Motion Forecasting Transformer with Global Intention Localization and Local Movement Refinement

Anonymous Author(s)
Affiliation
Address
email

Abstract

Predicting multimodal future behavior of traffic participants is essential for robotic vehicles to make safe decisions. Existing works explore to directly predict future trajectories based on latent features or utilize dense goal candidates to identify agent's destinations, where the former strategy converges slowly since all motion modes are derived from the same feature while the latter strategy has efficiency issue since its performance highly relies on the density of goal candidates. In this paper, we propose the Motion TRansformer (MTR) framework that models motion prediction as the joint optimization of global intention localization and local movement refinement. Instead of using goal candidates, MTR incorporates spatial intention priors by adopting a small set of learnable motion query pairs. Each motion query pair takes charge of trajectory prediction and refinement for a specific motion mode, which stabilizes the training process and facilitates better multimodal predictions. Experiments show that MTR achieves state-of-the-art performance on both the marginal and joint motion prediction challenges, ranking 1st on the leaderbaords of Waymo Open Motion Dataset. Code will be available.

1 Introduction

Motion forecasting is a fundamental task of modern autonomous driving systems. It has been receiving increasing attention in recent years [18, 39, 24, 47, 30] as it is crucial for robotic vehicles to understand driving scenes and make safe decisions. Motion forecasting requires to predict future behaviors of traffic participants by jointly considering the observed agent states and road maps, which is challenging due to inherently multimodal behaviors of the agent and complex scene environments.

To cover all potential future behaviors of the agent, existing approaches mainly fall into two different lines: the goal-based methods and the direct-regression methods. The goal-based methods [18, 51] adopt densely defined goal candidates to cover all possible destinations of the agent, predicting the probability of each goal candidate being a real destination and then completing the full trajectory for each selected goal candidate. Although these goal candidates alleviate the burden of model optimization by reducing trajectory uncertainty, their density largely affects the performance of the goal-based methods: fewer candidates will decrease the performance while more candidates will greatly increase computation and memory cost. Instead of using goal candidates to cover the agent's future motion, the direct-regression methods [30, 40] directly predict a set of trajectories based on the encoded agent feature, covering the agent's future behavior adaptively. Despite the flexibility in predicting a broad range of agent behaviors, they generally converge slowly as various motion modes are required to be regressed from the same agent feature without utilizing any spatial priors. They also tend to predict the most frequent modes of training data since these frequent modes dominate the optimization of the agent feature. In this paper, we present a unified framework, namely Motion TRansformer (MTR), which takes the best of both types of methods.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In our proposed MTR, we adopt a small set of novel motion query pairs to model motion prediction as the joint optimization of two tasks, *i.e.*, global intention localization and local movement refinement. Global intention localization aims to roughly identify the intention of the agents for achieving higher efficiency, while local movement refinement aims to adaptively refine the predicted trajectories of each intention for achieving better accuracy. Our approach not only stabilizes the training process without depending on dense goal candidates but also enables flexible and adaptive trajectory prediction by enabling local movement refinement for each motion mode.

Specifically, each motion query pair consists of two types of queries, i.e., a static intention query and a dynamic searching query, for facilitating the above two tasks. The static intention queries are introduced for global intention localization, where we formulate these queries based on a small set of spatially distributed intention points. Each static intention query is considered as the learnable positional embedding of an intention point for generating the trajectory of a specific motion mode, which not only stabilizes the training process by explicitly utilizing different queries for different modes, but also eliminates the dependency on dense goal candidates by requiring each query to take charge of a large region. The dynamic searching queries are utilized for local movement refinement, where they are also initialized as the learnable embeddings of the intention points but are responsible for retrieving fine-grained local features around each intention point. For this purpose, the dynamic searching queries are dynamically updated according to the predicted trajectories, which can adaptively gather latest trajectory features from a deformable local region for iterative motion refinement. These two types of queries complement each other and have been experimentally demonstrated their great effectiveness in predicting multimodal future trajectories. Besides that, we propose a dense future prediction module. Existing works generally focus on modeling the agent interactions over past trajectories while ignoring their interactions in the future. To compensate for such information, we adopt a simple auxiliary regression head to densely predict future trajectory and velocity for each agent, which are encoded as additional context features to enable future motion interactions between our interested agent and other agents. The experiments show that this simple auxiliary task works well and remarkably improves the performance of multimodal motion prediction.

Our contributions are three-fold: (1) We propose the concept of motion query pair that adopts two types of queries to model motion prediction as the joint optimization of global intention localization and local movement refinement. It not only stabilizes the training process with mode-specific motion query pairs, but also enables adaptive motion refinement by iteratively gathering fine-grained trajectory features. (2) We present an auxiliary dense future prediction module to facilitate the future interactions of the agents. It enables our framework to predict more scene-compliant future trajectories for the interacting agents. (3) By adopting the above techniques, we propose a novel MTR framework that explores transformer encoder-decoder structure for multimodal motion prediction. Our approach achieves state-of-the-art performance on both the marginal and joint motion prediction benchmarks of Waymo Open Motion Dataset [13], outperforming previous best ensemble-free approaches with +8.48% mAP gains for marginal motion prediction and $+7.98\%$ mAP gains for joint motion prediction. As of 19 May 2022, our approach ranks 1st on both the marginal and joint motion prediction leaderboards of Waymo Open Motion Dataset.

