Supplementary Material for Traj-MAE: Masked Autoencoders for Trajectory Prediction

1. Autobots Architecture.

Autobots [4] is a class of encoder-decoder architectures that process sequences of sets. This model is designed to process a tensor of dimensions $K \times M \times t$ as input, where K is the number of attributes of each agent, M is the number of agents, and t is the input sequence length. To transform the K-dimensional vectors to a new space of dimension d_K (hidden size), it applies a row-wise feed-forward network (rFFN) to each row along the $t \times M$ plane. Following the addition of positional encoding (PE) to the t axis, the encoder processes the tensor through L layers of repeated multi-head attention blocks (MAB) that apply time encoding and social encoding to the time and agent axes, respectively. Finally, the encoder outputs the context tensor. In the decoder, the encoded map and the learnable seed parameters tensor are first concatenated and then passed through an rFFN. The resulting tensor is then processed through L layers of repeated multi-head attention block decoder (MABD) along the time axis using the context from the encoder, followed by a MAB along the agent axis. The output of the decoder is a tensor of dimensions $d_K \times M \times T \times c$, which can then be element-wise processed using a neural network to produce the desired output representation. T is the output sequence length and c is the number of modes. AutoBot-Ego is a special case, which is similar to AutoBots but predicts future modes for only one agent in a scene.

2. Implementation Details.

For pre-training, we use an Adam optimizer [6] with a fixed learning rate 1e-3. We set the same number of training stages as the number of pre-training strategies. When performing continuous pre-training, the number of steps for each strategy in all training steps adds up to $120k$. In terms of the trajectory prediction task, we fine-tune Autobots on three challenging datasets, and utilize Adam as an optimizer. For the Argoverse dataset [3], we use Autobot-Ego as our baseline model, the initial learning rate is set at 3e-5, and for the Interaction [14] and TrajNet++ [9] datasets, Autobot is used as the baseline model, we set the initial learning rates at 5e-5 and 7e-5, respectively. We anneal the learning rate every $6k$ by a factor of 2 in the first $30k$ steps, and the total training steps are $120k$. The batch size of training and testing of all the above tasks is 64. The Traj-MAE is implemented by PyTorch and all experiments can be done on a single V100.

Method	minADE	minFDE	MR
DESIRE [7]	0.92	1.77	0.18
MultiPath [2]	0.80	1.68	0.14
TNT [15]	0.73	1.29	0.09
LaneRCNN [13]	0.77	1.19	0.08
TPCN [11]	0.73	1.15	0.11
mmTransformer [8]	0.71	1.15	0.11
DenseTNT [5]	0.82	1.37	**0.07**
HiVT[16]	0.66	0.96	0.09
DCMS[12]	0.64	**0.93**	-
GANet[10]	0.67	**0.93**	-
Autobot-Ego [4]	0.73	1.10	0.12
Traj-MAE	**0.60** ↓ 18%	1.00 ↓ 9%	0.09 ↓ 25%

Table 1: **Comparison with state-of-the-art methods on the Argoverse validation set**.

3. Metrics.

For the task to predict the ego-agent's future trajectory, we use minADE, minFDE and MR to evaluate our method, which are respectively the minimum Average Displacement Error, the minimum Final Displacement Error and the Miss Rate, respectively. Considering the multi-agents' future trajectories prediction task, we calculate ego-agent's prediction error and scene-level prediction error as defined by [1] on TrajNet++. Similarly, MinJointADE, MinJointFDE and MinJointMR are used to calculate multi-agents' prediction error. CrossCollisionRate represents the frequency of collisions happening among the predictions of all agents and EgoCollisionRate represents the collisions happening between ego-agent and others. When only considering those modalities without cross collision, Consistent MinJointMR is calculated as the case's miss rate. All of the above metrics are the lower the better. The lower metrics reflect better performance.

