Traj-MAE: Masked Autoencoders for Trajectory Prediction

Hao Chen1*  Jiaze Wang1*  Kun Shao4  Furui Liu2  Jianye Hao4,6  Chenyong Guan5
Guangyong Chen2†  Pheng-Ann Heng1,3

1 Department of Computer Science and Engineering, The Chinese University of Hong Kong
2 Zhejiang Lab  3 Institute of Medical Intelligence and XR, The Chinese University of Hong Kong
4 Huawei Noah’s Ark Lab  5 Gudsen Technology Co. Ltd  6 Tianjin University

Abstract

Trajectory prediction has been a crucial task in building a reliable autonomous driving system by anticipating possible dangers. One key issue is to generate consistent trajectory predictions without colliding. To overcome the challenge, we propose an efficient masked autoencoder for trajectory prediction (Traj-MAE) that better represents the complicated behaviors of agents in the driving environment. Specifically, our Traj-MAE employs diverse masking strategies to pre-train the trajectory encoder and map encoder, allowing for the capture of social and temporal information among agents while leveraging the effect of environment from multiple granularities. To address the catastrophic forgetting problem that arises when pre-training the network with multiple masking strategies, we introduce a continual pre-training framework, which can help Traj-MAE learn valuable and diverse information from various strategies efficiently. Our experimental results in both multi-agent and single-agent settings demonstrate that Traj-MAE achieves competitive results with state-of-the-art methods and significantly outperforms our baseline model. Project page: https://jiazewang.com/projects/trajmae.html.

1. Introduction

The goal of trajectory prediction is to predict the future trajectories of moving agents (e.g., pedestrians and vehicles), which is a crucial problem for building a safe, comfortable, and reliable autonomous driving system [32, 66, 37, 12, 51]. Many promising works [21, 7, 49, 26, 70, 60] have been proposed with great interest and demand from academia and industry. It has been demonstrated that modeling complex interactions between agents [47, 44, 46, 6, 27] is of great importance in trajectory prediction. On this basis, to address the colliding prediction problem and generate consistent trajectory predictions, it is essential to model social and temporal relations between agents and to have a global understanding of maps [2]. In this paper, we investigate this issue using self-supervised learning.

*Equal contribution
† Corresponding author: gychen@zhejianglab.com
Self-supervised learning aims to learn latent semantics from unlabeled data rather than building representations based on human annotations. Recent years have witnessed noteworthy advancements in the application of self-supervised learning to natural language processing [14, 62] and computer vision [57, 38, 4]. One of the most promising self-supervised methods is the masked autoencoders (MAE) [23] which achieve success in various tasks [39, 53, 73, 72, 22]. Furthermore, pre-train and fine-tune on the same small-scale datasets are also essential to learn a good representation [15]. Inspired by these works, we aim to explore the complex interactions between agents and the multiple granularities of maps using masked autoencoders. How to design an efficient masked autoencoder to generate consistent trajectory predictions? We attempt to answer the question from the following perspectives:

(i) The information density of trajectory and high definition (HD) maps differs significantly from that of images. While images are natural signals with high spatial redundancy, trajectories represent continuous temporal sequential signals with complex social interactions between agents, and HD maps contain highly structured information. Given the differences, models aimed at trajectory prediction require corresponding adjustments to capture informative features. Therefore, we investigate various masking strategies and suitable masking ratios for trajectories and HD maps. We develop both social and temporal masking to enable the trajectory encoder to capture information from diverse perspectives. We also study multiple granularities masking to enforce the map encoder to capture structural information from HD maps. Furthermore, we find that regardless of the masking strategy adopted, a high masking ratio (50% ~ 60%) yields favorable results, which demands the encoders to acquire a holistic understanding of historical trajectories and HD maps.