2 Related Work

Motion Prediction for Autonomous Driving. Recently, motion prediction for road participants has been extensively studied due to the growing interest in autonomous driving. It typically takes road map and agent history states as input to predict future trajectories. To encode such scene context features, early works [31, 26, 5, 11, 50, 3, 8] typically rasterize roads and agent trajectories into an image, which is then encoded by convolutional neural networks (CNNs). In contrast, LaneGCN [22] builds a lane graph to capture map topology. VectorNet [15] is widely adopted by recent works [18, 37, 30, 40] due to its efficiency and scalability for encoding scene context, where both road maps and agent trajectories are represented as sparse polylines. We also adopt this vector representation, but instead of building global graph of polylines as in [15], we propose to adopt transformer encoders on the local connected graph, which not only better maintains locality structure of input elements but also is more memory-efficient to enable larger map encoding for long-term motion prediction.

Given the encoded scene context features, existing works explore various strategies to model multi-modal future motion. Early works [1, 19, 34, 38, 35] propose to generate a set of trajectory samples to approximate the output distribution. Some other works [9, 20, 28, 32, 36] parameterize multi-modal predictions with Gaussian Mixture Models (GMMs) to generate compact output distribution.

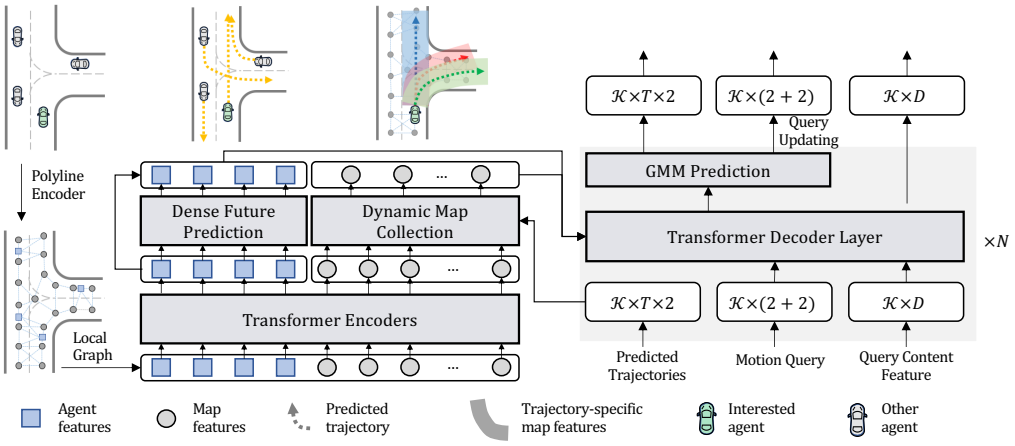


Figure 1: The architecture of our proposed MTR framework for multimodal motion prediction.

HOME [17] and its variant[16] predict a heatmap first and then generate trajectories with proper sampling process. IntentNet [7] considers intention prediction as a classification with 8 high level actions, while [24] proposes a region-based training strategy to recognize different region-based actions. Goal-based methods [51, 35, 14, 25] are another kinds of models where they first estimate several goal points of the agents and then complete full trajectory for each goal.

Recently, the large-scale Waymo Open Motion Dataset [13] is proposed for long-term motion prediction (8 seconds into the future) in realistic driving environment. To address this challenge, DenseTNT [18] adopts a goal-based strategy to classify endpoint of trajectory from densely sampled goal points. Other works directly predict the future trajectories based on the encoded agent features [30] or latent anchor embedding [40]. However, the goal-based strategy has the efficiency concern for long-term motion prediction due to a large number of goal candidates, while the direct-regression strategy converges slowly as the predictions of various motion modes are regressed from the same agent feature. In contrast, our approach adopts a small set of learnable motion query pairs, which not only eliminate the large number of goal candidates but also alleviate the optimization burden by utilizing mode-specific motion query pairs for predicting different motion modes.

Transformer. Transformer [41] has been widely applied in natural language processing [10, 2] and computer vision [12, 43, 4, 42, 48]. Our approach is inspired by DETR [4] and its follow-up works [53, 27, 23, 49], where object query is proposed for end to end object detection. Motivated by its great success in object detection, we introduce a novel concept of motion query pair to model multimodal motion prediction, where each motion query pair takes charge of predicting a specific motion mode and also enables iterative motion refinement by combining with transformer decoders.

3 Motion TRansformer (MTR)

We propose Motion TRansformer (MTR) framework for multimodal future prediction, which adopts a novel transformer encoder-decoder structure with iterative motion refinement for predicting multimodal future trajectories. The overall structure of our proposed MTR is illustrated in Figure 1. In Sec. 3.1, we first introduce our encoder network for scene context modeling with transformers. Next, in Sec. 3.2, we present transformer decoders with a novel concept of motion query pair for predicting multimodal trajectories. Finally, in Sec. 3.3, we introduce the optimization process of our framework.

3.1 Transformer Encoder for Scene Context Modeling

The future behaviors of the agents highly depend on both agents' interactions and road environment. To encode such scene context, state-of-the-art approaches have explored various strategies by building global interacting graph including both maps and agents [15, 18] or summarizing map features to agent-wise features [30, 40]. We argue that locality structure of the input elements is important for encoding scene context information, especially for the map elements. Hence, we propose a simple transformer encoder network with local self-attention to better maintain such structure information.

