ModeSeq: Technical Report for 2024 Waymo Open Motion Dataset Challenge - Motion Prediction

Zikang Zhou[1] Jianping Wang[1] Yung-Hui Li[2] Yu-Kai Huang[3]
[1]City University of Hong Kong [2]Hon Hai Research Institute [3]Carnegie Mellon University

Abstract

This technical report presents a brand-new framework for multimodal motion prediction. The framework is based on sequential mode modeling, where trajectory modes are decoded sequentially utilizing an RNN-style Transformer module. We also propose the Early-Match-Take-All (EMTA) loss, a customized training strategy for sequential mode modeling, to tackle the challenge of lacking multimodal ground truth. Our approach achieves state-of-the-art results on the 2024 Waymo Open Motion Prediction Benchmark, ranking second on the leaderboard.

1. Introduction

Motion prediction is a critical task for autonomous driving. To anticipate multimodal future behavior of traffic agents, existing approaches employ a parallel decoding module (*e.g.*, a DETR-like decoder [1, 6, 7, 10]) and the Winner-Take-All (WTA) training strategy [3]. However, such a paradigm leads to predicted trajectories with limited diversity. Take the DETR-like trajectory decoding module as an example. Unlike in object detection, where each object query can receive training signals via optimal bipartite matching, we have only one ground-truth future in the motion data, only being able to optimize one predicted trajectory per agent. As a result, the diversity of trajectory modes cannot be guaranteed even if we perform message passing among modes in the decoder. To improve the performance of mAP and Soft mAP, which requires predicting diverse multimodal trajectories, top-performing solutions on the Waymo Open Motion Prediction Benchmark [2] rely on dense mode prediction with a post-processing step of non-maximum suppression (NMS) [6, 7], which is not elegant.

In this technical report, we propose ModeSeq, an interesting (but probably not practical) method for multimodal motion prediction: no anchors, no NMS, no dense mode prediction, no auxiliary training tasks, just purely end-to-end learning. Our framework applies RNN-style modules to the mode dimension and performs sequential mode decoding, which can explicitly model the relationship between modes leveraging the inductive bias of RNNs. Similar ideas can be traced back almost ten years ago for achieving NMS-free object detection [9]. To make RNNs great again, we modernize RNNs with the Transformer architecture and derive a Memory Transformer module. To tackle the unique challenge in motion prediction that only one trajectory mode is available in the ground truth, we introduce an Early-Match-Take-All (EMTA) training strategy to accommodate sequential mode modeling and obtain more diverse multimodal trajectories. Our method performs well on mAP, Soft mAP, and Miss Rate without relying on dense mode prediction while outperforming all solutions except QCNet [10] on minADE and minFDE, which demonstrates the great potential of sequential mode modeling.

2. Methodology

This section introduces the ModeSeq framework, including the encoder, the decoder, and the training strategy.

2.1. Encoder

Our encoder follows QCNet [10], which employs factorized Transformers with relative spacetime representation to obtain scene embeddings with roto-translation invariance in space and translation invariance in time. The encoder stacks interleaved temporal Transformers, agent-map Transformers, and agent-agent Transformers, producing map embeddings of shape $[M, D]$ and agent embeddings of shape $[A, T, D]$, where M, A, T, D refer to the numbers of map instances, agents, past time steps, and hidden dimensions.

2.2. ModeSeq Layer

This section elaborates on the detailed structure of a single ModeSeq layer, which consists of a Memory Transformer module and a Factorized Transformer module. Stacking multiple ModeSeq layers can further improve the performance, which we leave the details to the next section.

Figure 1 shows the structure of the ℓ-th ModeSeq layer. In general, we let the ModeSeq layer decode multiple trajectory modes step by step. At the τ-th decoding step, the

