Forecast-MAE: Self-supervised Pre-training for Motion Forecasting with Masked Autoencoders

Jie Cheng[1] Xiaodong Mei[1] Ming Liu[1,2*]
HKUST[1] HKUST(GZ)[2]
{jchengai, xmeiab}@connect.ust.hk, eelium@ust.hk

Abstract

This study explores the application of self-supervised learning (SSL) to the task of motion forecasting, an area that has not yet been extensively investigated despite the widespread success of SSL in computer vision and natural language processing. To address this gap, we introduce Forecast-MAE, an extension of the mask autoencoders framework that is specifically designed for self-supervised learning of the motion forecasting task. Our approach includes a novel masking strategy that leverages the strong interconnections between agents' trajectories and road networks, involving complementary masking of agents' future or history trajectories and random masking of lane segments. Our experiments on the challenging Argoverse 2 motion forecasting benchmark show that Forecast-MAE, which utilizes standard Transformer blocks with minimal inductive bias, achieves competitive performance compared to state-of-the-art methods that rely on supervised learning and sophisticated designs. Moreover, it outperforms the previous self-supervised learning method by a significant margin. Code is available at https://github.com/jchengai/forecast-mae.

1. Introduction

Motion forecasting is a rapidly developing research field that plays a critical role in advanced autonomous driving systems [22]. This task involves predicting the future trajectories of other vehicles and pedestrians, while taking into account the intricate interactions and road layouts. The inherent multi-modal driving behaviors of agents, combined with diverse road networks, make motion forecasting an especially challenging undertaking.

Self-supervised learning (SSL) is an innovative approach that enables the acquisition of valuable latent features from unlabelled data. By pre-training the model on pretext tasks and pseudo-labels derived from the data, and subsequently fine-tuning on downstream tasks, SSL has demonstrated an ability to learn more extensive and adaptable latent features, leading to remarkable advancements in computer vision [3] and natural language processing (NLP) [10]. Nevertheless, despite its widespread popularity and success, there remains a notable lack of exploration of SSL in the motion forecasting domain. We have identified two principal challenges associated with integrating SSL into motion forecasting:

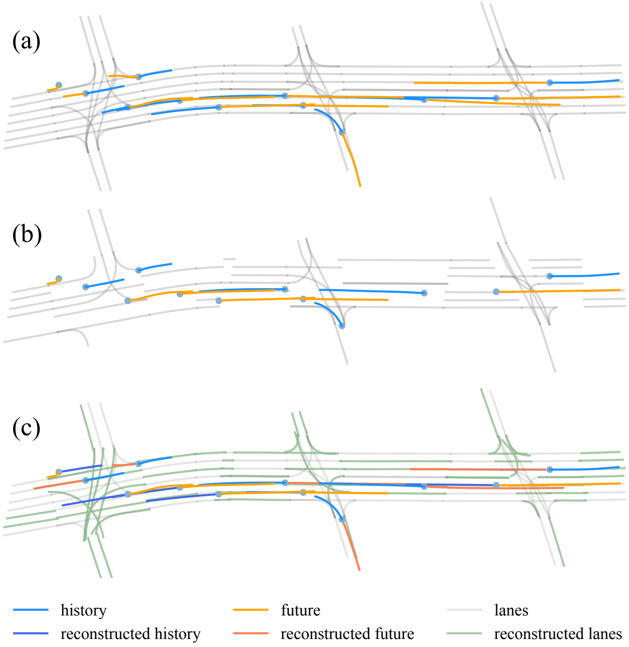

Figure 1. Reconstruction result on Argoverse 2 *validation* scenario. (a) The origin scenario. (b) 50% of agents' trajectory is masked using a complementary masking strategy (either history or future is masked). 50% of the lane segments are masked randomly. (c) Scenario reconstructed by the proposed Forecast-MAE.