(ii) The absence of an efficient framework for pre-training multiple strategies poses a challenge for effective multimodal trajectory prediction. While traditional multi-task learning from scratch [74] may struggle to converge due to the complex nature of this task, traditional continual learning methods [11, 40] are limited by their inability to train the network with multiple tasks without forgetting previously learned knowledge. To address this issue, we propose a novel approach that trains the new strategy simultaneously with the original strategies, utilizing previously learned parameters to initialize the network. Therefore, we ensure that our network can acquire new knowledge while retaining previously obtained knowledge.

Driven by the analysis, we present Masked Trajectory Autoencoder (Traj-MAE), an efficient and practical framework for self-supervised trajectory prediction. As depicted in Figure 1, Traj-MAE leverages partial masking of the input trajectory and HD map, utilizing the trajectory encoder and map encoder to reconstruct the masked segments, respectively. Through employing diverse masking strategies to reconstruct missing parts of the input trajectory and HD map, the trajectory encoder and map encoder can acquire a comprehensive understanding of the latent semantics of the inputs from various perspectives. Moreover, we introduce a novel continual pre-training framework, which is a highly-efficient learning approach that trains the model with multiple strategies simultaneously, mitigating the issue of catastrophic forgetting.

Our core contributions are as follows:

- To our best knowledge, we are the first to present a neat and efficient masked trajectory autoencoder for self-supervised trajectory prediction.
- We explore different masking strategies which fully utilize MAE to exploit the latent semantics of historical trajectory and HD map. Meanwhile, a continual pre-training framework is proposed to efficiently train the model with multiple strategies.
- We conduct extensive experiments on the Argoverse, and INTERACTION for autonomous driving trajectory prediction, and the synthetic partition of TrajNet++ for pedestrian trajectory prediction. Our Traj-MAE achieves competitive results on these benchmarks and outperforms our baseline model by a notable margin.

2. Related Works

Trajectory Prediction is widely considered a sequence modeling task with many RNN-based methods [1, 71, 34, 8] proposed to model the trajectory pattern of agents’ future locations, as RNN (e.g., LSTM [24]) have achieved remarkable success in sequence modeling. Due to the strong ability of Transformers [55, 59] to capture long-range dependencies, many transformer-based methods have emerged and flourished. STAR [65] is proposed to capture complex spatio-temporal interactions by interleaving between spatial and temporal Transformers. mmTransformer [32] is designed to hierarchically aggregate the past trajectories, the road information, and the social interaction. For predicting multi-agents future trajectories, AgentFormer [66] and AutoBots [20] have given solutions to model the time dimension and social dimension simultaneously. The enhancement of the encoder’s ability to model information in both dimensions is an interesting and central focus of this work.

Self-supervised Learning has shown significant success in natural language processing and computer vision fields recently, especially the autoencoding method. Denoising autoencoders (DAE) [56] is a learning representation method that reconstructs original signals from corrupted inputs. BERT [14] can be seen as a development of DAE, which masks input tokens and trains the model to predict the miss-
Traj-MAE is mainly composed of three stages: (a) Trajectory encoder pre-train stage with continual trajectory masking and reconstruction strategies. (b) Map encoder pre-train stage with continual map masking and reconstruction strategies. (c) Fine-tune and inference stage where the encoders are initialized by the pre-trained models’ parameters.

Continual Learning is a method to tackle the catastrophic forgetting problem that happens in sequentially learning samples of different input patterns. The methods can be roughly categorized into replay, regularization-based, and parameter isolation approaches [13]. With respect to replay methods [42, 43, 25, 10, 52, 31], previous task samples are replayed while learning a new task to alleviate forgetting. Instead, when learning new data, regularization-based methods [50, 41, 68, 29, 61] often introduce a regularization term in the loss function to consolidate previous knowledge. Parameter isolation methods [33, 48] dedicate different model parameters to each task to prevent any possible forgetting. In this work, we propose a continual pre-training framework to tackle the forgetting problem, in which way we are able to improve the generalization of model encoders by leveraging the specific information contained in the training samples of related masking strategies.

