MPA: MultiPath++ Based Architecture for Motion Prediction

Stepan Konev
stevenkonev@gmail.com

Abstract

Autonomous driving technology is developing rapidly and nowadays first autonomous rides are being provided in city areas. This requires the highest standards for the safety and reliability of the technology. Motion prediction part of the general self-driving pipeline plays a crucial role in providing these qualities. In this work we present one of the solutions for Waymo Motion Prediction Challenge 2022 based on MultiPath++ [16] ranked the 3rd as of May, 26 2022. Our source code is publicly available on GitHub[1].

1. Introduction

The most popular approach for creating an autonomous driving technology consists of multiple steps: receiving data from sensors, solving perception problem to recognize the surrounding objects, localization, motion prediction and finally motion planning. In this paper we focus solely on the motion prediction problem, considering the relative locations of other agents and road lines as ground truth. Motion prediction problem has been approached in multiple works however it remains a complex and challenging problem so far. One of the key difficulties is a natural uncertainty of other agents' behavior. In this paper we propose a solution based on MultiPath++ [16] which efficiently scores the 3rd on Waymo Motion Prediction Challenge 2022 and significantly outperforms our previous solution based on Convolutional Neural Network.

2. Related work

Motion prediction task is a challenging problem that attracts a lot of attention from researchers. There are two main approaches for this task: based on CNN [2, 3, 6, 8, 11, 13] and GNN [1, 4, 5]. For CNN the input data is represented as a dense tensor where the channel dimension is usually referred to as a discrete-time dimension. The GNN approach takes graph data as input where the road lines and previous agents' positions are represented as nodes within polylines. More details about our specific representation are in section 3.1. However CNN approach seems comfortable for its simplicity because CNNs are greatly explored and stable due to the regular structure of the input data it tends to lose its popularity in favor of the GNN approach.

3. Method

In this section we provide the description of the input data structure, its preprocessing, model architecture and training strategy followed by a postprocessing description.

3.1. Input data

For training our model we first did some preprocessing for agents of interest. Since our model is based on MultiPath++ [16] we follow similar process of data input data preparation as in MultiPath++ [16]. A usual approach is to first transform the frame into the canonical coordinate system where the agent we make prediction for is always located in the same position with the same heading at the moment we make prediction for. This step helps us to eliminate redundant symmetries. Road graph data was represented as in MultiPath++ [16]. For the target agent and other agents that surround the target agent for each timestamp in history (past + current) we computed x, y coordinates, heading, velocity in the mentioned canonical coordinate system and the validity boolean flag. We cached this precomputed data along with road graph data for faster training. The size of the dataset and corresponding splits remain the same as described in our previous work MotionCNN [10].

3.2. Model architecture

Our model follows the architecture of MultiPath++ [16] mostly, however we tried to alternate it a little (see Fig. 1). First we did not have a specific encoder for autonomous vehicle itself. Second we did not use proposed EM algorithm because in our experiments it performed numerically unstable. Instead we experimented with a single decoder with 6 modes and 5 decoders each with 6 modes followed by attention mechanism [17] and multi-context gating block (MCG) from MultiPath++ [16] that mapped 30 modes into required 6. With multiple decoders we used a proposed strategy of

[1] https://github.com/stepankonev/waymo-motion-prediction-chalenge-2022-multipath-plus-plus

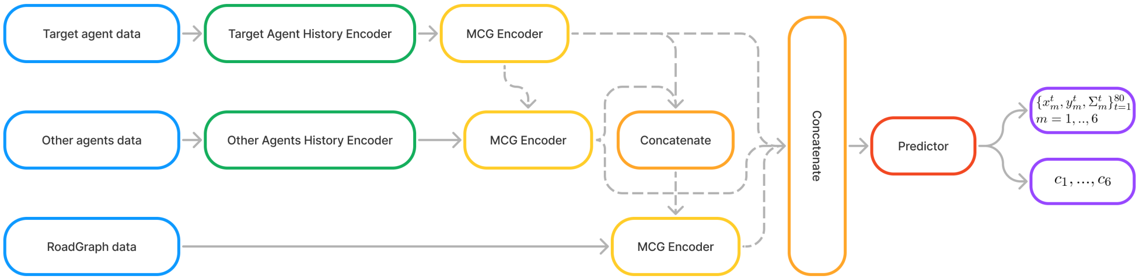

Figure 1. General overview of the architecture of out model based on MultiPath++ [16]