(i) Motion forecasting pre-training requires annotated data, which sets it apart from fields such as computer vision and NLP where unlabeled raw inputs are easily accessible. In motion forecasting, we rely on annotated tracking sequences and hand-crafted high-definition maps that are typically collected by expensive onboard sensors and

[*] Corresponding author: *Ming Liu*

require human annotation labor [5, 43, 12]. This poses a challenge to scaling up self-supervised pre-training, a key aspect of SSL's success. To address this challenge, very recent work PreTraM [44] proposed generating additional rasterized map patches (28.8M) cropped from local regions of the entire HD map to train a robust map encoder with contrastive learning. Although this approach yielded notable performance improvements compared to the baseline, it is limited to models based on rasterized map representations, which have a significant performance gap compared to more recent vector-based or graph-based models. However, another pioneering work, SSL-Lanes [1], demonstrated that carefully designed pretext tasks can significantly enhance performance without using extra data by learning richer features. In this paper, we follow this approach to learn better and more generalized features using the existing dataset.

(ii) The task of motion forecasting involves incorporating multiple modal inputs, such as static map features, spatiotemporal agent motion features, and semantic scene contexts [39, 26, 13, 4, 36, 50, 29, 9, 52]. While various self-supervised learning methods have proven successful in dealing with single-modal inputs such as the image [3], text [10], or point cloud [47, 32], developing pretexts that establish cross-modal interconnections is not an easy task. SSL-Lanes concentrated on designing pretext tasks for each specific input modality, such as lane node masking or agent maneuver classification. Nevertheless, they did not explore the combination of these different tasks or develop pretext tasks that explicitly involve multiple modal inputs. Authors of PreTraM drew inspiration from CLIP's [35] cross-modal contrastive learning framework involving text and images. They devised a technique for pre-training map and trajectory encoders by pairing batches of (map, trajectory) training instances. Nevertheless, their approach merely encompasses the history trajectory-map connection, thereby restricting the scope of modality interconnections to a particular type. This study confronts this challenge by utilizing a masked autoencoder framework that can assimilate all cross-modal interdependencies within a unified scene reconstruction task.

The masked autoencoder (MAE) [20] has garnered significant attention due to its recent achievements in image-based self-supervised learning. This approach involves masking a portion of the input data and reconstructing the missing part using an autoencoder structure. The effectiveness of MAE has also been demonstrated in other domains, such as audio [21] and point cloud [32]. An intriguing question arises: *can we extend MAE to motion forecasting?* Indeed, motion forecasting itself can be viewed as a masking and reconstructing task, wherein the future trajectory of agents is masked and predicted. Based on the strong correlation between agents' historical and future trajectories and road networks, we further extend this concept to the entire scene reconstruction. Specifically, we mask agents' history trajectory or future trajectory in a complementary manner (*i.e.* either history or future is masked), and randomly mask non-overlapping lane segments, shown in Figure 1. This masking scheme offers several advantages. Firstly, the model must learn how to reconstruct the future from past motion and, in turn, infer history from the future, with limited access to lane structures. This pretext task allows the model to establish a robust bidirectional relationship between past and future motion. Secondly, the model learns to reconstruct lane segments by jointly utilizing neighboring visible lanes, agents' history and future trajectories, thereby establishing a more profound cross-modal understanding.

To this end, we introduce Forecast-MAE, an extension of the masked autoencoder framework specifically designed for self-supervised learning of the motion forecasting task. Our methodology comprises a novel masking design that exploits the strong interdependencies among all agents' trajectories and road networks. Despite being simple and incorporating minimal inductive bias, our proposed Forecast-MAE performs strongly on the challenging Argoverse 2 (AV2) motion forecasting benchmark [43] and significantly outperforms the previous self-supervised learning method.

Our contribution can be summarized as follows:

- To our best knowledge, we propose the first masked autoencoding framework for self-supervised learning on the motion forecasting task. Without extra data or pseudo-labels, our method greatly improves the performance of motion forecasting through pre-training compared to training from scratch.
- We introduce a straightforward yet highly effective masking scheme that facilitates the learning of bidirectional motion connections and cross-modal relationships within a single reconstruction pretext task.
- We show that our approach, based entirely on standard Transformers with minimal inductive bias, achieves competitive performance compared to the state-of-the-art with supervised learning on the challenging Argoverse 2 benchmark, and significantly outperforms the previous self-supervised learning method.
- Our findings suggest that SSL can be a promising approach for motion forecasting, and we anticipate that this may spark greater interest in the field.

