QML for Argoverse 2 Motion Forecasting Challenge

Tong Su **Xishun Wang** **Xiaodong Yang**
QCraft

Abstract

To safely navigate in various complex traffic scenarios, autonomous driving systems are generally equipped with a motion forecasting module to provide vital information for the downstream planning module. For the real-world onboard applications, both accuracy and latency of a motion forecasting model are essential. In this report, we present an effective and efficient solution, which ranks the 3rd place [1] in the Argoverse 2 Motion Forecasting Challenge 2022.

1 Introduction

As a core component of an autonomous vehicle (AV), motion forecasting or trajectory prediction plays a crucial role to understand the behaviors of traffic agents. A prediction module leverages the perception information [2] and outputs multi-modal future trajectories for nearby agents. However, this is a challenging task due to the uncertainty of traffic actors and complexity of road topology [3]. In this report, we present an effective and efficient approach, which forecasts the future trajectories of multi-class agents around the ego vehicle. Specifically, our model consists of five parts: (1) agent history encoder that takes the agent-centric data as input, (2) agent interaction encoder dynamically models the interactions of traffic agents, (3) vector map encoder making use of the local vector map to provide the scene context [4], (4) anchor decoder and (5) prediction decoder which receive the context features and respectively produce the proposals and future predictions.

2 Methodology

A trajectory prediction model aims to predict the trajectories of agents in the future T timesteps based on the history observation of their past H timesteps. In this section, we will describe our proposed approach as illustrated in Figure 1.

2.1 Agent History Encoder

In order to accurately model the history observation of traffic agents, we first apply an agent-centric paradigm that coverts the trajectory of each agent to an ego-view coordinate system. The history observation contains trajectory, velocity, heading and object type. By taking advantage of this agent-specific ego-view, we standardise each actor trajectory such that the last timestep of history observation is always at the origin of coordinate system.

2.2 Agent Interaction Encoder

In the complex traffic scenarios, agents plan their future routes by interacting with their neighbors [5]. Therefore, modeling the interaction between agents is beneficial to predict their future trajectories. The agent interaction encoder takes the interaction information as input, which includes the relative position, velocity and heading. It is noted that we also convert the interaction input to the agent-centric coordinate system.

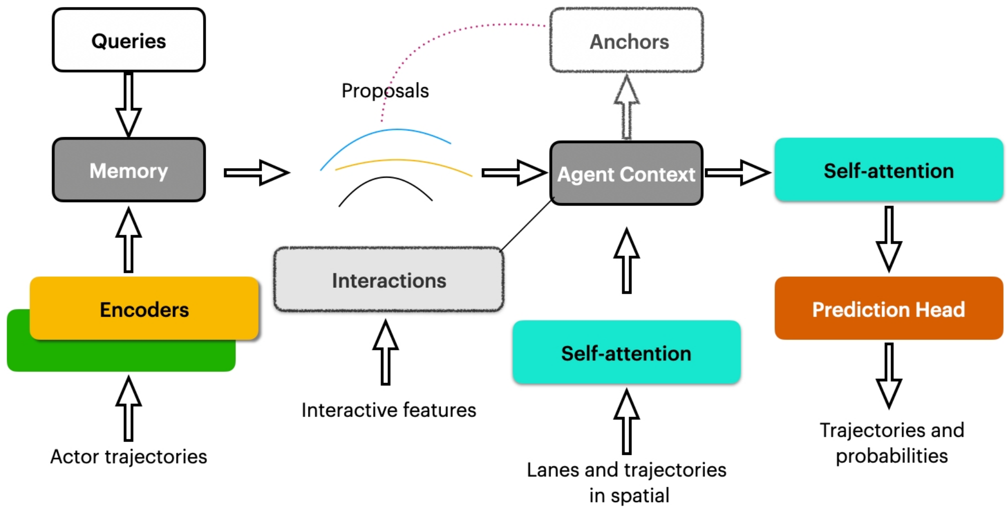

Figure 1: Illustration of the proposed approach. Encoders extract the temporal features and interaction features of history observations. We fuse the dynamic features to get the proposals, and then use self-attention to encode both trajectories and lanes to obtain the spatial context features. Together with the proposals, the agent context that is from the spatial context features are decoded to anchors. Finally, the prediction head outputs the multiple predicted trajectories and probabilities.

2.3 Vector Map Encoder

While the history observations of different agents are informative, the high-definition (HD) map further provides the complementary social context for agents to better plan their future trajectories. For the vector map, we pre-process each single vector map to lane segments. By splitting the vector map we can improve our vector map encoder both in accuracy and efficiency. For a single agent, we choose the lane segments within D meters. Note that D varies for different agent categories. Furthermore, we transform the lane segments to each agent view and construct a spatial topology feature. After that, the lane segment features are encoded into the social context [6].

2.4 Anchor Decoder and Prediction Decoder

Given the aforementioned features, we adopt the transformer network to integrate the social context [7]. In the decoder stage, the social context is fed into the anchor decoder to predict the embeddings of N anchors for each agent. Here the anchor embeddings can be decoded into future waypoints. At the same time, we can select the target agent feature from the social context, and the target agent feature is then decoded as the proposals. For the prediction decoder, we aim to produce K proposed future trajectories along with their corresponding probabilities for each agent. Together with proposals, the anchor embeddings are fed into the prediction decoder [8]. Finally, our model outputs the multi-modal trajectories and probabilities for each agent.