Method	Pre-training strategy	Training steps			Fine-tuning result		
		Stage1	Stage2	Stage3	minADE	minFDE	MR
Continual Learning	S	120k	-	-	0.664	1.075	0.108
	T	-	120k	-			
	ST	-	-	120k			
Multi-task Learning	S		120k		0.693	1.089	0.113
	T		120k				
	ST		120k				
Continual Pre-training	S	60k	30k	30k	**0.621**	**1.027**	**0.099**
	T	-	90k	30k			
	ST	-	-	120k			

Table 2: **Continual Pre-training for trajectory reconstruction**. Note that 'S', 'T', 'ST' represent social masking, temporal masking, social and temporal masking strategy, respectively.

Method	Pre-training strategy	Training steps			Fine-tuning result		
		Stage1	Stage2	Stage3	minADE	minFDE	MR
Continual Learning	Po	120k	-	-	0.656	1.069	0.107
	Pa	-	120k	-			
	B	-	-	120k			
Multi-task Learning	Po		120k		0.685	1.081	0.112
	Pa		120k				
	B		120k				
Continual Pre-training	Po	60k	30k	30k	**0.627**	**1.033**	**0.102**
	Pa	-	90k	30k			
	B	-	-	120k			

Table 3: **Continual Pre-training for map reconstruction**. Note that 'Po', 'Pa', 'B' represent point masking, patch masking, block masking strategy, respectively.

Num	Modules	minADE	minFDE	MR
E1	baseline	0.730	1.100	0.120
E2	E1 + map encoder	0.732	1.096	0.119
E3	E2 + trajectory pre-train	0.621	1.027	0.099
E4	E3 + map pre-train	**0.604**	**1.003**	**0.092**

Table 4: **Ablation on the effectiveness of our modules**.

4. Experimental Results

Argoverse validation set. In Table 1, we verify our method on the Argoverse validation set and demonstrate superior performance, achieving the lowest minADE score of 0.60 compared to other approaches. Furthermore, in comparison to our baseline model, our Traj-MAE exhibits a noteworthy reduction in minFDE and MR from 1.10 to 1.00 (9%) and 0.12 to 0.09 (25%), respectively.

Continue Pre-training. To investigate the properties of our proposed continual pre-training, we compare it with continual learning and multi-task learning. As we can see from Table 2 and Table 3, all method train each strategy with the same steps for a fair comparison. Our proposed continual pre-training achieves the best fine-tuning results. Moreover, multi-task learning, which learns multiple strategies at the same time, achieves little improvement compared to our baseline model. This could be that reconstructing with so many strategies simultaneously is too hard for the model to learn, thus even hurting the model's representation ability. We hope that future work will explore the different novel designs of continual learning and multi-task learning to make them suitable for trajectory prediction.

Module Analysis. Table 4 shows the effectiveness of different modules of our Traj-MAE. First, we find that adding a map encoder directly to process the HD map brings little improvement to Autobots, and even hurts the model's performance on minADE. Pre-training the trajectory encoder can improve the accuracy significantly, especially on minADE. What's more, the accuracy can be further improved when we pre-train the map encoder on the basis of $E3$. Thus, the validity of our proposed trajectory pre-training module and map pre-training module can be proved.

5. Visualization

We show several examples of reconstruction in Figure 1 and Figure 2. The motion forecasting results on the Argoverse validation set are shown in Figure 3. Samples are all chosen from the Argoverse validation set.