2. Related Work

Motion Forecasting. The performance of motion forecasting models has significantly advanced in recent years, primarily attributable to the amplified interest in self-driving vehicles and the widespread availability of standard benchmarks. Herein, we concisely outline three key aspects contributing to its improvements.

(i) *Improvement on scene representation*. In the early stages, rasterized top-down semantic images are commonly utilized for scene representation, and off-the-shelf image encoders are used for learning [39, 33, 4, 15]. Although this image-based representation is simple and unified, it inevitably results in the loss of detailed structural information during rasterization. The popularity of vectorized representations has increased significantly with the introduction of VectorNet [13], owing to their higher representation capacity and significantly stronger performance. Furthermore, graphs [26, 48, 9, 16, 25] are widely used as another promising scene representation. TPCN [45], as a standalone approach, achieves impressive results by treating the agents' trajectories and lanes as the point cloud.

(ii) *Improvement on model architectures*. Early rasterized methods naturally relied on well-established convolutional networks. Later, inspired by the impressive performance of Transformer [41], attention mechanisms have been extensively used for interaction modeling and information aggregation, given their superior flexibility and efficacy. Some works [31, 29, 52, 17] have directly incorporated transformers for forecasting and achieved satisfactory outcomes. A more recent work, MTR [36], builds on cutting-edge vision object detection architecture DETR [2], resulting in state-of-the-art performance. Advances in the graph neural network (GNN) domain are also widely explored [48, 16, 25, 9, 6, 28, 7]. LaneGCN [26] modified graph convolutional operation tailed for lane graph encoding. HDGT [24] uses the heterogeneous graph to encode different types of agents and map elements. HiVT [52], QCNet [51] and [23] explores different corrdiantes systems.

(iii) *Introducing of prior knowledge*. Incorporating prior knowledge to tackle the complex problem of multi-modal future prediction has become increasingly prevalent in recent literature. Several works utilize predefined candidate trajectories [33, 37] or anchor points [4, 40] by clustering the ground truth or generating with planners. Another line of research involves sampling goals within the drivable areas and utilizing a two-stage prediction pipeline [48, 9, 50, 18, 15, 16]. DCMS [46] introduces temporal consistent constraints based on the assumption that predictions should not change abruptly. However, these methods typically require additional computation or have a higher model complexity.

Despite significant advancements in motion forecasting, there is a recent trend towards greater architectural complexity and utilization of prior knowledge. In this study, we explore a different direction for enhancing performance, namely self-supervised learning. By leveraging the simplicity of the MAE framework, we demonstrate that our proposed Forecast-MAE, employing a standard transformer architecture with minimal prior knowledge, can achieve competitive performance compared to state-of-the-art supervised learning-based methods with sophisticated designs.

Self-supervised Learning in Motion Forecasting. There are only a few studies that explore SSL in motion forecasting. To the best of our knowledge, VectorNet is the earliest work that incorporates a BERT-like [10] graph completion task to better capture interactions between agents and maps. However, it is a very preliminary attempt, and the graph completion is treated as an auxiliary training objective that is jointly optimized with the motion forecasting task. PreTraM and SSL-Lanes are two recent works that systematically study SSL. The authors of PreTraM believe that the scarcity of trajectory data restricts the application of SSL in motion forecasting. They generate additional local map patches from the entire maps and leverage single-modal and cross-modal contrastive learning to pretrain the map and trajectory encoders separately. In contrast, our method adopts a completely different MAE-based framework, where representations of different modalities are learned jointly. SSL-Lanes demonstrated that SSL could learn better latent features without using extra data. It studied four pretext tasks, each focusing on one specific input modality, such as lane masking or agents' maneuver classification. However, they do not explore combining these different tasks or designing pretext tasks involving multimodal inputs. On the contrary, the proposed Forecast-MAE learns cross-modal interconnections by design and outperforms SSL-Lanes by a large margin.

3. Methodology

We propose Forecast-MAE, a simple and neat MAE-based framework for self-supervised pre-training of the motion forecasting task. The pre-training process is illustrated in Figure 2. Visible agents' history/future trajectories and lane segments are embedded as tokens and then processed with a standard transformer encoder. Following the asymmetric design of the vision MAE [20], different mask tokens are added to the decoder's input sequence and later used to reconstruct the masked trajectories and lane segments with simple prediction heads.