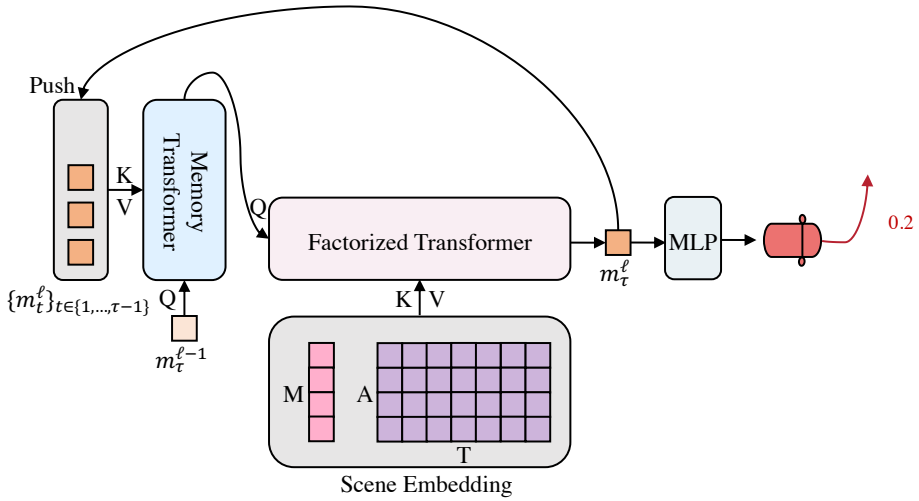

Figure 1. Overview of the ℓ-th ModeSeq Layer at the $\tau = 4$ recurrent step, where the 4-th trajectory mode is being decoded.

initial query feature $\mathbf{m}_\tau^{\ell-1}$ is updated by a Memory Transformer and a Factorized Transformer to become $\mathbf{m}_\tau^\ell$. The updated feature $\mathbf{m}_\tau^\ell$ is decoded by MLP layers to produce the τ-th trajectory and its confidence score. On the other hand, we push $\mathbf{m}_\tau^\ell$ into a queue to prepare for the subsequent decoding steps. In the following, we detail the Memory Transformer and Factorized Transformer modules.

Memory Transformer. The Memory Transformer is a Transformer-style RNN that models the sequential dependencies of trajectory modes. When decoding the τ-th trajectory mode based on the query feature $\mathbf{m}_\tau^{\ell-1}$, we hope it can be aware of the previously decoded modes. To this end, the Memory Transformer updates $\mathbf{m}_\tau^{\ell-1}$ by letting it attend to $\{\mathbf{m}_t^\ell\}_{t\in\{1,\ldots,\tau-1\}}$, the queries stored in a queue at previous decoding steps.

Factorized Transformer. Now that the query feature updated by the Memory Transformer is aware of the previously decoded modes, we further enrich it with the scene context features produced by the encoder. To this end, the factorized Transformer module refines the query feature using a temporal Transformer, an agent-map Transformer, and an agent-agent Transformer. The query at the τ-th decoding step goes through these three layers and becomes $\mathbf{m}_\tau^\ell$.

Prediction Head. Given $\mathbf{m}_\tau^\ell$, we use two MLPs to decode the τ-th trajectory and confidence score, respectively. On the other hand, we push $\mathbf{m}_\tau^\ell$ into a queue comprising $\{\mathbf{m}_t^\ell\}_{t\in\{1,\ldots,\tau-1\}}$ since the decoding of the subsequent modes relies on $\{\mathbf{m}_t^\ell\}_{t\in\{1,\ldots,\tau-1\}} \cup \mathbf{m}_\tau^\ell$.

2.3. Stacked ModeSeq Layers

Inspired by DETR [1], we stack multiple ModeSeq layers and apply training losses to the output of each layer for iterative refinement. At the first layer where $\ell = 1$, we set $\mathbf{m}_t^0$ as a learnable embedding $\mathbf{e} \in \mathbb{R}^D$ for $\forall t \in \{1, \ldots, K\}$, where K is the maximum number of recurrent steps. We call this learnable embedding the "next query" since, at each recurrent step, it looks at the preceding mode features stored in the queue before decoding the next mode. Starting from the second ModeSeq layer, we use the mode embeddings in the previous layer's queue as the query input of the Memory Transformer. We also perform mode rearrangement, an important operation during the transition between two ModeSeq Layers, which we will introduce next.

Mode Rearrangement. In sequential mode modeling, we desire the decoder to output trajectory modes with monotonically decreasing confidence scores. To this end, we rearrange the mode embeddings in the queue before starting a new round of recurrent decoding in the next ModeSeq layer. Specifically, given the mode embeddings $\{\mathbf{m}_t^\ell\}_{t\in\{1,\ldots,K\}}$ produced by the ℓ-th layer, we sort them according to the descending order of the confidence scores predicted from them. The sorted mode embeddings will then be sequentially input to the $(\ell + 1)$-th ModeSeq layer for recurrent decoding.

2.4. Early-Match-Take-All Training Strategy

In this section, we illustrate the regression and classification losses of the EMTA training strategy.