Input representation. We follow the popular vectorized representation [15] to organize both agent's history state and road map as polylines. For predicting future trajectory of a agent, we adopt the agent-centric strategy as in [51, 18, 40] that normalizes all inputs to the coordinate system centered

at this agent. Then, a simple polyline encoder is firstly adopted to encode each polyline as an input token feature for the following transformer encoders. Specifically, we denote the history motion state of N_a agents as $A_{\text{in}} \in \mathbb{N}^{N_a \times t \times C_a}$, where t is the number of history frames, C_a is the number of state information (*e.g.*, location, heading angle and velocity), and we pad zeros for trajectories that have less than t frames. The road map is denoted as $M_{\text{in}} \in \mathbb{R}^{N_m \times n \times C_m}$, where N_m is the number of map polylines, n is the number of points in each polyline and C_m is the number of attributes of each point (*e.g.*, location and road type). Both of them are encoded by a PointNet-like [33] polyline encoder as:

$$A = \phi\left(\text{MLP}(A_{\text{in}})\right), \quad M = \phi\left(\text{MLP}(M_{\text{in}})\right), \tag{1}$$

where $\text{MLP}(\cdot)$ is a multilayer perceptron network, and ϕ is max-pooling to summarize each polyline features as agent features $A \in \mathbb{R}^{N_a \times D}$ and map features $M \in \mathbb{R}^{N_m \times D}$ with feature dimension D.

Scene context encoding with local transformer encoders. The local structure in road maps and agents is important for motion prediction. For example, the relation of two parallel lanes is important for modelling the motion of changing lanes, but adopting attention on global connected graph equally considers relation of all lanes and needs to learn such local information from their position encodings. In contrast, we introduce such prior knowledge to context encoder by adopting transformer encoders with local self-attention, which better maintains their locality structure and are also more memory-efficient. Specifically, the attention module of transformer encoder can be formulated as:

$$G = \text{MultiHeadAttn}(\text{query}{=}G + \text{PE}_G,\ \text{key}{=}\kappa(G) + \text{PE}_{\kappa(G)},\ \text{value}{=}\kappa(G)), \tag{2}$$

where $\text{MultiHeadAttn}(\cdot,\cdot,\cdot)$ is the multi-head attention layer [41], $G = [A, M] \in \mathbb{N}^{(N_a+N_m)\times D}$ concatenating the features of agents and map, and $\kappa(\cdot)$ denotes k-nearest neighbor algorithm to find k closest polylines for each query polyline. PE denotes sinusoidal position encoding of input tokens, where we utilize the latest position for each agent and utilize polyline center for each map polyline.

Thanks to the local self-attention on input elements, our framework can encode a much larger area of road map, which provides more context features for the following decoder network. The encoder network generates both agent features $A \in \mathbb{R}^{N_a \times D}$ and map features $M \in \mathbb{R}^{N_m \times D}$, which are considered as the inputs to provide scene context features in the following decoder network.

Dense future prediction for future interactions. Interactions with other agents heavily affect behaviors of our interested agent. However, existing works generally focus on learning such interactions over past trajectories while ignoring their future interactions. Therefore, considering that the encoded features A have already learned rich context information of all agents, we propose to densely predict both future trajectories and velocities of all agents by adopting a simple regression head on A:

$$S_{1:T} = \text{MLP}(A), \tag{3}$$

where $S_i \in \mathbb{R}^{N_a \times 4}$ includes future position and velocity of each agent at time step i, and T is the number of future frames to be predicted. The predicted future trajectories are encoded by adopting the same polyline encoder as Eq. (1) to encode the agents' future states as features $A_f \in \mathbb{R}^{N_a \times D}$, which are then utilized to enhance the above features A by using a feature concatenation and three MLP layers as $A = \text{MLP}([A, A_f])$. This auxiliary task provides additional future context information to the decoder network, and enables future interactions of the agents. The experiments in Table 3 demonstrates that this simple and light-weight auxiliary task can effectively improve the performance of multimodal motion prediction.

3.2 Transformer Decoder with Motion Query Pair

Given the scene context features, a transformer-based decoder network is adopted for multimodal motion prediction, where we propose a novel concept of *motion query pair* to model motion prediction as the joint optimization of global intention localization and local movement refinement. Each motion query pair contains two types of queries, *i.e.*, static intention query and dynamic searching query, for conducting global intention localization and local movement refinement respectively. As shown in Figure 2, our decoder network contains several stacked transformer decoder layers for iteratively refining the predicted trajectories with motion query pairs. Next, we illustrate the detailed structure.

Global intention localization. The global intention localization aims to localize the potential motion intentions of the agents in an efficient and effective manner. Hence, we propose *static intention query* to narrow down the uncertainty of future trajectory by utilizing different intention queries for different motion modes. Specifically, we generate $\mathcal{K}$ representative intention points $I \in \mathbb{R}^{\mathcal{K} \times 2}$ by utilizing k-means clustering algorithm on the endpoints of ground-truth (GT) trajectories,

where each intention point represents an implicit motion mode that considers both motion direction and velocity. We model each static intention query as the learnable positional embedding of the intention point as:

$$Q_I = \text{MLP}\left(\text{PE}(I)\right), \tag{4}$$

where PE$(\cdot)$ is the sinusoidal position encoding, and $Q_I \in \mathbb{R}^{\mathcal{K}\times D}$. Notably, each intention query takes charge of predicting trajectories for a specific motion mode, which stabilizes the training process and facilitates predicting multimodal trajectories since each motion mode has their own learnable embedding. Thanks to their learnable and adaptive properties, we only need a small number of queries (*e.g.*, 64 queries in our setting) for efficient intention localization, instead of using densely-placed goal candidates [51, 18] to cover the destinations of the agents.

