MTR v3: 1st Place Solution for 2024 Waymo Open Dataset Challenge - Motion Prediction

Chen Shi[1,2] Shaoshuai Shi[2] Li Jiang[1†]
[1]The Chinese University of Hong Kong (Shenzhen) [2]DiDi Global
{cshi0776, shaoshuaics}@gmail.com jiangli@cuhk.edu.cn

Abstract

In this report, we present the winning solution, named MTR v3, for the Waymo Open Dataset Motion Prediction Challenge 2024. The proposed MTR v3 builds upon MTR++ and offers improvement in two aspects: (1) we incorporate raw lidar data to provide fine-grained semantic information for motion prediction; and (2) we utilize evolving and distinct anchors to promote model's regression capability. In addition, a simple model ensemble technique is adopted to further boost the final performance. MTR v3 reaches a soft mAP of 0.4967 and ranks 1st place in the challenge, outperforming other methods with remarkable margins.

1. Introduction

Motion prediction, which refers to estimating agents' future trajectories based on historical tracks and HD maps, is a fundamental task in the field of autonomous driving. In recent years, motion prediction has attracted extensive attention [4, 5, 7, 9, 11, 17] as it is vital for robotic vehicles to make safe decisions. Among existing research, MTR [8, 13–15] series have achieved remarkable success. They employ an encoder network for scene context encoding, paired with a decoder network that generates multimodal trajectories from a set of intention queries which are initialized from predefined anchors.

Our solution, MTR v3, is an extension of MTR++ [14], a cutting-edge motion prediction framework, and improves with the incorporation of raw lidar data in scene encoding and the evolution of intention queries in trajectory decoding. Specifically, we introduce a lidar encoder to capture essential scene context information, such as vegetation and buildings, which are often missing in HD maps yet crucial for predicting pedestrian motion. Additionally, vanilla MTR++ suffers from high regression errors due to the sparsity of anchors. To mitigate this issue, we employ evolving and distinct anchors in [8] to adaptively update anchors based on specific scenes. Leveraging these techniques, our solution achieves 0.4593 soft mAP on the validation set with a single model. By applying a model ensemble strategy, we further boost the performance to 0.4967 soft mAP on the test set, placing 1st in the motion prediction track of 2024 Waymo Open Dataset Challenges.

†: corresponding author.

2. Method

The overall architecture of our approach is illustrated in Fig 1. Our proposed solution builds upon MTR++ [14]. To augment the 3D context information, we integrate lidar point clouds into the scene context encoder layer (described in Sec 2.1). The agents dynamically collect relative point features based on their motion modes, as detailed in Section 2.2. Finally, we outline the ensemble technique used in our study (see Sec 2.3).

2.1. Model Design

Our solution, MTR v3, evolves from MTR++, a state-of-the-art motion prediction method. Given past states of agents, road maps, and raw lidar points, an encoder network encodes scene context and generates multimodal scene tokens. Then a motion decoder network is adopted to predict multimodal future trajectories.

Scene Encoder Network. The historical agent's states are denoted as $S_{\text{A}} \in \mathbb{R}^{N_a \times T_h \times C_a}$, where N_a is the number of agents, T_h is the length of the historical observations, and C_a is the feature dimension. The road maps are transformed into polylines and represented as $S_{\text{M}} \in \mathbb{R}^{N_m \times n \times C_m}$, where N_m denotes the number of polylines, n indicates the number of points in each polyline, and C_m is the feature dimension associated with each point along the polyline. Following MTR++, both S_{A} and S_{M} are normalized to their local coordinate systems and encoded using a PointNet-like [12] polyline encoder, producing the agent feature $A \in \mathbb{R}^{N_a \times D}$ and map feature $M \in \mathbb{R}^{N_m \times D}$. Subsequently, a simple transformer encoder network, composed of a set of query-centric local self-attention layers, is adopted to aggregate features from concatenated tokens $F_{\text{AM}} = [A, M] \in \mathbb{R}^{(N_a+N_m) \times D}$,

