Supplementary Material for Learn TAROT with MENTOR: A Meta-Learned Self-supervised Approach for Trajectory Prediction

Mozhgan Pourkeshavarz, Changhe Chen, Amir Rasouli
Noah's Ark Lab, Huawei
Toronto, Canada
firstname.lastname@huawei.com

1. What is Exactly TAROT Modeling?

In the paper, we presented meTA ROad paTh (TAROT) to formulate common driving patterns in road topology. More specifically, we argued that this formulation is capable of embedding the map constraints; hence supplying this information to the model is similar to providing tips for navigating through the road. Therefore, TAROT prediction as a Heterogeneous Structural Learning (HSL) task can improve learning relations of nonadjacent interacting lanes. To highlight the effectiveness of TAROT in this way, we show an example in Fig. 1 where two scenarios are presented: Scenario 1 in which the adjacent lane runs in the opposite direction of the vehicle, and scenario 2 where the adjacent lane has the same direction as the vehicle. As noted in the paper, we model the scene as a directed Heterogeneous Information Network (HIN) where only lane vertices and edges are drawn (for two arbitrary nodes A and B, the edge from A to B is different from the edge from B to A, and for successors and predecessors relations, they are connected based on the direction of the lanes). Based on this formulation, here, we raise the question of whether is lane-changing feasible or not in both scenarios?

We define TAROT as a composite relation in the directed HIN, so the transition from the red point to the blue point in the driving scene is equal to reaching node v from node u with a path in HIN. From the figure, we can see that by defining a TAROT $p_{u \rightsquigarrow v}$ as S-L-S-S we can distinguish between two scenarios. Particularly, in scenario 2 node v is reachable from node u with TAROT $p_{u \rightsquigarrow v}$ while in scenario 1 it is not. Thus, by predicting the existence of TAROT, the model can gain a sense of the constraints and rules of the road.

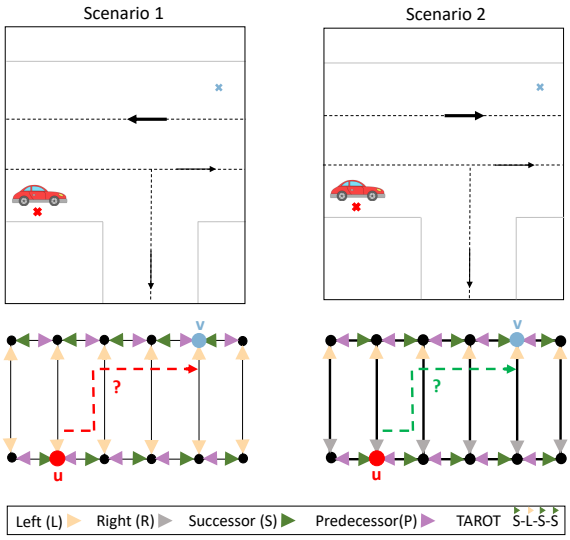

Figure 1: Two scenarios where lane changing is only feasible in one (scenario 2).

2. Method to Extract Challenging Scenarios

We divide the Argoverse benchmark dataset [4] into 3 categories based on the complete trajectory of the agent vehicle, namely (left/right/blind) turn, stationary and cruising, using the directions of the lane segments that vehicles travel through. Since cruising scenarios are generally simpler to predict, the focus is on the turn and stationary scenarios as challenging. We extract these scenarios as follows:

2.1. Left/Right/Blind Turn:

In general, turn scenarios can be split into Left/Right turns. To determine whether a turn occurs, the heading angle difference between the initial point in observation and the final point in ground truth is calculated. If the difference falls between 20 and 110 degrees, a turn is detected. Note that due to the occasional noisy trajectory data in the Argoverse Dataset, an upper limit of 110 degrees is set rather than 90 degrees to add some margin for misdetections. Next, we further split the turns into Left/Right based on the change in the vehicle's current lane direction using the Argoverse API. From general turn categories, we extract blind turns which are deemed to be more challenging. We observed that in

Table 1: Comparison to SOTA models on challenging turn scenarios. Arrows show lower (↓) or higher (↑) values are better.

Method	Left				Right				Blind (left/right)			
	minADE(↓)	minFDE(↓)	DAO(↑)	RF(↑)	minADE(↓)	minFDE(↓)	DAO(↑)	RF(↑)	minADE(↓)	minFDE(↓)	DAO(↑)	RF(↑)
LaneGCN [12]	0.99	1.79	64.34	3.08	0.99	1.83	63.10	3.13	1.17	2.56	71.55	2.66
MMTransformer[10]	1.00	1.89	68.84	3.58	0.94	1.69	62.32	3.61	1.12	2.36	73.95	3.10
FTGN [1]	1.01	1.78	64.57	3.34	0.99	1.80	63.69	3.49	1.12	2.27	77.18	3.09
HiVT [15]	0.94	**1.76**	65.56	3.00	0.95	1.83	63.77	3.04	1.17	2.66	69.02	2.56
SSL-Lane [2]	0.95	1.82	66.78	3.29	0.99	1.70	56.54	3.43	1.18	2.48	72.34	3.12
MENTOR (Ours)	**0.93**	**1.76**	**76.78**	**3.87**	**0.91**	**1.61**	**79.12**	**3.83**	**1.11**	**2.25**	**80.32**	**3.51**

Table 2: Comparisons with SOTA on nuScenes

Method	Venue	minADE_5	minADE_{10}	$\text{MR}_5(2m)$	$\text{MR}_{10}(2m)$
AutoBot [8]	ICLR 2022	1.37	1.03	0.62	0.44
PGP [6]	CoRL 2022	1.30	1.00	0.61	0.37
Thomas [7]	ICLR 2022	1.33	1.04	0.55	0.42
FRM [14]	ICLR 2023	**1.18**	**0.88**	**0.48**	**0.30**
MENTOR (Ours)		1.28	0.94	0.60	0.36

a number of turn scenarios, trajectories are straight at the observation horizon, and turns only occur at the end of the prediction horizon. Since the straight observation trajectory does not provide any hints about the turning, it is likely that the models mispredict the turning action.