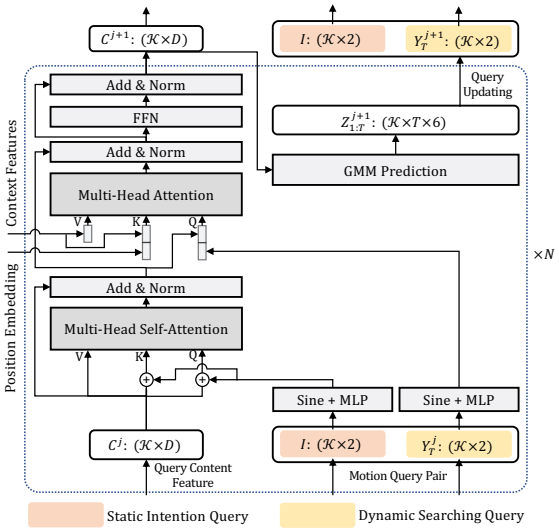


Figure 2: The network structure of our decoder network with motion query pair.

Local movement refinement. The local movement refinement aims to complement with global intention localization by iteratively gathering fine-grained trajectory features for refining the predicted trajectories. Therefore, we propose *dynamic searching query* to adaptively probe trajectory features for each motion mode. Each dynamic searching query is also the position embedding of a spatial point, which is initialized with the same position of its corresponding intention point but will be dynamically updated according to the predicted trajectory in each decoder layer. Specifically, given the predicted future trajectories $Y_{1:T}^j = \{Y_i^j \in \mathbb{R}^{\mathcal{K}\times 2} \mid i = 1, \cdots, T\}$ in j-th decoder layer, the dynamic searching query of $(j+1)$-th decoder layer is updated as follows:

$$Q_S^{j+1} = \text{MLP}\left(\text{PE}(Y_T^j)\right). \tag{5}$$

As shown in Figure 3, for each motion query pair, we propose a *dynamic map collection* module to extract fine-grained trajectory features by querying map features from a trajectory-aligned local region, which is implemented by collecting L polylines whose centers are closest to the predicted trajectory. As the agent's behavior largely depends on road maps, this local movement refinement strategy can continually provide updated fine-grained context information for iterative motion refinement.

Attention module with motion query pair. In each decoder layer, static intention query is utilized to propagate information among different motion intentions, while dynamic searching query is utilized to aggregate trajectory-specific features from scene context features. Specifically, we utilize static intention query as the query position embedding of self-attention module to enable information propagation among different queries, which is formulated as follows:

$$C_{\text{sa}}^j = \text{MultiHeadAttn}(\text{query}{=}C^{j-1} + Q_I,\ \text{key}{=}C^{j-1} + Q_I,\ \text{value}{=}Q_I), \tag{6}$$

where $C^{j-1} \in \mathbb{R}^{\mathcal{K}\times D}$ is query content features from $(j-1)$-th decoder layer, and $C_{\text{sa}}^j \in \mathbb{R}^{\mathcal{K}\times D}$ is the updated query content features. Next, we utilize dynamic searching query as query position embedding of cross attention module to probe trajectory-specific features from the outputs of encoder network. Inspired by [27, 23], we concatenate content features and position embedding for both query and key to decouple their contributions to the attention weights. Two cross-attention modules are adopted separately for aggregating features from both agent features A and map features M as:

$$\begin{aligned} C_A^j &= \text{MultiHeadAttn}(\text{query}{=}[C_{\text{sa}}^j, Q_S^j],\ \text{key}{=}[A, \text{PE}_A],\ \text{value}{=}A), \\ C_M^j &= \text{MultiHeadAttn}(\text{query}{=}[C_{\text{sa}}^j, Q_S^j],\ \text{key}{=}[\alpha(M), \text{PE}_{\alpha(M)}],\ \text{value}{=}\alpha(M)), \\ C^j &= \text{MLP}([C_A^j, C_M^j]) \end{aligned} \tag{7}$$

where $[\cdot,\cdot]$ indicates feature concatenation, $\alpha(M)$ is the aforementioned dynamic map collection module to collect L trajectory-aligned map features for motion refinement. Note that for simplicity, in Eq. (6) and (7), we omit the residual connection and feed-forward network in transformer layer [41].

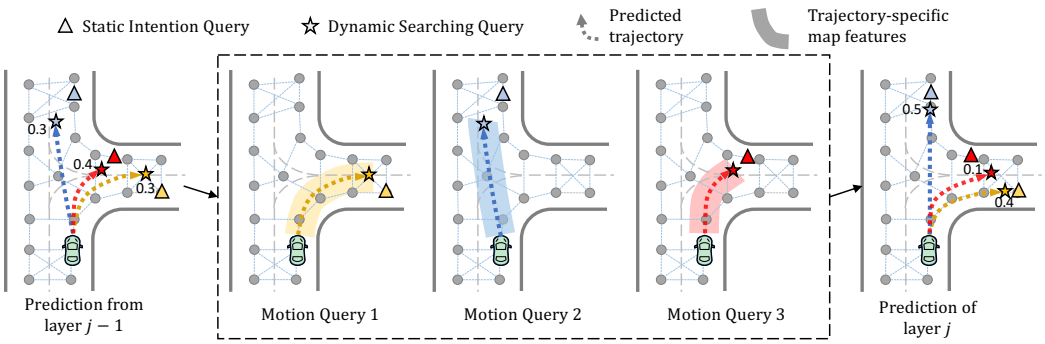


Figure 3: The illustration of dynamic map collection module for iterative motion refinement.