3. Approach

Our Traj-MAE is a sophisticated yet efficient self-supervised approach. Figure 2 provides an overview of the Traj-MAE framework. In this section, we begin by introducing our network backbone. We then delve into our analysis of the masking strategies for trajectory and HD-map reconstruction. Finally, we discuss how we incorporate Traj-MAE into our continual pre-training framework.

3.1. Network Backbone

In this work, We use Autobots [20] that has a transformer encoder-decoder architecture (detailed in supplementary material) as the baseline model to verify the effectiveness of the proposed method. Our Traj-MAE masks random parts from the input trajectory and HD map, then reconstructs the missing parts respectively. Following MAE [23] and Video-MAE [53], we adopt the asymmetric encoder-decoder design to reduce computation.

Traj-MAE Encoder. In Autobots, historical trajectories are encoded into context tensors, together with learnable seed parameters and map context, are passed to the decoder to predict future trajectories. Inspired by this design, we adopt the Autobots encoder as our trajectory encoder. However, in Autobots, the HD map is directly fed to the decoder, making it challenging for the model to capture the inherent information of the HD map. To address this limitation, we introduce a map encoder with a similar architecture to the trajectory encoder to better reconstruct the masked HD map. However, we observed that directly adding the map encoder to the Autobots results in little improvement (see supplementary material). Nevertheless, we found that pre-training the map encoder with our proposed masking and reconstruction strategy can further improve the accuracy, validating the effectiveness of our pre-training strategy.

Traj-MAE Decoder. The encoder in Traj-MAE processes only the unmasked parts of the input, while the decoder reconstructs the missing parts from the latent representation and mask tokens. Mask tokens are shared vectors that indi-
cate the presence of the missing parts that need to be predicted. Additionally, positional embeddings are added to all tokens to provide location information. Traj-MAE decoder is designed with Transformer blocks that are shallower than the encoder and are used solely during pre-training to perform the trajectory and map reconstruction strategies. This enables the decoder architecture to be flexible and independent of the encoder architecture. Pre-training with a lightweight decoder can notably reduce pre-training time.

### 3.2. Masking Strategy

Different masking strategies determine the pretext task with different latent information that the network encoder can learn. To capture the social and temporal information from historical trajectories and multiple granularity representations from HD maps, we devise three masking strategies for the trajectory encoder and map encoder, respectively. Each strategy masks different scales and components of the input, with the goal of reconstructing the missing parts of the input.

**Trajectory Masking Strategy.** We introduce three effective masking and reconstruction strategies to enhance representation learned by the trajectory encoder. The three different strategies are illustrated in Figure 3.

- **Social Masking.** Understanding the social relationships between agents is a fundamental concern when predicting trajectories. Social masking aims to reconstruct each agent’s trajectory from surrounding agents. We mask the ego-agent’s trajectory in the last observed time and nearby agents’ trajectories at the beginning of the observed time.

- **Temporal Masking.** Temporal masking strategy endeavors to reconstruct the trajectories of all agents that have been randomly masked in the time domain. By inferring the positions of agents at various temporal intervals based on their positions at specific times, our model is able to efficiently capture the temporal dynamics of historical trajectories.

- **Social and Temporal Masking.** We also construct a masking strategy that reconstructs the historical trajectories in both temporal and social aspects. Specifically, half of the trajectories of surrounding agents are randomly masked in the time dimension, and half of that are masked in the last observed time, while the trajectory of the ego-agent in the last observed time is masked. By reconstructing these trajectories simultaneously, the social and temporal masking can further enhance the trajectory encoder to obtain the temporal and social information.

**Map Masking Strategy.** The input vector of the map encoder is based on the VectorNet approach [17], which selects a starting point and direction, and uniformly samples key points from the splines at the same spatial distance. To capture the latent semantics of the HD map, we propose a mask and reconstruction strategy that operates at multiple granularities.

- **Point Masking.** Our point masking strategy randomly samples and masks key points from the input map vector.
**Reconstruction Targets.** A single continuous line segment.