2.5 Training Details

Our loss is composed of two parts: anchor loss and prediction loss. For the anchor loss, we use the mean squared error and cross entropy of the future waypoints for each agent. For the prediction loss, we adopt the same strategy while for all future timesteps. We set the batch size as 12 for training. $K = 6$ is used to predict the multi-modal future trajectories.

3 Experiments

3.1 Dataset

We evaluate our proposed approach on Argoverse 2. This dataset consists of 250,000 scenarios with trajectory data for multiple object types (vehicle, pedestrian, motorcyclist, cyclist and bus). Each scenario is 11s long and is annotated at 10 frames per second. Each sample contains an agent marked as "focal track" that is needed to be predicted. Our trajectory prediction horizon is 60 timesteps (6s) based on the observed history of 15 timesteps (1.5s).

3.2 Quantitative Results

We follow the same evaluation protocol as defined in the challenge, and use the metrics including average displacement error (ADE), final displacement error (FDE), miss rate (MR), brier-minADE and brier-minFDE to evaluate the performance. The result on test set is shown in Table 1. For the final submission, we independently train M models and obtain MK candidate trajectories. At the inference stage, we perform the K-means clustering for all endpoints of the candidate trajectories. The trajectories and their probabilities in the same cluster are averaged and K predictions are generated. It is worth noting that our approach performs particularly well (ranks the 1st place) on metric minADE ($K = 6$), indicating that our model is stable among all modalities.

Table 1: Evaluation on the test set of Argoverse 2.

	minADE	minFDE	Miss Rate	brier-minADE	brier-minFDE
K = 1	1.8369	4.9779	0.6212	-	-
K = 6	0.6882	1.3850	0.1894	2.3183	1.9547

Furthermore, we report our baseline performance for each object category on the validation set of Argoverse 2, as shown in Table 2. We observe that the results of cyclist and bus are relatively inferior to other object types, which may result from the fact that they have different behaviors and insufficient training samples.

Table 2: Evaluation of our baseline on the validation set of Argoverse 2.

	minADE	minFDE	brier-minADE	brier-minFDE
Vehicle	0.8271	1.4478	1.5051	2.1258
Pedestrian	0.3482	0.6266	1.0196	1.2890
Cyclist	1.0017	1.7990	1.6721	2.4694
Motorcyclist	0.6643	1.2257	1.3401	1.9015
Bus	1.0320	1.4625	1.7144	2.1450

3.3 Qualitative Results

As demonstrated in Figure 2, we visualize two prediction examples of our approach on the validation set of Argoverse 2. The black lines and blue points represent the road lanes and other agents around the target agent. The cyan points denote the observed history trajectory of the target agent, and the yellow points are the multi-modal outputs of our model and the red line is the ground truth future trajectory. The endpoints of predictions and ground truth are colored magenta and green. As can be seen, when encountering crowded intersections, our model can make correct predictions no matter the target agent is moving or static.

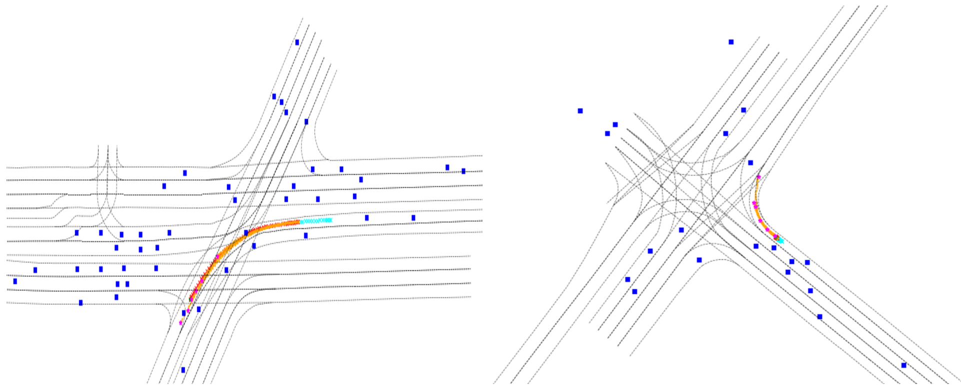

Figure 2: Visualization of two representative scenes. At the intersections, agents usually tend to have diverse maneuvers. Our approach can provide multiple reasonable predicted trajectories.

References

[1] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks Track*, 2021.

[2] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Exploring simple 3d multi-object tracking for autonomous driving. In *ICCV*, 2021.

[3] 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. *arXiv preprint arXiv:2111.14973*, 2021.

[4] 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.

[5] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017.

[6] 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.

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

[8] Bo Dong, Hao Liu, Yu Bai, Jinbiao Lin, Zhuoran Xu, Xinyu Xu, and Qi Kong. Multi-modal trajectory prediction for autonomous driving with semantic map and dynamic graph attention network. *arXiv preprint arXiv:2103.16273*, 2021.