Finally, $C^j \in \mathbb{R}^{\mathcal{K} \times D}$ is the updated query content features for each motion query pair in j-th layer.

Multimodal motion prediction with Gaussian Mixture Model. For each decoder layer, we append a prediction head to C^j for generating future trajectories. As the behaviors of the agents are highly multimodal, we follow [9, 40] to represent the distribution of predicted trajectories with Gaussian Mixture Model (GMM) at each time step. Specifically, for each future time step $i \in \{1, \cdots, T\}$, we predict the probability p and parameters $(\mu_x, \mu_y, \sigma_x, \sigma_y, \rho)$ of each Gaussian component as follows

$$Z^j_{1:T} = \text{MLP}(C^j), \tag{8}$$

where $Z^j_i \in \mathbb{R}^{\mathcal{K} \times 6}$ includes $\mathcal{K}$ Gaussian components $\mathcal{N}_{1:\mathcal{K}}(\mu_x, \sigma_x; \mu_y, \sigma_y; \rho)$ with probability distribution $p_{1:\mathcal{K}}$. The predicted distribution of agent's position at time step i can be formulated as:

$$P^j_i(o) = \sum_{k=1}^{\mathcal{K}} p_k \cdot \mathcal{N}_k(o_x - \mu_x, \sigma_x; o_y - \mu_y, \sigma_y; \rho). \tag{9}$$

where $P^j_i(o)$ is the occurrence probability of the agent at spatial position $o \in \mathbb{R}^2$. The predicted trajectories $Y^j_{1:T}$ can be generated by simply extracting the predicted centers of Gaussian components.

3.3 Training Losses

Our model is trained end-to-end with two training losses. The first auxiliary regression loss L_a is to optimize the outputs of Eq. (3) by adopting $L1$ regression loss. For the second Gaussian regression loss L_G, we adopt negative log-likelihood loss according to Eq. 9 to maximum the likelihood of ground-truth trajectory. Inspired by [9, 40], we adopt a hard-assignment strategy that selects one closest motion query pair as positive Gaussian component for optimization, where the selection is implemented by calculating the distance between each intention point and the endpoint of GT trajectory. The Gaussian regression loss is adopted in each decoder layer, and the final loss is the sum of the auxiliary regression loss and all the Gaussian regression loss with equal loss weights.

4 Experiments

4.1 Experimental Setup

Dataset and metrics. We evaluate our approach on the large-scale Waymo Open Motion Dataset (WOMD) [13], which mines interesting interactions from real-world traffic scenes and is currently the most diverse interactive motion dataset. There are two tasks in WOMD with separate evaluation metrics: (1) The *marginal motion prediction challenge* that independently evaluates the predicted motion of each agent (up to 8 agents per scene). (2) The *joint motion prediction challenge* that needs to predict the joint future positions of 2 interacting agents for evaluation. Both of them provide 1 second of history data and aim to predict 6 marginal or joint trajectories of the agents for 8 seconds into the future. There are totally $487k$ training scenes, and about $44k$ validation scenes and $44k$ testing scenes for each challenge. We utilize the official evaluation tool of WOMD to calculate the evaluation metrics, including minADE, minFDE, miss rate and mAP. The mAP and miss rate are the most important evaluation metrics as in the official leaderboard[45, 44].

Implementation details. For the context encoding, we stack 6 transformer encoders. The road map (*e.g.*, lanes, crosswalk) is represented as multiple polylines, where each polyline contains up to 20

Table 1: Performance comparison of marginal motion prediction on the validation and test set of Waymo Open Motion Dataset. †: The results are shown in *italic* for reference since their performance is achieved with model ensemble techniques. We only evaluate our default setting MTR on the test set by submitting to official test server due to the limitation of submission times of WOMD.

	Method	Reference	minADE ↓	minFDE ↓	Miss Rate ↓	**mAP** ↑
Test	MotionCNN [21]	CVPRw 2021	0.7400	1.4936	0.2091	0.2136
	ReCoAt [52]	CVPRw 2021	0.7703	1.6668	0.2437	0.2711
	DenseTNT [18]	ICCV 2021	1.0387	1.5514	0.1573	0.3281
	SceneTransformer [30]	ICLR 2022	0.6117	**1.2116**	0.1564	0.2788
	†MultiPath++ [40]	ICRA 2022	*0.5557*	*1.1577*	*0.1340*	*0.4092*
	MTR (Ours)	-	**0.6050**	1.2207	**0.1351**	**0.4129**
Val	MTR (Ours)	-	0.6046	1.2251	0.1366	0.4164
	MTR-e2e (Ours)	-	0.5160	1.0404	0.1234	0.3245
	†MTR-ens (Ours)	-	*0.5686*	*1.1534*	*0.1240*	*0.4323*

Table 2: Performance comparison of joint motion prediction on the interactive validation and test set of Waymo Open Motion Dataset.