Consecutive line segments, whereas block masking only masks the polyline in HD map, patch masking masks several consecutive line segments and patch masking differ in their granularities. For each strategy, the map encoder must infer the map architecture from longer distance points, allowing it to learn the long-term distance relation of the map.

**Block Masking.** The block masking approach removes large blocks from the input and predicts the missing parts in the input HD map, enabling the map encoder to have a better global understanding of the whole map. Block masking and patch masking differ in their granularities. For each polyline in HD map, patch masking masking several consecutive line segments, whereas block masking only masks a single continuous line segment.

**Reconstruction Targets.** Traj-MAE reconstructs the input by predicting the coordinates of location points in every masked part. Our loss function comprises the Huber loss between the reconstructed trajectory and original trajectory to pre-train the trajectory encoder, and the Huber loss between the reconstructed HD map and original HD map to pre-train the map encoder. Similar to MAE [23] and PointMAE [39], we compute the loss only on masked parts.

### 3.3. Continual Pre-training Framework

We propose a continual pre-training framework to learn diverse information from multiple masking strategies efficiently.

Traditional continual learning method [11, 40] trains the model on only one strategy at each stage, and the model may suffer from the catastrophic forgetting problem, forgetting the previously learned knowledge. Multi-task learning method [74] trains the model with multiple tasks at the same time. In this way, we find it hard for our model to converge after many steps, and the performance could be even worse than the model pre-trained on a single strategy.

Therefore, there are two issues to solve. The first issue is to learn the strategies without forgetting the previous knowledge. The second issue is how to pre-train the model with multiple strategies efficiently. To overcome these issues, we propose a practical continual pre-training framework that enables model training with satisfactory efficiency and alleviates catastrophic forgetting. The core idea is to train the model over multiple stages with cross-stage parameter sharing. That means, for each pre-training stage except the first one, we use the parameters learned in previous stage to initialize the model. Then we pre-train a new strategy together with the previous strategies simultaneously in this stage. As shown in Figure 4, the number of pre-training stages is equal to that of strategies. Our framework allocates each strategy a fixed number of pre-training steps ($N$) and distributes the steps for each strategy over the whole pre-training stages. $M$ is the steps to pre-train the previous strategy in the new stage. To pre-train different strategies in a single stage, we randomly select training samples from each strategy. In this way, the effectiveness of our continual pre-training framework can be guaranteed.

### 4. Experiments

#### 4.1. Experimental Setup

**Datasets.** Argoverse motion forecasting dataset [9] provides 333K real-world driving sequences, which are sampled at 10Hz, with 2 seconds history and 3 seconds future. The whole dataset is split into train, validation and test sets, with 205942, 39472, and 78143 sequences, respectively. The Interaction dataset [69] consists of various highly interactive driving situations, and each trajectory has 1 second history and 3 seconds future sampled at 10Hz. In the multi-agent prediction track, the target is to predict multiple target agents’ coordinates and yaw jointly. The synthetic partition of the TrajNet++ dataset [45] has 54513 scenes, which