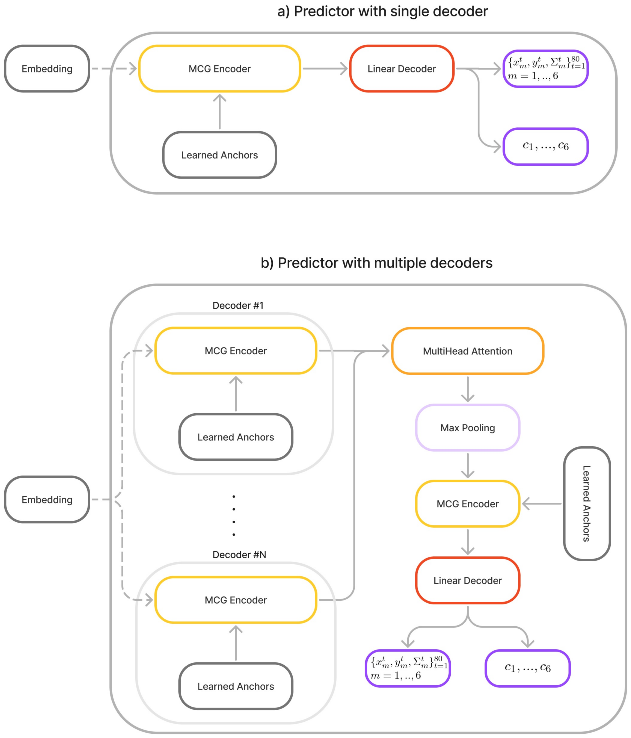

Figure 2. Architectures of the predictors. a) Predictor with a single decoder. b) Predictor with multiple decoders and attention [17] mechanism

blocking weighs update for randomly selected decoders (see Fig. 2).

3.3. Training strategy

For each target agent we predicted 6 modes, each consisted of 80 coordinates $\{(x_m^t, y_m^t)\}_{t=1}^{80}$ for $m = 1, ..., 6$ along with covariance matrices $\{\Sigma_m^t\}_{t=1}^{80}$. For each mode we also predicted its probability c_m. Thus the likelihood has the form

$$l = \sum_{m=1}^{6} c_m \prod_{t=1}^{80} N(\mu_{gt}^t - \mu_m^t, \Sigma_m^t)$$

where $\mu_m^t = (x_m^t, y_m^t)$ predicted coordinates for timestamp t and mode m and $\mu_{gt}^t = (x_{gt}^t, y_{gt}^t)$ is the corresponding ground truth coordinate. We used the negative log-likelihood as an objective function and optimized it w.r.t. the whole set of predicted trajectories. In our experiments models with trained covariance matrices performed better. We trained our model for 1.5 million iterations with initial learning rate of 10^{-4} with Adam [9] optimizer and ReduceLROnPlateau scheduler.

3.4. Augmentations

While training the model we faced some overfitting especially for minor agent classes. The natural way to avoid this problem is to use augmentations, however is it not obvious what augmentations may provide a quality improvement in this specific task. Thus finally we have chosen to use masking for the history data. For each timestamp of the history we randomly put the values to zero and the validity flag to false during training. The probability of masking a single timestamp $p_{mask} = 0.15$. We use masking only for historical data of both target and surrounding agents and do not mask the road graph.

3.5. Postprocessing

Since the proposed loss function does not directly optimize the target SoftMAP metric we decided to apply postprocessing. As MAP metric is typical for detection tasks we decided to use a typical approach - non-maximum suppression. This algorithm has been successfully used in multiple of previous works [5, 7, 12, 14, 18]. More specifically, in case where two trajectories appear to be close enough to each other the less probable was suppressed in favour for more probable one. However as the number of input trajectories is equal to the number of the output ones we do not completely drop the suppressed trajectories but assign some minor constant probability to them.

4. Results

Object Type	Soft mAP	mAP
Avg Vehicle	0.4467	0.4382
Avg Pedestrian	0.3831	0.3758
Avg Cyclist	0.3493	0.3458
Avg 3s	0.4875	0.4773
Avg 5s	0.3935	0.3883
Avg 8s	0.2981	0.2943
Total	0.3930	0.3866

Table 1. Detailed evaluation of our model on test set of Waymo Open Motion Dataset [15].

For the final submission we selected the best model for each of agent types. For cars and pedestrians the model with single decoder performed the best, whereas for cyclist the model with multiple decoders followed by attention mechanism outperformed other models. Numerical results are presented in Tab. 1

References

[1] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *International Conference on Robotics and Automation (ICRA)*, 2020. 1

[2] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, 2019. 1

[3] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019. 1

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 1

[5] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets, 2021. 1, 3

[6] 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. 1

[7] Xin Huang, Guy Rosman, Igor Gilitschenski, Ashkan Jasour, Stephen G. McGill, John J. Leonard, and Brian C. Williams. Hyper: Learned hybrid trajectory prediction via factored inference and adaptive sampling, 2021. 3

[8] Atsushi Kawasaki and Akihito Seki. Multimodal trajectory predictions for autonomous driving without a detailed prior map. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3723–3732, 2021. 1

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 2

[10] Stepan Konev, Kirill Brodt, and Artsiom Sanakoyeu. Motioncnn: A strong baseline for motion prediction in autonomous driving, 2022. 1

[11] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017. 1

[12] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers, 2021. 3

[13] 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*, pages 14074–14083, 2020. 1

[14] Haoran Song, Di Luan, Wenchao Ding, Michael Yu Wang, and Qifeng Chen. Learning to predict vehicle trajectories with model-based planning, 2021. 3

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

[16] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S. Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, and Benjamin Sapp. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction, 2021. 1, 2

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 1, 2

[18] Pengxiang Wu, Siheng Chen, and Dimitris Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps, 2020. 3