	Method	Reference	minADE ↓	minFDE ↓	Miss Rate ↓	**mAP** ↑
Test	Waymo LSTM baseline [13]	ICCV 2021	1.9056	5.0278	0.7750	0.0524
	HeatIRm4 [29]	CVPRw 2021	1.4197	3.2595	0.7224	0.0844
	AIR^2 [46]	CVPRw 2021	1.3165	2.7138	0.6230	0.0963
	SceneTransformer [30]	ICLR 2022	0.9774	2.1892	0.4942	0.1192
	M2I [37]	CVPR 2022	1.3506	2.8325	0.5538	0.1239
	MTR (Ours)	-	**0.9181**	**2.0633**	**0.4411**	**0.2037**
Val	MTR (Ours)	-	0.9132	2.0536	0.4372	0.1992

map points (about $10m$ in WOMD). We select $N_m = 768$ nearest map polylines around the interested agent for map encoding. The number of neighbors in encoder's local self-attention is set to 16. The hidden feature dimension is set as $D = 256$. For the decoder modules, we stack 6 decoder layers. Given the predicted trajectories in each layer, L is set to 128 to collect the closest map polylines from context encoder for motion refinement. By default, we utilize 64 motion query pairs where their intention points are generated by conducting k-means clustering algorithm on the training set. To generate 6 future trajectories for evaluation, we use non-maximum suppression (NMS) to select top 6 predictions from 64 predicted trajectories by calculating the distances between their endpoints, and the distance threshold is set as $2.5m$. We provide more implementation details in Appendix.

Training details. Our model is trained in an end-to-end manner by AdamW optimizer with a learning rate of 0.0001 and batch size of 80 scenes. We train the model for 60 epochs with 8 GPUs (NVDIA RTX 8000), and the learning rate is decayed by a factor of 0.5 every 5 epochs from epoch 30. The weight decay is set as 0.01 and we do not use any data augmentation.

MTR-e2e for end-to-end motion prediction. We also propose an end-to-end variant of MTR, called MTR-e2e, where only 6 motion query pair are adopted so as to remove NMS post processing. In the training process, instead of using static intention points for target assignment as in MTR, MTR-e2e selects positive mixture component by calculating the distances between its 6 predicted trajectories and the GT trajectory, since 6 intention points are too sparse to well cover all potential future motions.

4.2 Main Results

Performance comparison for marginal motion prediction. Table 1 shows our main results for marginal motion prediction on WOMD [13], our proposed MTR outperforms previous ensemble-free approaches [18, 30] with remarkable margins, $i.e.$, increasing the mAP by $+8.48\%$ and decreasing the miss rate from 15.64% to 13.51%. In particular, our single-model results of MTR also achieve better mAP than the latest work MultiPath++ [40], where MultiPath++ uses a novel model ensemble strategy that greatly improves its the performance.

Table 1 also shows the comparison of MTR variants. MTR-e2e achieves much better performance on minADE and minFDE due to the removing of NMS post-processing, while MTR achieves much better mAP since it learns explicit meaning of each motion query pair that can produce more confident intention predictions. We also propose a simple model ensemble strategy to merge the predictions of MTR and MTR-e2e and utilize NMS to remove redundant predictions (denoted as MTR-ens), and it takes the best of these two models and achieves much better mAP. Using a more advanced model

Table 3: Effects of different components in MTR framework. All models share the same encoder network. "latent learnable embedding" indicates using 6 latent learnable embeddings as queries of decoder network, and "iterative refinement" indicates using 6 stacked decoders for motion refinement.

Global Intention Localization	Iterative Refinement	Local Movement Refinement	Dense Future Prediction	minADE ↓	minFDE ↓	Miss Rate ↓	**mAP** ↑
Latent learnable embedding	×	×	×	0.6829	1.4841	0.2128	0.2633
Static intention query	×	×	×	0.7036	1.4651	0.1845	0.3059
Static intention query	✓	×	×	0.6919	1.4217	0.1776	0.3171
Static intention query	✓	✓	×	0.6833	1.4059	0.1756	0.3234
Static intention query	✓	×	✓	0.6735	1.3847	0.1706	0.3284
Static intention query	✓	✓	✓	**0.6697**	**1.3712**	**0.1668**	**0.3437**

ensemble technique (*e.g.*, the strategy proposed by MultiPath++) may further boost our performance, which is not the focus of this paper and is considered as one of the future works.

Performance comparison for joint motion prediction. To evaluate our approach for joint motion prediction, we combine the marginal predictions of two interacting agents into joint prediction as in [6, 13, 37], where we take the top 6 joint predictions from 36 combinations of these two agents. The confidence of each combination is the product of marginal probabilities. As shown in Table 2, our approach significantly outperforms state-of-the-arts [30, 37] with large margins on all metrics. Particularly, our MTR boosts the mAP from 12.39% to 20.37% and decreases the miss rate from 49.42% to 44.11%. The remarkable performance gains demonstrate the effectiveness of MTR for predicting scene-consistent future trajectories, and also show the effectiveness of our dense future prediction module to enable the future interactions of the agents. Besides that, we also provide some qualitative results in Figure 5 to show our predictions in complicated interacting scenarios.

As of May 19, 2022, our method ranks 1st on the motion prediction leaderboard of WOMD for both two challenges [45, 44]. The significant improvements manifest the effectiveness of MTR framework.