3.1. Masking

In contrast to all current self-supervised learning frameworks for motion forecasting, we utilize the future trajectories of agents as an additional input for pre-training. Our experiments reveal that masking future trajectories is a crucial aspect for Forecast-MAE to be effective. To begin, the road maps are initially segmented into non-overlapping lane sections. We then randomly mask a subset of lane segments according to a uniform distribution. The masking technique for agents differs slightly. Although random masking is still employed for agent trajectories, we only mask the history or future trajectory of each agent (*e.g.*, 40% of agents retain their history, while the remaining 60% retain their future). We refer to this process as *complementary random masking*. This constraint is sensible, as reconstructing trajectories from a single pose is not a meaningful pretext task

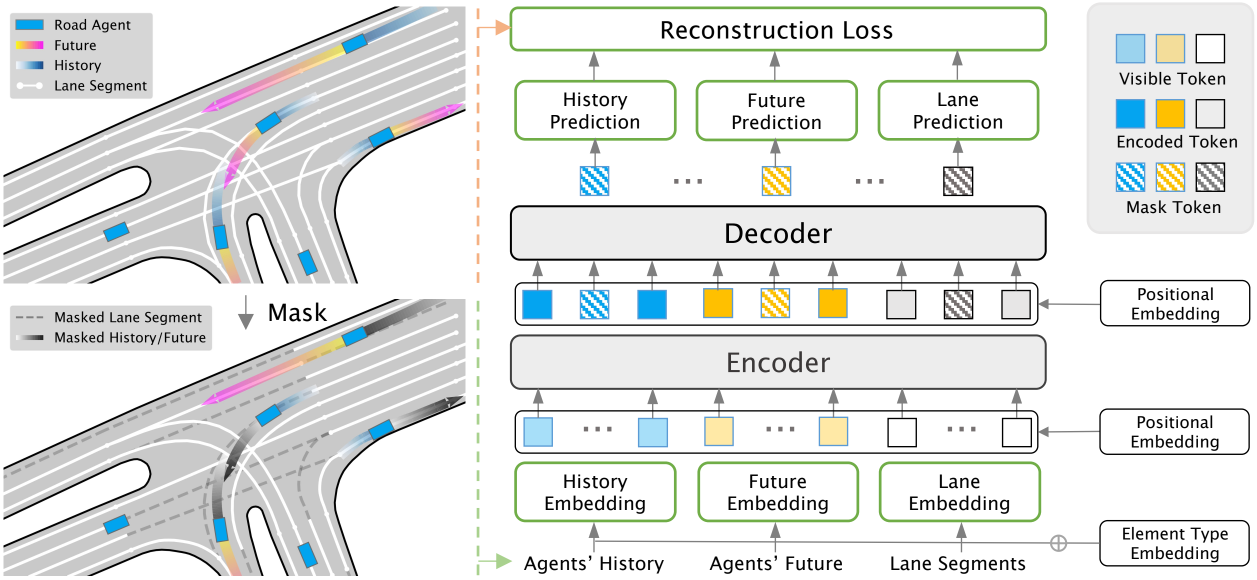

Figure 2. **Overall pre-training scheme of our Forecast-MAE**. The left part shows the masking process of an example scenario (two agents are static within the observation horizon). We randomly mask out the entire agents' history or future trajectory, as well as lane segments. The pre-training scheme is shown on the right. Only the visible history, future trajectory, and lane segments will be embedded into tokens and processed by the encoder. Three different types of mask tokens are added to the input sequence of the decoder to reconstruct history, future trajectory and lane segment, respectively.

when both history and future trajectories are masked.