### Table 1: Comparison with state-of-the-art methods on the Argoverse test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>minADE</th>
<th>minFDE</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNT [75]</td>
<td>0.94</td>
<td>1.54</td>
<td>0.13</td>
</tr>
<tr>
<td>LaneRCNN [67]</td>
<td>0.90</td>
<td>1.45</td>
<td>0.12</td>
</tr>
<tr>
<td>mmTransformer [32]</td>
<td>0.84</td>
<td>1.34</td>
<td>0.15</td>
</tr>
<tr>
<td>GOHOME [19]</td>
<td>0.94</td>
<td>1.45</td>
<td>0.11</td>
</tr>
<tr>
<td>TPCN [63]</td>
<td>0.87</td>
<td>1.38</td>
<td>0.16</td>
</tr>
<tr>
<td>Scene Transformer [37]</td>
<td>0.80</td>
<td>1.23</td>
<td>0.13</td>
</tr>
<tr>
<td>MultiPath++ [54]</td>
<td>0.79</td>
<td>1.21</td>
<td>0.13</td>
</tr>
<tr>
<td>DenseTNT [21]</td>
<td>0.88</td>
<td>1.28</td>
<td>0.13</td>
</tr>
<tr>
<td>HiVT [76]</td>
<td>0.77</td>
<td>1.17</td>
<td>0.13</td>
</tr>
<tr>
<td>Wayformer [36]</td>
<td>0.77</td>
<td>1.16</td>
<td>0.12</td>
</tr>
<tr>
<td>DCMS [64]</td>
<td>0.77</td>
<td>1.14</td>
<td>0.11</td>
</tr>
<tr>
<td>GANet [60]</td>
<td>0.81</td>
<td>1.16</td>
<td>0.12</td>
</tr>
<tr>
<td>DSP [70]</td>
<td>0.82</td>
<td>1.22</td>
<td>0.13</td>
</tr>
<tr>
<td>Autobot-Ego [20]</td>
<td>0.89</td>
<td>1.41</td>
<td>0.16</td>
</tr>
<tr>
<td>Traj-MAE</td>
<td>0.81 ↓ 9%</td>
<td>1.25 ↓ 11%</td>
<td>0.137 ↓ 14%</td>
</tr>
</tbody>
</table>

Figure 4: Illustration of our continual pre-training framework. For each strategy, we guarantee it has the same training steps throughout the pre-training process.

and reconstructs the whole map by predicting the missing points. This fine-grained learning approach is shown to have the best effect in our experiments.

**Patch Masking.** Patch masking refers to masking the map vector at the patch level, where patches are randomly sampled from the map vector. Patch masking and reconstruction are more complex than point masking due to the unknown surrounding points of the middle point. Consequently, the map encoder must infer the map architecture from longer distance points, allowing it to learn the long-term distance relation of the map.
4.2. Experimental Results.

In this subsection, we pre-train and fine-tune the model on the same benchmark to validate the benefit brought by our proposed self-supervised learning method. We forecast 6 future trajectories on all datasets. The meaning of the used metrics is presented in the supplementary material and the lower metrics indicate better performance.

Results on Argoverse. The trajectory prediction results on Argoverse are reported in Table 1. On the Argoverse test set, our Traj-MAE decreases the minADE by 9%, minFDE by 11% and MR by 14%, respectively. Although Traj-MAE does not achieve state-of-the-art, the performance demonstrates that our Traj-MAE significantly improves our baseline model (AutoBot-Ego). Moreover, it uses much less computation, which is an effective and GPU-friendly pre-training mechanism that only pre-trained on a single GPU (V100) in under 48h. Additionally, the performance improvement achieved by Traj-MAE is also demonstrated on the validation set, as shown in 4.3.

Results on INTERACTION. In Table 2, we evaluate our method on the INTERACTION multi-agent track and achieve state-of-the-art performance on this benchmark.
Table 4: Ablation study on continual pre-training framework. We show the results using different pre-train strategies with their best masking ratio. Note that 'S', 'T', 'ST' represent social masking, temporal masking, social and temporal masking strategy, respectively. 'Po', 'Pa', 'B' represent point masking, patch masking, block masking strategy, respectively. The table shows that Traj-MAE outperforms all other approaches in terms of Consistent MinJointMR, the ranking metric. This metric encourages the model to make consistent predictions, thus the best result on this metric indicates that our Traj-MAE can well capture multi-agents’ interaction. As for other metrics, our model still has competitive results. Especially, the reductions of 51% and 40% in CrossCollisionRate and EgoCollisionRate demonstrate that our Traj-MAE can better utilize social information and avoid collisions between agents.