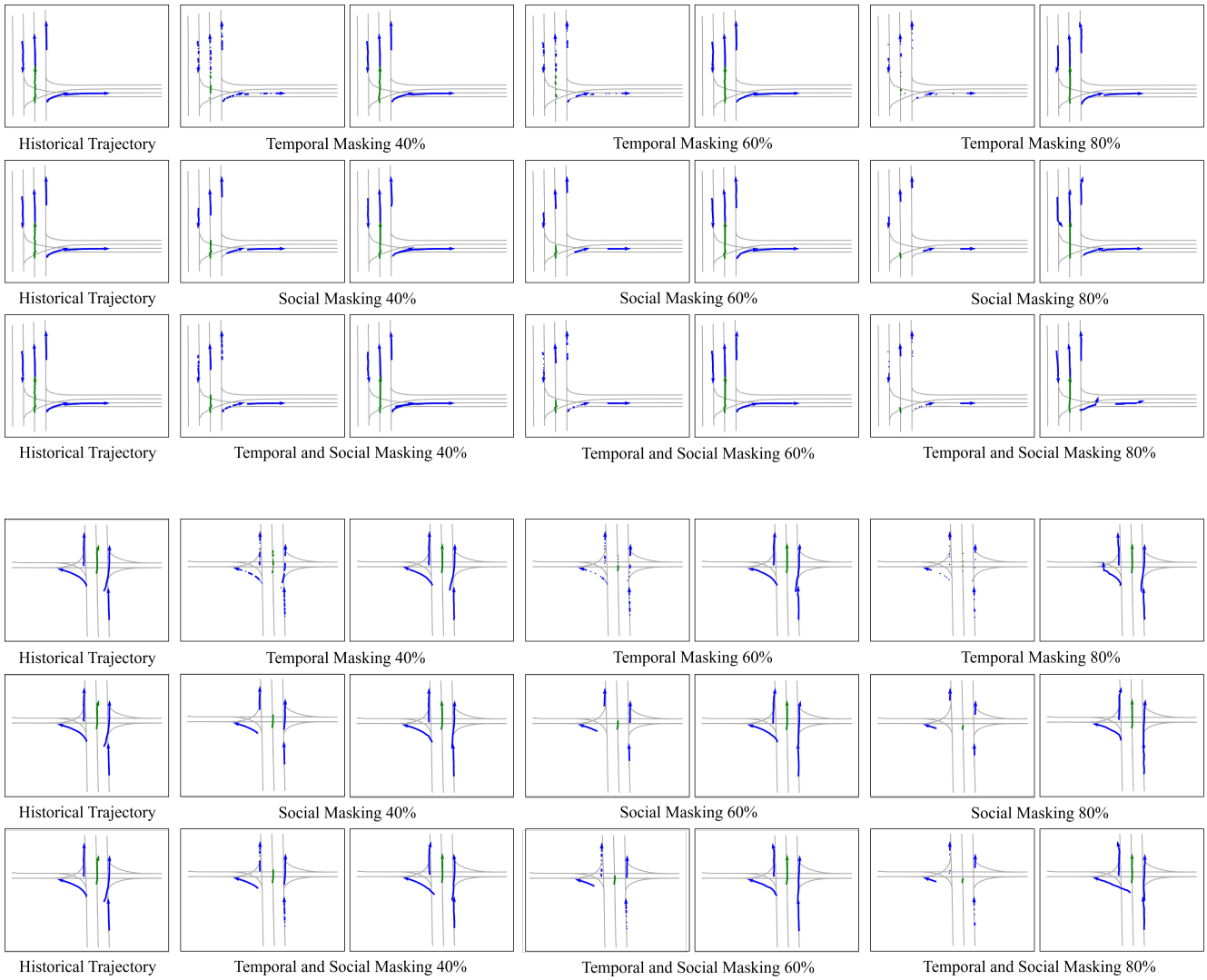
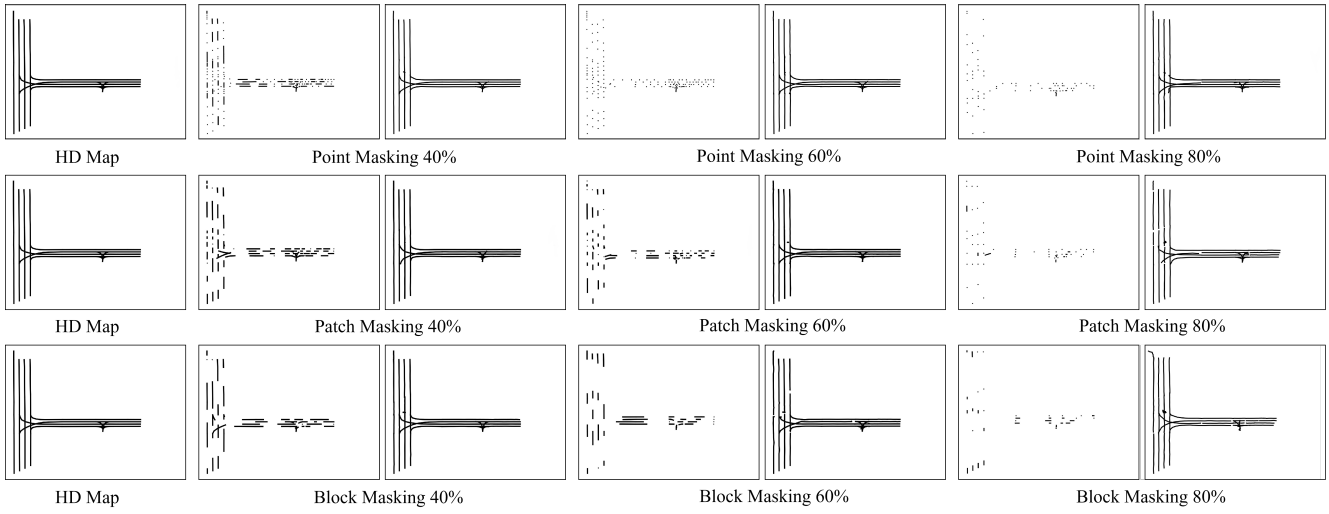