3.2. Input Representation and Embedding

Following the popular vectorized representation, we treat all agents' trajectories and lane segments as polylines. Specifically, we denote the history trajectories of N agents as $A_H \in \mathbb{R}^{N\times T_H \times C_H}$, where T_H is the number of history frames and C_H is the history feature channels including step-wise displacement/velocity difference, and a padding flag indicating the observation status of this frame. Similarly, the future trajectories are denoted as $A_F \in \mathbb{R}^{N\times T_F \times C_F}$, where T_F is the number of future frames, C_F is the future feature channels, including future coordinates normalized to the current position of agents and a padding flag indicating availability. The non-overlapping lane segments are denoted as $L \in \mathbb{R}^{M\times P\times C_L}$, where M is the number of lane segments within a certain radius of the target agent, P is the number of points of each polyline and C_L is the lane feature channels (e.g., coordinates, availability). Note that we normalize all the coordinates of each lane polyline to its geometric center.

The primary goal of the embedding layer is to encode sequential features into one-dimensional vectors or tokens that can be directly processed by the standard Transformer. We use a Feature Pyramid Network (FPN) [27] similar to LaneGCN to fuse multi-scale agent motion features. 1D neighborhood attention [19] is employed at each scale to extract local motion features. Agents' historical and future features are embedded separately. To capture a broader range of the road map, we employ a lightweight mini-PointNet [34], mainly comprising MLPs and max pooling layers, to embed lane polylines. The embedding process can be formulated as

$$T_H = \mathrm{FPN}(A_H),\ T_F = \mathrm{FPN}(A_F),\ T_{H,F} \in \mathbb{R}^{N\times C} \quad (1)$$

$$T_L = \mathrm{MiniPointNet}(L),\ T_L \in \mathbb{R}^{M\times C}, \quad (2)$$

where T_H, T_F, T_L are history, future, lane tokens respectively, C is the embedding dimension.

Semantic attributions such as agent category (e.g., vehicle, pedestrian, cyclist) or lane types are initialized as learnable embeddings and added to the embedded tokens. Given that the coordinates of agents and lane features are normalized, it is crucial to include global position information in the tokens. The position embedding (PE) is implemented with a simple two-layer MLP following [47], formulated as

$$\mathrm{PE} = \mathrm{MLP}\big([x, y, \cos(\theta), \sin(\theta)]\big), \mathrm{PE} \in \mathbb{R}^C \quad (3)$$

where (x, y, θ) is the latest observed pose of agents or the geometric center pose for lane polylines. The PE is added to the tokens before being processed by the autoencoder.

3.3. AutoEncoder

The autoencoder is entirely based on standard Transformers. The encoder consists of several Transformer blocks and only encodes concatenated visible agents and lane tokens, resulting in encoded latent tokens $T_E \in$

$\mathbb{R}^{(N+M)\times C}$. Following the asymmetric autoencoder design of MAE, history, future, and lane mask tokens $M = (M_H, M_F, M_L)$ are added together with the encoded latent tokens as the input sequence of the decoder and then output the decoded mask tokens $M' = (M_H^{'}, M_F^{'}, M_L^{'})$ after the decoding. Positional embeddings are added to the full input sequence, including the mask tokens. Each type of mask token is a learned vector shared by the corresponding type of masked element. The autoencoding process is formulated as

$$T_E = \text{Encoder}\big(\text{concat}\,(T_H, T_F, T_L) + \text{PE}\big), \quad (4)$$

$$M' = \text{Decoder}\big(\text{concat}\,(T_E, M) + \text{PE}\big). \quad (5)$$

The decoded mask tokens are subsequently used for reconstructing the masked element through a simple prediction head, which is implemented as a linear projection layer in practice.

3.4. Reconstruction Target

The prediction heads predict the normalized 2-dimensional coordinates of history/future trajectories $P_{H/F}$ and lane polylines P_L,

$$P_H = \text{PredictionHead}(M_H'),\ P_H \in \mathbb{R}^{\alpha N\times T_H\times 2}, \quad (6)$$

$$P_F = \text{PredictionHead}(M_F'),\ P_F \in \mathbb{R}^{(1-\alpha)N\times T_F\times 2}, \quad (7)$$

$$P_L = \text{PredictionHead}(M_L'),\ P_L \in \mathbb{R}^{\beta M\times P\times 2}, \quad (8)$$

where α is the agents' history mask ratio, β is the lane segments mask ratio. We use L1 loss $\mathcal{L}_H, \mathcal{L}_F$ for trajectory reconstruction and mean squared error (MSE) loss $\mathcal{L}_L$ for lane polyline reconstruction, and w_H, w_F, w_L correspond to the loss weight respectively. The final loss is

$$\mathcal{L}_{MAE} = w_H\mathcal{L}_H + w_F\mathcal{L}_F + w_L\mathcal{L}_L. \quad (9)$$