2.2. Stationary:

We define stationary scenarios by measuring the relative length difference between observation and ground truth prediction portions of the trajectories. If the length of the ground truth trajectory is 3 times greater than the length of the observation trajectory, this scenario will be defined as short observation or stationary. Such scenarios tend to be more challenging for benchmark models given the limited historical information.

3. Per Scenario Metrics Comparison

In this section, we present per scenario evaluation of our method against past arts on turn cases. We use the same metrics as the paper and summarize the results in Tab. 1. As shown in the table, the right and left turns are generally easier for the models, and blind ones are more challenging. The performance of past arts varies in different scenarios, e.g. HiVT, performs better in left turn whereas MMTransformer and FTGN do better in right and blind turn scenarios. Our model, MENTOR, however, consistently performs best in all scenarios by achieving state-of-the-art performance on all metrics. In terms of diversity and admissibility metrics, our model also performs more consistently across different scenarios whereas other models' performance fluctuates significantly.

4. Experiments on a smaller dataset

Meta-learning paradigm aims at learning to learn models efficiently and effectively, and in particular is effective when large data is not available, as shown in few-shot learning settings [5]. Hence, the proposed MENTOR is also effective on small datasets, such as nuScenes [3], as shown in Table 2.

5. Inference Speed and Complexity vs Accuracy

Model complexity and inference speed are two major concerns in real-time applications, such as autonomous driving. We compare the inference time of our method with past arts and report the results in terms of minFDE and inference time on the Argoverse test set, and the number of model parameters. As shown in Fig. 2, although DenseTNT [9] has the smallest number of parameters (the smallest circle), its average inference speed is the highest (50ms per agent) due to the computationally expensive optimization algorithm to find dense goal sets to minimize expected error. Compared to MENTOR (Ours), in terms of inference time and the number of parameters, the closest model is LaneGCN [12], which slightly lags behind as it uses a complicated four-stage fusion mechanism. In terms of minFDE, however, the performance gap is much bigger because LanceGCN only models the local structure of the roads and uses multiple layers of processing, which can lead to problems with over-smoothing for map-encoders [11, 2].

SceneTransformer [13] as a more recent state-of-the-art model has a much better minFDE value compared to the past arts, however, this model has $15K$ parameters making it a very heavy model, and consequently difficult to train. Comparatively, our method, MENTOR, has both low complexity and inference time while achieving the best performance in terms of minFDE. These results further highlight the effectiveness of adopting self-supervised learning coupled with meta-learning as both techniques are designed for better representational learning while maintaining a reasonable number of model parameters. In addition, it is worth noting that our method's overhead is only on the training and not the inference.

6. Cost of training

As shown in Figure 1 of the paper, our method has two more components compared to arbitrary baseline f: MENTOR network (a two-layer MLP) and task-specific layers Φ_t (a single-layer MLP per task) for transforming the embeddings from f to calculate task-specific loss values. In our work, we use HGT as our baseline, which has fewer parameters than most recent SOTA models. Here, the added learnable parameters by the mentioned components are less than 15% of the base model's learnable parameters. Fur-

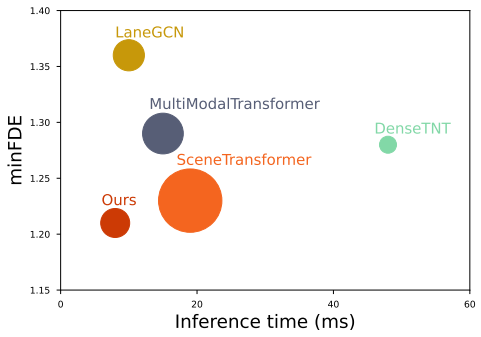

Figure 2: Performance comparison between Our model (MENTOR) and past arts in terms of minFDE, inference time, and the number of parameters (represented as the size of circles). Results are reported on the Argoverse test set and for all values, smaller is better.

thermore, K-fold cross-validation changes the training time from $O(n)$ to $O(Kn)$. In addition, All the experiments are conducted on NVIDIA Tesla V100.

References

[1] Görkay Aydemir, Adil Kaan Akan, and Fatma Güney. Trajectory forecasting on temporal graphs. *arXiv:2207.00255*, 2022. 2

[2] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. SSL-Lanes: Self-supervised learning for motion forecasting in autonomous driving. In *CoRL*, 2022. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2

[4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 1

[5] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *CVPR*, 2021. 2

[6] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *CoRL*, 2022. 2

[7] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. THOMAS: Trajectory heatmap output with learned multi-agent sampling. In *ICLR*, 2022. 2

[8] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. AutoBot: Latent variable sequential set transformers for joint multi-agent motion prediction. In *ICLR*, 2022. 2

[9] Junru Gu, Chen Sun, and Hang Zhao. DenseTNT: End-to-end trajectory prediction from dense goal sets. In *ICCV*, 2021. 2

[10] Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *ICRA*, 2022. 2

[11] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018. 2

[12] 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. 2

[13] 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 *ICLR*, 2022. 2

[14] Daehee Park, Hobin Ryu, Yunseo Yang, Jegyeong Cho, Jiwon Kim, and Kuk-Jin Yoon. Leveraging future relationship reasoning for vehicle trajectory prediction. In *ICLR*, 2023. 2

[15] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. HiVT: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, 2022. 2