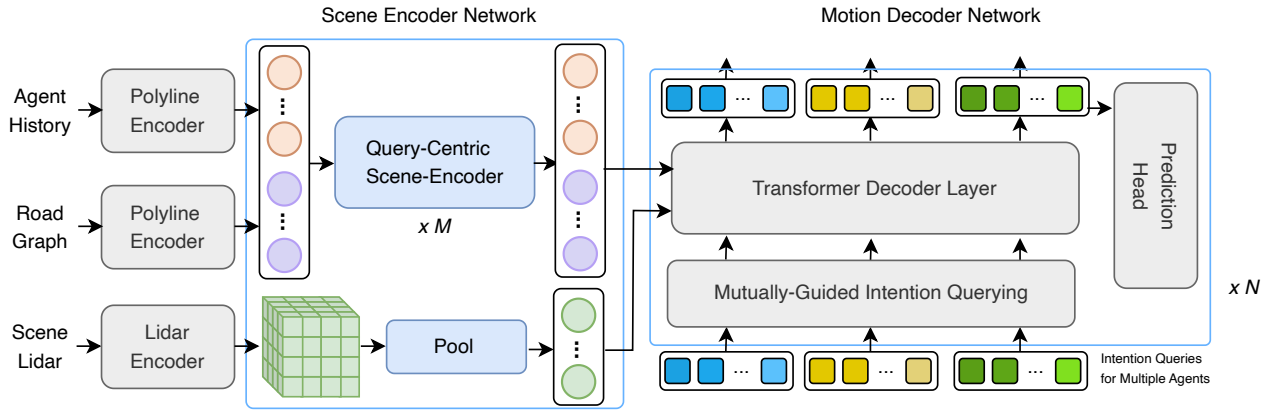

Figure 1. An overview of our proposed MTR v3 framework. Multimodel inputs are fed into an encoder network to extract scene tokens. Then, a decoder network is used to generate motion prediction of multiple agents.

which can be formulated as follows:

$$\begin{aligned} \mathrm{Q} &= \left[F_{\mathrm{AM}}[i], \mathrm{PE}\left(R_{\mathrm{AM}}[i,i]\right)\right], \\ \mathrm{K} &= \left\{\left[F_{\mathrm{AM}}[j], \mathrm{PE}\left(R_{\mathrm{AM}}[i,j]\right)\right]\right\}_{j\in\Omega(i)}, \\ \mathrm{V} &= \left\{F_{\mathrm{AM}}[j] + \mathrm{PE}\left(R_{\mathrm{AM}}[i,j]\right)\right\}_{j\in\Omega(i)} \\ F'_{\mathrm{AM}}[i] &= \mathrm{MHSA}(\mathrm{Q},\mathrm{K},\mathrm{V}), \end{aligned} \tag{1}$$

where $\Omega(i)$ indicates the operation that retrieves index set of the k neighborhoods of the i-th token and $R_{\mathrm{AM}}[i,j]$ signifies the relative position and direction of the j-th token in the local coordinate system centered on the i-th token. $\mathrm{PE}(\cdot)$ represents sinusoidal positional encoding function and $\mathrm{MHSA}(\cdot,\cdot,\cdot)$ denotes multi-head self-attention layer [18].

In order to supplement context information that may be absent in the agent tracks or the road maps, we incorporate point cloud data $P \in \mathbb{R}^{N_p \times C_p}$ collected at the current time to provide a more detailed understanding of the environment. Specifically, we first normalize point data according to the positions and headings of N_o focal agents, yielding $\{P^{(i)} \in \mathbb{R}^{N_p^{(i)} \times C_p}\}_{i=1}^{N_o}$. Then a lidar segmentation network is employed to extract voxel features and generate semantic label of each voxel. Following [3], the one-hot encoded labels and voxel positions are concatenated with the extracted features, resulting in an enriched voxel feature representation $P_v = \{P_v^{(i)} \in \mathbb{R}^{C_v \times H \times W \times Z}\}_{i=1}^{N_o}$, where H, W, Z are spatial resolution of voxel space and C_v is the feature dimension after concatenation. Furthermore, P_v is pooled into BEV space and encoded through a multi-layer perceptron as follows:

$$L^{(i)} = \mathrm{MLP}(\mathrm{AvgPool}(P_v^{(i)})), \tag{2}$$

where $L^{(i)} \in \mathbb{R}^{N_l^{(i)} \times D}$ indicates $N_l^{(i)}$ non-empty lidar tokens.