Figure 1: **Reconstruction results on historical trajectory.**



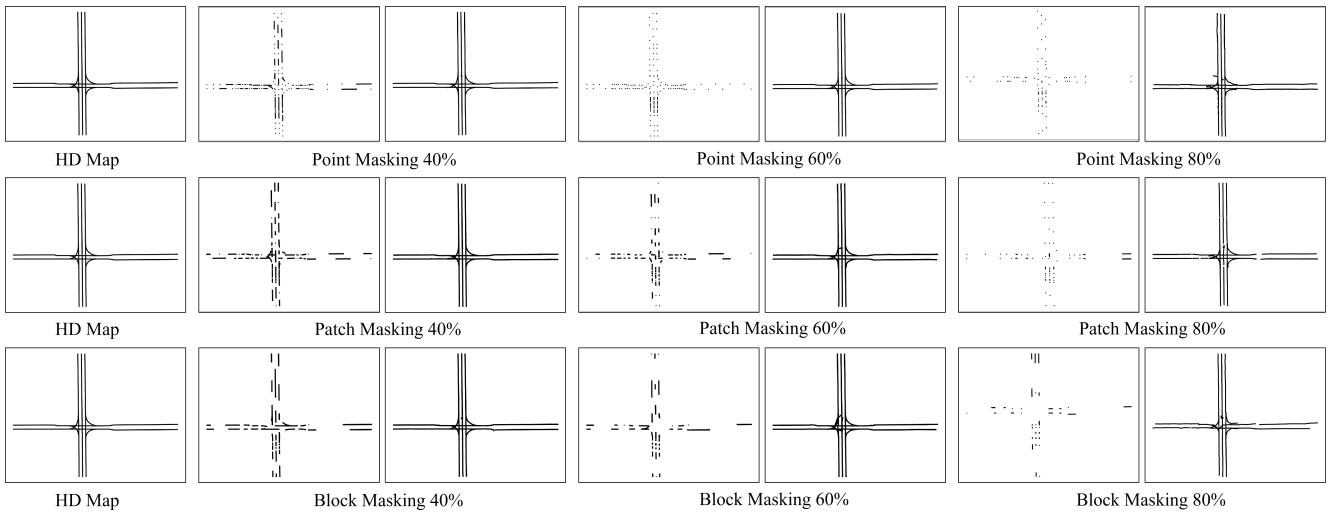


Figure 2: **Reconstruction results on HD map.**

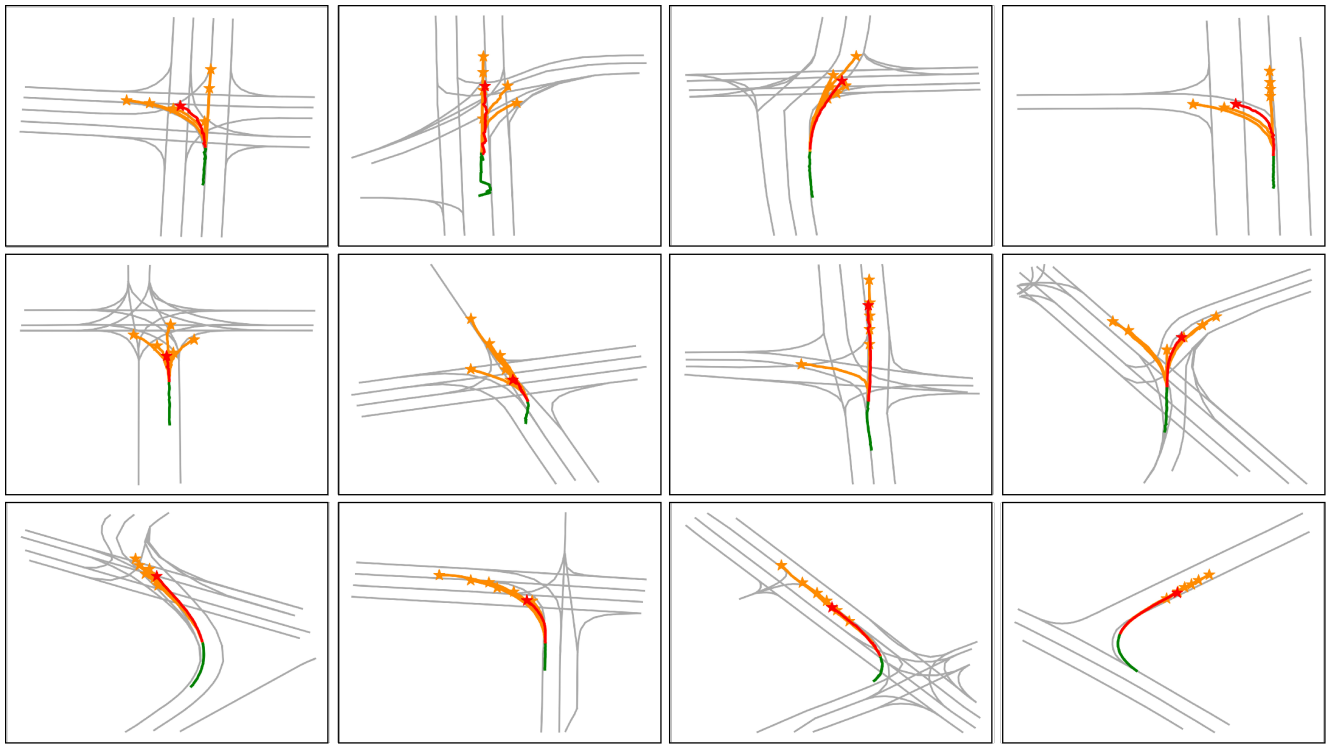


Figure 3: **The motion forecasting results on the Argoverse validation set.** The historical trajectory of the target agent is in green, predicted trajectories in orange and ground truth in red, respectively.

References

[1] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *European Conference on Computer Vision*, pages 624–641. Springer, 2020. 1

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

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 1

[4] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *International Conference on Learning Representations*, 2022. 1

[5] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. 1

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[7] 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

[8] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021. 1

[9] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018. 1

[10] Mingkun Wang, Xinge Zhu, Changqian Yu, Wei Li, Yuexin Ma, Ruochun Jin, Xiaoguang Ren, Dongchun Ren, Mingxu Wang, and Wenjing Yang. Ganet: Goal area network for motion forecasting. *arXiv preprint arXiv:2209.09723*, 2022. 1

[11] Maosheng Ye, Tongyi Cao, and Qifeng Chen. Tpcn: Temporal point cloud networks for motion forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11318–11327, 2021. 1

[12] Maosheng Ye, Jiamiao Xu, Xunnong Xu, Tongyi Cao, and Qifeng Chen. Dcms: Motion forecasting with dual consistency and multi-pseudo-target supervision. *arXiv preprint arXiv:2204.05859*, 2022. 1

[13] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 532–539. IEEE, 2021. 1

[14] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019. 1

[15] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *arXiv preprint arXiv:2008.08294*, 2020. 1

[16] 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*, pages 8823–8833, 2022. 1