3.5. Motion Forecasting

For the target motion forecasting task, we adopt an end-to-end fine-tuning approach. During fine-tuning, the following modifications are made to the pre-training model: (1) we discard the MAE decoder and mask tokens; (2) agents' future features are eliminated from the input, and masking is not employed; (3) the pretext prediction heads are substituted with a multi-modal future decoder.

Multi-modal decoder. Given the multi-modal nature of agents' behavior, motion forecasting entails producing multiple potential future predictions, distinct from the masked future reconstruction pretext task. To maintain a neat framework with minimal inductive bias, we implement the multi-modal decoder using a simple three-layer MLP. A separate three-layer MLP is utilized to generate the confidence score for each prediction. The decoding process can be formulated as

$$P_{Traj} = \text{MLP}(T_H'),\ P_{Traj} \in \mathbb{R}^{N\times K\times T_F\times 2}, \quad (10)$$

where T_H' is the encoded history tokens and K is the number of output modes. The predicted future trajectories are normalized to each agent's latest observed position.

Training loss. We adopt the widely used Huber loss for trajectory regression and cross-entropy loss for confidence classification with equal weights. The winner-take-all strategy is employed, which only optimizes the best prediction with minimal average prediction error to the ground truth. We compute the loss using all agents present in the scene.

4. Experiments

4.1. Experimental Setup

Dataset. We evaluate the proposed framework on the recently released large-scale Argoverse 2 (AV2) dataset. This dataset includes 250K non-overlapping scenarios, divided into 199,908, 24,988, and 24,984 samples for training, validation, and testing, respectively. Each sample contains 5 seconds of history and requires a prediction of 6 seconds in the future, with a sampling rate of 10 Hz. Every scenario includes a focal track agent that needs to be predicted, with detailed high-definition map patches provided for each sample. We choose to evaluate on the Argoverse 2 dataset as it offers the best balance between diversity and dataset size. The popular Argoverse 1 [5] dataset has a similar size but lacks scenario diversity (e.g., the majority of vehicles driving straight-forward). Conversely, Argoverse 2 is intended to be more varied and complicated. Another widely-used dataset, the Waymo Open Motion Dataset (WOMD) [12], has similar scenario complexity but only contains less than half the number of scenarios (104K). We believe that a larger and more complex dataset is more appropriate for evaluating SSL frameworks.

Metrics. We use the official benchmark metrics, including minADE, minFDE, MR, and brier-minFDE, which refer to six prediction modes, if not specified.

Implementation Details. Detailed model architecture and training settings are provided in the supplementary.

4.2. Ablation Study

We conduct ablation studies on the Argoverse 2 validation set. By default, the pre-training epoch is set to 40, the fine-tuning epoch to 30, the history and lane mask ratio to 0.5, and the encoder and decoder depth to 4. The pre-training is only conducted on the training set.

Masking ratio. Figure 3 depicts the impact of varying masking ratios. Employing a well-balanced masking ratio, ranging from 40% to 50%, between an agent's history and future leads to the most favorable outcomes, in agreement with common sense. We posit that a balanced masking ratio for agent trajectory helps prevent the learning of biased features by the model and enhances its comprehension of