4.3 Ablation Study

We study the effectiveness of each component in MTR. For efficiently conducting ablation experiments, we uniformly sampled 20% frames (about $97k$ scenes) from the WOMD training set according to their default order, and we empirically find that it has similar distribution with the full training set. All models are evaluated with marginal motion prediction metric on the validation set of WOMD.

Effects of the decoder network of MTR. We study the effectiveness of each component in our decoder network, including global intention localization, iterative refinement and local movement refinement process. As shown in Table 3, all components contributes remarkably to the final performance in terms of the official ranking metric mAP. Especially, our proposed static intention queries with intention points achieves much better mAP (*i.e.*, $+4.26\%$) than the latent learnable embeddings thanks to its mode-specific querying strategy, and both the iterative refinement and local movement refinement strategy continually improve the mAP from 30.59% to 32.34% by aggregating more fine-grained trajectory features for motion refinement.

Effects of dense future prediction. Table 3 shows that our proposed dense future prediction module significantly improves the quality of predicted trajectories (*e.g.*, $+1.78\%$ mAP), which verifies that future interactions of the agents are important for motion prediction and our proposed strategy can learn such interactions to predict more reliable trajectories.

Effects of local attention for context encoding. Table 4 shows that by taking the same number of input map polylines as input, our proposed local self-attention in transformer encoders achieves better performance than global attention (*i.e.*, $+0.77\%$ mAP for 256 polylines and $+1.12\%$ mAP for 512 polylines), which verifies that the local structure of input elements is important for motion prediction and introducing such prior knowledge with local attention can benefit the performance of motion prediction. More importantly, transformer encoders with local attention are more memory-efficient and the performance keeps growing when improving the number of input map polylines from 256 to 1,024, while global attention will run out of memory due to its quadratic complexity.

Effects of the number of motion query pairs with different training strategies. As mentioned before, during training process, MTR and MTR-e2e adopt two different strategies for assigning positive mixture component, where MTR depends on static intention points (denoted as α) while MTR-e2e utilizes predicted trajectories (denoted as β). Figure 4 investigates the effects of the number of motion query pairs under these two strategies, where we have the following observations: (1)

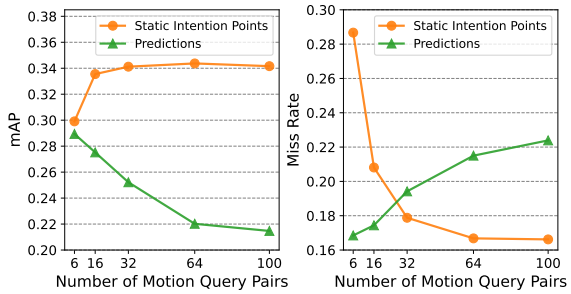
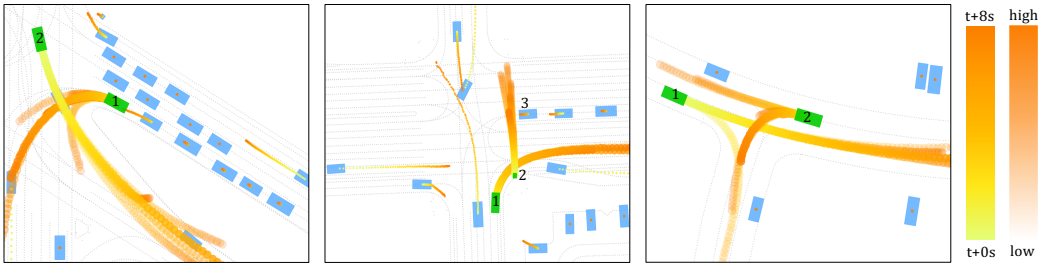


Figure 4: MTR framework with different number of motion query pairs, and two different colored lines demonstrate different strategies for selecting the positive mixture component during training process.

Table 4: Effects of local self-attention in transformer encoders. "#polyline" is the number of input map polylines used for context encoding, and a large number of polylines indicate that there is a larger map context around the interested agent. "OOM" indicates running out of memory.

Attention	#Polyline	minADE ↓	minFDE ↓	MR ↓	**mAP** ↑
Global	256	0.683	1.4031	0.1717	0.3295
Global	512	0.6783	1.4018	0.1716	0.3280
Global	768	OOM	OOM	OOM	OOM
Local	256	0.6724	1.3835	0.1683	0.3372
Local	512	0.6707	1.3749	0.1670	0.3392
Local	768	**0.6697**	**1.3712**	0.1668	0.3437
Local	1024	0.6757	1.3782	**0.1663**	**0.3452**



(a) **V2** is passing the intersection to turn left with high speed. Our model predicts multimodal behaviors for **V1:** turn left or make a U-turn. In any case, **V1** is predicted to yield for **V2**.

(b) **P2** is passing the road through the crosswalk while **V1** is on the right-turn lane to turn right. Both **V1** and **V3** are predicted to yield for **P2**.

(c) Our model predicts multimodal behaviors for **V1**: go straight and turn right, since it still has a distance to the intersection. **V2** is predicted to yield for **V1** when turning left, since **V1** is moving fast towards the intersection.