Motion Decoder Network. In adherence to the core designs of MTR++, we derive intention queries of each focal agent from $\mathcal{K}$ intention points (i.e. anchors), which are generated by utilizing the k-means clustering algorithm on the endpoints of ground truth trajectories. Given intention queries, we stack a set of decoder layers to aggregate context features from the encoded scene context features. Concretely, a mutually-guided intention querying module is used to model the interaction among intention queries, which are then updated through three separate cross-attention modules for aggregating information from A, M, and L, respectively. Moreover, we employ dense predictions of all agents to identify interactive agents for each focal agent, primarily focusing on them to encode scene-compliant information. After each decoder layer, we apply a simple prediction head with several MLP layers on refined queries to predict multimodal future trajectories, which is represented by Gaussian Mixture Model [1, 14].

Similar to MTR++, the training loss of our framework is a weighted combination of a classification loss on predicted intention probabilities, a GMM regression loss on trajectories of positive intention queries, and an auxiliary loss derived from dense predictions of all agents. MTR++ uses a hard-assignment strategy that chooses positive queries closest to the endpoints of ground truth trajectories, which tends to limit the regression capability of the model. To tackle this issue, we adopt the evolving and distinct anchors in EDA [8], where positive intention queries are dynamically selected based on the endpoints of predicted trajectories and updated as the predictions evolve.

2.2. Motion-Guided Lidar Search

To reduce the computational burden, we follow [3] to employ the motion-guided lidar search to gather spatially and temporally aligned lidar tokens. For each focal agent, we choose its latest position and the future location projected from its current velocity as search targets. Then $\tilde{N}_l$ nearest lidar tokens are collected and inputted into decoder network for extracting meaningful lidar context.

Methods	**Soft mAP** ↑	mAP ↑	minADE ↓	minFDE ↓	Miss Rate ↓
MTR v3 1^{st} (Ours)	**0.4967**	**0.4859**	**0.5554**	**1.1062**	**0.1098**
ModelSeq 2^{nd}	0.4737	0.4665	0.5680	1.1766	0.1204
RMP_Ensemble 3^{rd}	0.4726	0.4553	0.5596	1.1272	0.1113
BeTop	0.4698	0.4587	0.5716	1.1668	0.1176
BehaveOcc	0.4678	0.4556	0.5723	1.1668	0.1176
QMTR	0.4649	0.4445	0.5702	1.1627	0.1177
QMTR-V2	0.4646	0.4441	0.5700	1.1621	0.1174
EDA	0.4596	0.4487	0.5718	1.1702	0.1169
RMP	0.4572	0.4423	0.5695	1.1658	0.1160
ControlMTR	0.4572	0.4414	0.5897	1.1916	0.1282

Table 1. Waymo Open Dataset Motion Prediction Challenge leaderboard. Top 10 entries are presented and the soft mAP is the primary ranking metric.

Setting	Category	mAP ↑	minADE ↓	minFDE ↓	Miss Rate ↓
MTR ++	Vehicle	0.4702	0.7500	1.4822	0.1447
	Pedestrian	0.4891	0.3489	0.7263	0.0711
	Cyclist	0.3555	0.7106	1.4322	0.1853
	Avg	0.4382	0.6031	1.2135	0.1337
MTR v3	Vehicle	0.4819	0.6884	1.3830	0.1149
	Pedestrian	0.5168	0.3351	0.6898	0.0612
	Cyclist	0.3792	0.7139	1.4700	0.1764
	Avg	0.4593	0.5791	1.1809	0.1175
MTR v3 (Ensemble)	Vehicle	0.5141	0.6726	1.3210	0.1155
	Pedestrian	0.5196	0.3309	0.6816	0.0602
	Cyclist	0.4295	0.6582	1.3096	0.0811
	Avg	0.4877	0.5539	1.1041	0.1097

Table 2. Per-class performance on the validation set of Waymo Open Motion Dataset.