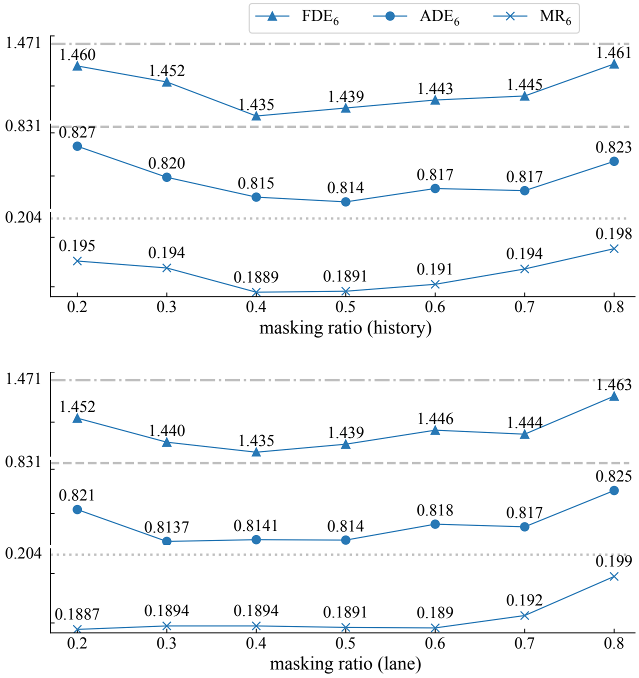

Figure 3. **The impact of the history and lane masking ratio**. The marked blue lines show the fine-tuning results on different metrics (lower is better). The dashed lines in grey correspond to the results of the baseline model (trained from scratch for 30 epochs).

the bidirectional relationship between historical and future motion. This is further demonstrated by the performance of extreme history masking ratios (20% and 80%), which significantly underperform.

Forecast-MAE is relatively insensitive to lane masking ratio, as a wide range of ratios (30% to 60%) perform well. Nonetheless, when the lane masking ratio exceeds 70%, performance suffers notably. The possible reason is with a masking ratio of more than 70%, most of the road structure information loses, which tremendously increases the difficulty of scene reconstruction and geometry feature extraction from the map. Conversely, when the lane masking ratio is below 20%, both ADE and FDE experience a significant increase. We deduce that the masked lanes can be easily extrapolated by nearby visible lanes when only a small subset of lanes are masked.

Hist. mask	Lane mask	Fut. input	Fut. mask	minADE	minFDE	MR
	✓			0.865	1.51	0.212
	✓	✓		0.828	1.47	0.203
✓		✓		0.864	1.53	0.216
✓	✓	✓		0.866	1.52	0.214
	✓	✓	✓	0.820	1.45	0.198
✓	✓	✓	✓	**0.814**	**1.44**	**0.189**
training from scratch				0.8314	1.471	0.2038

Table 1. **Results of different input and masking strategies.** History and lane are used as input for experiments. When history and future trajectories are simultaneously masked, it is masked in a complementary manner. For other situations, we use random masking by default.

encoder depth	minADE	minFDE	MR
2	0.854	1.55	0.221
3	0.823	1.46	0.198
4	**0.814**	1.44	0.189
5	0.815	**1.43**	**0.188**

Table 2. **Results of different encoder depth.** A encoder depth of 4 offers the best performance-efficiency trade-off.

Masking strategy. One distinctive aspect of our approach, compared to existing SSL methods, is the introduction of agents' future trajectories as additional input during pretraining. The outcomes of various inputs and masking strategies are presented in Table 3. When only lane masking is employed, the utilization of future trajectories as input makes a significant difference (minADE is 0.865 without using future, and 0.828 using future). One possible explanation is that the model can establish better connections between lanes and future trajectories through lane reconstruction, which is beneficial for the forecasting task. Interestingly, if we use the future as input and do not mask it, merely masking the history performs even worse than training from scratch (minADE 0.864/0.866 *vs.* 0.8314). A reasonable justification is that the dataset is intended to make the distribution of agents' future trajectories diverse and multi-modal (e.g., an agent is beginning to pass an intersection), while the historical trajectory is much simpler and more predictable. The model might take a shortcut to reconstruct the history by extrapolating the future trajectory, resulting in a failure to learn meaningful features from the agents' motion. As a result, the learned latent features are useless and even harmful for the later forecasting task. Adding future masking promptly addresses this problem, and minADE improves to 0.820 and 0.814. The proposed complementary masking strategy achieves the best performance in all metrics.

Encoder depth. A relative deep encoder is necessary, as studied in Table 2. The performance improved 4.6% in terms of minADE by increasing encoder depth from 2 to 4. Adding more encoder layers does not make a significant difference. We use an encoder depth of 4 as our default setting for its better efficiency-performance trade-off.