4.3. Ablation Studies.

To investigate the properties of our method, we perform in-depth ablation studies on the Argoverse validation set. For these experiments, all models share the same experiment settings and architecture. We evaluate our models on minADE, minFDE, and miss rate (MR) of the predicted trajectories. Our baseline model (Autobot-Ego with Map Encoder) achieves \(0.732, 1.096\) and \(0.119\) on minADE, minFDE, and MR, respectively. When fine-tuning the trajectory encoder with the pre-trained model, the map encoder is trained from scratch and vice versa.

Trajectory Masking Strategy. In Figure 5(a), we compare the performance of different trajectory masking strategies and ratios. As the masking ratio increases from 30% to a threshold of 50-60%, the performance of all three strategies improves simultaneously. However, the performance degrades as the masking ratio increases beyond the threshold to 80%. Furthermore, our experiments show that social and temporal masking outperforms the other two strategies, particularly in terms of minADE and minFDE. This suggests that incorporating social and temporal information is crucial for effective trajectory prediction. We attribute this to the fact that social and temporal masking encourages the...
5. Conclusion

This paper proposes a novel masked trajectory autoencoder (Traj-MAE) for self-supervised trajectory prediction learning. Our Traj-MAE incorporates diverse masking strategies to capture more essential temporal and social information, thereby making Traj-MAE a challenging self-supervised trajectory prediction framework.

Map Masking Strategy. Figure 5(b) shows the influence of masking strategies with different masking ratios for HD map. Point masking works best for pre-training our map encoder. Besides, point masking and patch masking allow for a higher masking ratio (60%) compared to block masking (50%), which can provide a more significant speedup benefit. We suppose that with the increase of connected masked parts, the mask and reconstruct task is more challenging, and the optimal mask ratio is smaller.

Continual Pre-training Strategy. Table 4 shows the effectiveness of our Traj-MAE. First, we find that our continual pre-training framework further improves the network performance than pre-training on a single strategy. With continual pre-training, the trajectory encoder is better equipped to capture social and temporal information, while the map encoder excels at capturing structured information from multiple granularities. Moreover, our experiments reveal that the sequence of pre-training strategies can also impact model performance. For trajectory encoder pre-training, the best sequence is social masking → temporal masking → social and temporal masking. For map encoder pre-training, the best sequence is point masking → patch masking → block masking. By integrating our pre-trained trajectory and map encoders, we achieve impressive results on the Argoverse validation set, with a \text{minADE} of 0.604, a \text{minFDE} of 1.003, and an \text{MR} of 0.092.

Qualitative Analysis. We visualize the reconstruction results on the Argoverse validation set for qualitative analysis, where Figure 6 illustrates the different masking strategies. Although social masking is challenging for trajectory reconstruction and block masking is difficult for map reconstruction, adopting a moderate masking ratio with these strategies can still yield satisfactory reconstruction results. We compare the different temporal masking ratios and point masking ratios in Figure 7. Our Traj-MAE can produce satisfying reconstructed trajectories and HD maps even under a high masking ratio (e.g., 80%), which indicates that our Traj-MAE can learn useful high-level representations. Furthermore, in Figure 8, we present two examples to demonstrate how Traj-MAE can effectively reduce the occurrence of collisions. Specifically, in a scene where our baseline model predicts a collision, Traj-MAE leverages the interaction relations among agents’ historical trajectories to avoid collision happening.
strategies that facilitate the trajectory encoder learning the social and temporal information and map encoder capturing structural information with multiple granularities. We also propose a continual pre-training framework that enables efficient pre-training of multiple strategies. Experimental results show that our Traj-MAE produces impressive results on various challenging datasets in both multiple-agent and single-agent settings. We hope that this work will inspire further investigation into self-supervised learning for trajectory prediction.

Acknowledgement

This work was supported by the National Key R&D Program of China (2022YFE0200700), the National Natural Science Foundation of China (Project No. 62006219), the Natural Science Foundation of Guangdong Province (2022A1515011579), in part by the InnoHK Clusters via Hong Kong Center for Logistics Robotics and by the Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems.

References


[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16000–16009, 2022. 


[57] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. Advances in Neural Information Processing Systems, 32, 2019. 2