Figure 5: Qualitative results of MTR framework on WOMD. There are two interested agents in each scene (green rectangle), where our model predicts 6 multimodal future trajectories for each of them. For other agents (blue rectangle), a single trajectory is predicted by dense future prediction module. We use gradient color to visualize the trajectory waypoints at different future time step, and trajectory confidence is visualized by setting different transparent. Abbreviation: Vehicle (**V**), Pedestrian (**P**).

When increasing the number of motion query pairs, strategy α achieves much better mAP and miss rate than strategy β. Because intention query points can ensure more stable training process since each intention query points is responsible to a specific motion mode. In contrast, strategy β depends on unstable predictions and the positive component may randomly switch among all components, so a large number of motion query pairs are hard to be optimized with strategy β. (2) The explicit meaning of each intention query points also illustrates the reason that strategy α consistently achieves much better mAP than strategy β, since it can predict trajectories with more confident scores that benefit mAP. (3) From another side, when decreasing the number of motion query pairs, the miss rate of strategy α greatly increases, since a limit number of intention query points can not well cover all potential motions of agents. Conversely, strategy β works well for a small number of motion query pairs since its queries are not in charge of specific region and can globally adapt to any region.

5 Conclusion

In this paper, we present MTR, a novel framework for multimodal motion prediction based on transformers. The motion query pair is defined to model motion prediction as the joint optimization of global intention localization and local movement refinement. The global intention localization adopts a small set of learnable static intention queries to efficiently capture motion intentions of the agents, while the local movement refinement conducts iterative motion refinement by continually probing fine-grained trajectory features. We evaluate our approach on the large-scale Waymo Open Motion Dataset, and the experiments on both marginal and joint motion prediction challenges show that our approach achieves state-of-the-art performance.

Limitations. The default setting of our approach (MTR) adopts the rule-based NMS as a post processing to generate 6 future trajectories, which results in suboptimal predictions, especially for the minADE and minFDE metrics. We plan to develop a learning-based module to produce 6 future trajectories from our full multimodal predictions (*e.g.*, 64 modes) in a more robust manner.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[3] Yuriy Biktairov, Maxim Stebelev, Irina Rudenko, Oleh Shliazhko, and Boris Yangel. Prank: motion prediction based on ranking. *Advances in Neural Information Processing Systems*, 33:2553–2563, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

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

[6] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *European Conference on Computer Vision*, pages 624–641. Springer, 2020.

[7] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, 2018.

[8] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[9] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, 2019.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[13] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[14] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

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

[16] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. *arXiv preprint arXiv:2109.01827*, 2021.

[17] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Home: Heatmap output for future motion estimation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

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

[19] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

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

[21] Stepan Konev, Kirill Brodt, and Artsiom Sanakoyeu. Motioncnn: A strong baseline for motion prediction in autonomous driving. *Workshop on Autonomous Driving, CVPR*, 2021.

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

[23] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022.

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

[25] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, 2020.

[26] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[27] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021.

[28] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[29] Xiaoyu Mo, Zhiyu Huang, and Chen Lv. Multi-modal interactive agent trajectory prediction using heterogeneous edge-enhanced graph attention network. *Workshop on Autonomous Driving, CVPR*, 2021.

[30] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022.

[31] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *European Conference on Computer Vision*. Springer, 2020.

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

[33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[34] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[35] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

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

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

[38] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. *Advances in Neural Information Processing Systems*, 2019.

[39] Ekaterina Tolstaya, Reza Mahjourian, Carlton Downey, Balakrishnan Vadarajan, Benjamin Sapp, and Dragomir Anguelov. Identifying driver interactions via conditional behavior prediction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

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

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] Jiashun Wang, Huazhe Xu, Medhini Narasimhan, and Xiaolong Wang. Multi-person 3d motion prediction with multi-range transformers. *Advances in Neural Information Processing Systems*, 2021.

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[44] Waymo. Waymo open dataset interaction prediction challenge 2021. `https://waymo.com/open/challenges/2021/interaction-prediction/`, 2021. Accessed: 2022-05-19.

[45] Waymo. Waymo open dataset motion prediction challenge 2021. `https://waymo.com/open/challenges/2021/motion-prediction`, 2021. Accessed: 2022-05-19.

[46] David Wu and Yunnan Wu. Air2 for interaction prediction. *Workshop on Autonomous Driving, CVPR*, 2021.

[47] Maosheng Ye, Tongyi Cao, and Qifeng Chen. Tpcn: Temporal point cloud networks for motion forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11318–11327, 2021.

[48] Fangao Zeng, Bin Dong, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. *arXiv preprint arXiv:2105.03247*, 2021.

[49] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022.

[50] Yifan Zhang, Jinghuai Zhang, Jindi Zhang, Jianping Wang, Kejie Lu, and Jeff Hong. A novel learning framework for sampling-based motion planning in autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[51] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, 2020.

[52] Chen Lv Zhiyu Huang, Xiaoyu Mo. Recoat: A deep learning framework with attention mechanism for multi-modal motion prediction. *Workshop on Autonomous Driving, CVPR*, 2021.

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work is only for academic research purpose.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our implementation details are posted on Section 4.1 and the supplemental materials. The data is public dataset and our code will be available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1 for training details, and the data splits of ablation experiments are mentioned in Section 4.3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We fix the random seed for all our experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See our reference.
   (b) Did you mention the license of the assets? [No] They are publicly available for academic use.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] They are publicly available for academic use.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We only use the public Waymo Open Motion Dataset.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.