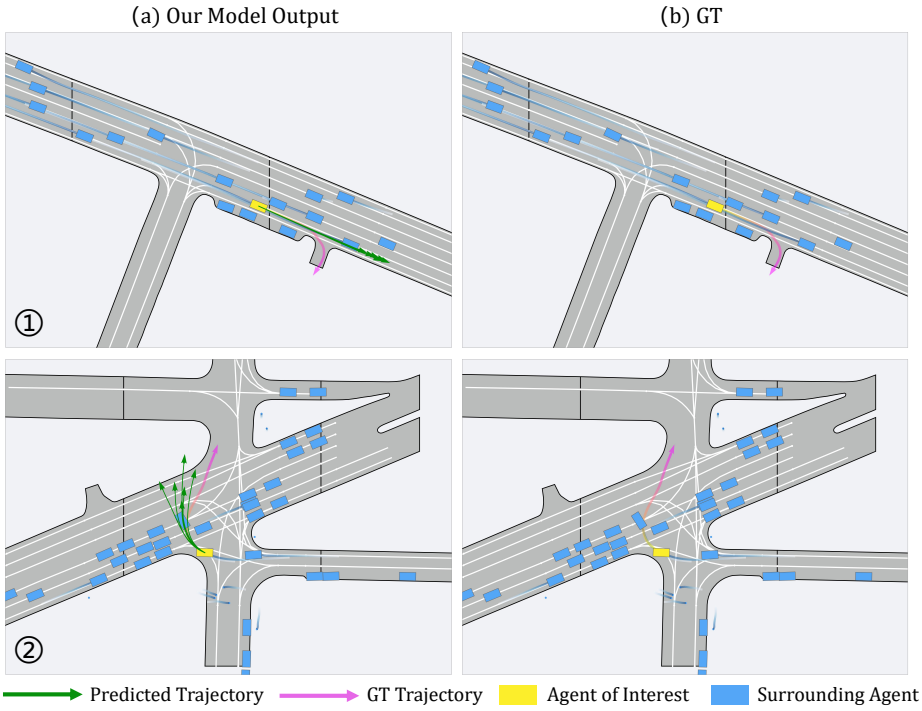

Figure 4: Failure cases.

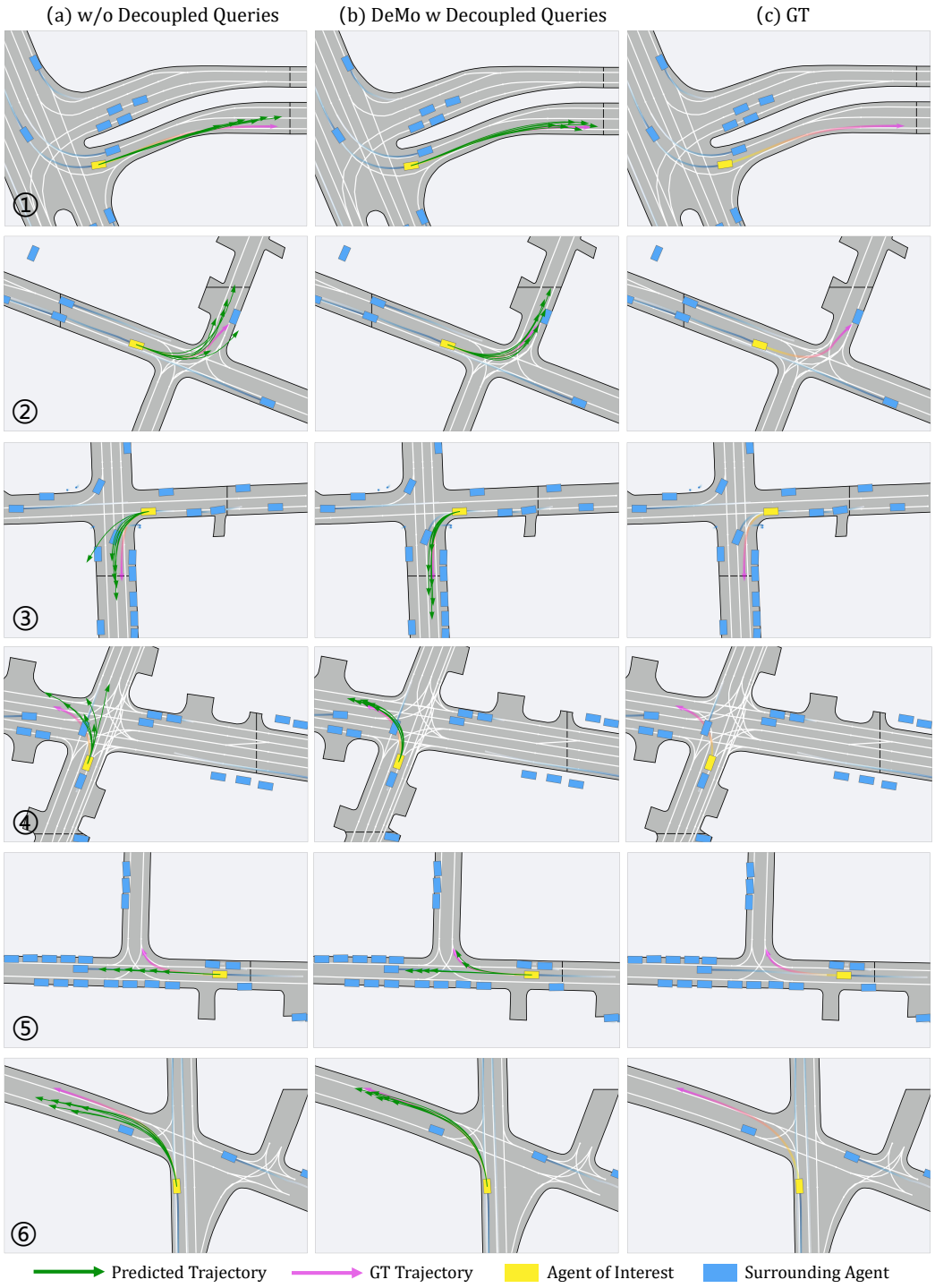

Figure 5: More qualitative results.

https://doi.org/10.52202/079017-3383

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/079017-3383

Answer: [Yes]

Justification: The details of the model and experiments are provided in the main paper and the appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our main idea is specifically designed for applications in real-world autonomous driving.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-3383

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Section 4.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

 https://doi.org/10.52202/079017-3383