2.3. Model Ensemble

In order to further enhance the performance, we train N_e variants of our framework and adopt model ensemble strategy in inference. Each model outputs 6 predicted trajectories, which yields $6N_e$ results for each agent. After combination, we apply non-maximum-suppression (NMS) on them to select top 6 predictions.

3. Experiments

3.1. Implementation Details

Network Details. We build the MTR v3 framework with 6 encoder modules and 6 decoder modules. The hidden feature dimension D is established at 256 and 16 neighbors are gathered in the encoder's local self-attention. As for the lidar encoder, we use off-the-shelf lidar segmentation network MSeg3D [6], which is pre-trained on Waymo Open Dataset [16] and is frozen throughout our training process. We downsample the voxel feature using average pooling with a stride of (16, 16, 32), resulting in lidar tokens with a grid size of $(1.6m, 1.6m, 6m)$.

For the decoder modules, the number of intention points is set as $\mathcal{K} = 64$ and we collect 192 lidar tokens for local context enhancement. The anchors evolve twice, specifically at layers 2 and 4. During testing, for each agent, NMS based on the distances between endpoints is adopted to select top 6 predictions from 64 predicted trajectories. The distance threshold is held as $2.5m$.

Training details. Our model is trained end-to-end on 8 Tesla A100 GPUs for 30 epochs with an initial learning rate of 0.0001, a weight decay of 0.01, and a batch size of 80. We use AdamW [10] optimizer with one cycle policy, which decays the learning rate by a factor of 0.5 every 2 epochs from epoch 20. Data augmentation strategies like random

scene crop and random scene scale are included during the training. All models submitted to leaderboard are trained on training set of Waymo Open Motion Dataset (WOMD) [2].

3.2. Main Results

Top 10 results of 2024 Waymo Open Dataset Motion Prediction challenge are summarized in Table 1. Our approach ranks 1^{st} on the leaderboard and reaches 0.4967 in terms of soft mAP, surpassing all other submissions by 2.30%. MTR v3 also achieves state-of-the-art performance on mAP, min ADE, min FDE, and the miss rate. Besides, in Table 2, we reported our single-model results and the model ensemble results on the validation set. Notably, it achieves a +2.77% improvement in mAP of pedestrian.

4. Conclusion

In this technical report, we present MTR v3, which leverages raw LiDAR data to enhance scene context features, thereby facilitating the motion prediction of pedestrians. In addition, we incorporate dynamically updating anchors to further enhance the model's regression performance.

References

[1] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 2

[2] 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 *ICCV*, 2021. 4

[3] Yiqian Gan, Hao Xiao, Yizhe Zhao, Ethan Zhang, Zhe Huang, Xin Ye, and Lingting Ge. Mgtr: Multi-granular transformer for motion prediction with lidar. *arXiv preprint arXiv:2312.02409*, 2023. 2

[4] 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 *CVPR*, 2020. 1

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

[6] Jiale Li, Hang Dai, Hao Han, and Yong Ding. Mseg3d: Multi-modal 3d semantic segmentation for autonomous driving. In *CVPR*, 2023. 3

[7] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 1

[8] Longzhong Lin, Xuewu Lin, Tianwei Lin, Lichao Huang, Rong Xiong, and Yue Wang. Eda: Evolving and distinct anchors for multimodal motion prediction. In *AAAI*, 2024. 1, 2

[9] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[11] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, 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 multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 1

[12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1

[13] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *NeurIPS2022*, 2022. 1

[14] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *TPAMI*, 2024. 1, 2

[15] Jiawei Sun, Chengran Yuan, Shuo Sun, Shanze Wang, Yuhang Han, Shuailei Ma, Zefan Huang, Anthony Wong, Keng Peng Tee, and Marcelo H Ang Jr. Controlmtr: Control-guided motion transformer with scene-compliant intention points for feasible motion prediction. *arXiv preprint arXiv:2404.10295*, 2024. 1

[16] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay K. Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *CVPR*, 2020. 3

[17] 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 *ICRA*, 2022. 1

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2