Regression. Our regression loss is based on the Laplace negative log-likelihood [10, 11]. Typical WTA loss optimizes only the trajectory with the minimum displacement error to the ground-truth trajectory. In contrast, our EMTA loss optimizes the matched trajectory decoded at the earliest recurrent step. For example, if both the 2nd and the 3rd trajectory modes match the ground truth, only the 2nd one will be optimized, regardless of which mode has the minimum displacement error. Here, we decide whether a predicted trajectory is a match based on the velocity-aware distance

Model	Soft mAP↑	mAP↑	minADE↓	minFDE↓	Miss Rate↓
QCNet	0.4508	0.4452	**0.5122**	**1.0225**	0.1254
ModeSeq	**0.4562**	**0.4507**	0.5237	1.0681	**0.1206**

Table 1. Single-model results on the validation set of WOMD.

thresholds defined in the Miss Rate metric of the Waymo Open Motion Prediction Benchmark. We also require the future waypoints at every time step to match the ground truth. If none of the predicted modes match the ground truth, we will fall back to the regular WTA loss. With such a training strategy, we encourage models to decode matching trajectories as early as possible, thereby improving the performance of Miss Rate.

Classification. We use the Binary Focal Loss to optimize the confidence scores. As for label assignment, we treat the earliest matches as positive samples, the modes decoded later than the positive samples as negative samples, and the modes decoded earlier than the positive samples as ignored samples. By assigning monotonically decreasing labels, we encourage the model to produce the more confident modes at the beginning and the less confident ones at the end.

3. Experiments

3.1. Implementation Details

The hidden size we use is 128, totaling 11M model parameters. The implementation details of the encoder can be referred to QCNet [10]. Our decoder stacks 6 ModeSeq layers for iterative refinement, and each ModeSeq layer executes 6 recurrent steps to obtain exactly 6 modes as required by the challenge. We use the AdamW optimizer [5] to train models for 30 epochs on the training split of the Waymo Open Motion Dataset (WOMD) with a batch size of 32, an initial learning rate of 5×10^{-4}, a weight decay rate of 0.1, and a dropout rate of 0.1. The learning rate is decayed to 0 based on the cosine annealing schedule [4].

3.2. Ensembling

Inspired by Weighted Boxes Fusion (WBF) [8], we propose Weighted Trajectory Fusion (WTF) to aggregate multimodal trajectories produced by multiple models. Our ensemble method is almost the same as WBF, except we are fusing trajectories according to distance thresholds rather than bounding boxes according to IOU thresholds. The WTF can improve mAP/Soft mAP/Miss Rate by sacrificing minADE/minFDE, which indicates that the performance on various metrics often disagrees.

3.3. Quantitative Results

The single-model results on the validation set of WOMD are shown in Tab. 1. Compared with QCNet [10], ModeSeq performs better on mAP, Soft mAP, and Miss Rate at the cost of worse minADE and minFDE. The significantly better minFDE of QCNet may be attributed to its recurrence mechanism in the time dimension, which may also apply to our ModeSeq framework (though the inference latency would be much higher). We feel the performance gap in minADE and minFDE is acceptable, given that the sequential mode modeling framework was developed in merely one month while QCNet is a highly optimized model. Also, we are unaware of any other motion prediction model with better minADE/minFDE than ours. A new framework usually undergoes years of iteration, and we welcome the community to explore more to fill in the critical technical details we have neglected here.

Category	Soft mAP↑	mAP↑	minADE↓	minFDE↓	Miss Rate↓
Vehicle	0.5181	0.5095	0.6780	1.4046	0.1129
Pedestrian	0.4781	0.4709	0.3407	0.7117	0.0776
Cyclist	0.4248	0.4190	0.6852	1.4136	0.1706
Avg	0.4737	0.4665	0.5680	1.1766	0.1204

Table 2. Ensemble results on the test set of WOMD.

The WTF-based ensemble results on the test set of WOMD are shown in Tab. 2. The critical hyperparameters in WTF are the distance thresholds used for trajectory clustering. We choose the velocity-aware thresholds defined for the Miss Rate metric of the benchmark as the base thresholds. On top of this, we found that using larger thresholds can improve mAP and Soft mAP by substantially sacrificing Miss Rate, minADE, and minFDE, implying the disagreement of various metrics. Since this is a competition, we overfitted Soft mAP by multiplying the base thresholds with the scaling factors of 1.5, 1.4, and 1.4 for vehicles, pedestrians, and cyclists, respectively, without caring about the actual performance of models.

4. Conclusion

We propose ModeSeq, a sparse mode prediction method for multimodal motion forecasting based on sequential mode modeling and the Early-Match-Take-All (EMTA) training strategy. This modeling paradigm enables diverse multimodal motion prediction without sacrificing too much in minADE and minFDE, achieving state-of-the-art results on the Waymo Open Motion Prediction Benchmark. We hope the combination of sequential mode modeling and the EMTA training strategy can provide new insights into multimodal problems and serve as an alternative to parallel multimodal decoding trained with the Winner-Take-All loss.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-

end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[3] Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1

[4] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 3

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 3

[6] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022. 1

[7] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[8] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021. 3

[9] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[10] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3

[11] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2