4.3. Results

For the final leaderboard submission, we use a depth of 4 for both the decoder and encoder. The history and lane masking ratios are 0.4 and 0.5, respectively. We set the pretraining and fine-tuning epochs both to 60. Our final motion forecasting model is simple and lightweight, with only 1.9M parameters in total.

	Method	minADE$_1$	minFDE$_1$	MR$_1$	minADE$_6$	minFDE$_6$	MR$_6$	b-FDE$_6$
Supervised Learning	THOMAS [14]	1.96	4.71	0.64	0.88	1.51	0.20	2.16
	GoReLa [8]	1.82	4.62	0.61	0.76	1.48	0.22	2.01
	GANet [42]	1.78	4.48	0.60	0.73	1.35	**0.17**	1.97
	QML w/ ensemble [38]	1.84	4.98	**0.59**	**0.69**	1.39	0.19	1.95
	BANet w/ ensemble [49]	1.79	4.61	0.60	0.71	1.36	0.19	1.92
SSL-Lanes [1]	Lane Masking	2.167	5.675	0.671	0.835	1.698	0.248	2.379
	Dist. to Inter.	2.176	5.71	0.667	0.839	1.710	0.248	2.391
	S/F Classification	2.218	5.905	0.687	0.828	1.671	0.249	2.352
Forecast-MAE	Scratch	1.845	4.602	0.623	0.727	1.427	0.187	2.062
	Fine-tune w/o ensemble	1.741	4.355	0.607	0.709	1.392	0.172	2.029
	Fine-tune w/ ensemble	**1.658**	**4.145**	**0.592**	**0.690**	**1.338**	0.173	**1.911**
SSL-Lanes [1]	Lane Masking	2.014	5.194	0.649	0.850	1.520	0.220	2.197
	Dist. to Inter.	2.006	5.187	0.651	0.840	1.490	0.212	2.182
	S/F Classification	2.120	5.613	0.675	0.861	1.536	0.224	2.216
Forecast-MAE	Scratch	1.813	4.570	0.622	0.811	1.436	0.189	2.074
	Fine-tune w/o ensemble	**1.755**	**4.388**	**0.609**	**0.801**	**1.409**	**0.178**	**2.042**

Table 3. Comparisons with previous results on the Argovesrse 2 test set (upper group) and validation set (lower group). For all the metrics, the lower is the better. We **bold** the best results and underline the second best results.

	epochs	minADE$_6$	minFDE$_6$	MR$_6$
Scratch	60	0.811	1.436	0.189
	70	0.815	1.436	0.187
	80	0.814	1.450	0.190
Fine-tune	60	**0.801**	**1.409**	**0.178**

Table 4. **Comparison with training from scratch of different training epochs**. Continue increase training iterations does not further improves the performance of training from scratch.

Comparison with the other SSL method. We compare our method with SSL-Lanes, as it is the only published approach that employs vector representation and SSL. We make minimal modifications to its official code base[1] to adapt it to the AV2 dataset. Our experiments utilize three of its pretext tasks, specifically lane making, distance to the intersection (Dist. to Inter.), and success-failure classification (S/F classification). We do not implement the maneuver classification pretext task, as AV2 lacks lane-turning information. Table 3 (lower group) displays the comparison results on the AV2 validation set. Our Forecast-MAE outperforms all SSL-Lanes variants significantly across all metrics. Notably, SSL-Lanes suffers from performance degradation between the validation and test sets, whereas our approach achieves consistent performance on both sets and even performs slightly better on the test set. This suggests that our method learns superior and more generalized features through MAE-based self-supervised pre-training.

Comparison with State-of-the-art. Our Forecast-MAE, developed using standard Transformer blocks and minimal prior knowledge, demonstrates impressive performance on the leaderboard, depicted in Table 3 upper group. Particularly noteworthy is that our approach (w/o ensemble) outperforms all other methods, including ensemble models, in terms of minADE$_1$ and minFDE$_1$, indicating its superior ability to predict the most likely future. We attribute this to our SSL pre-training scheme, which requires the model to reconstruct the most likely masked history and future trajectories. Additionally, Forecast-MAE (w/o ensemble) achieves the best minADE$_6$ among all non-ensemble methods and performs on par with QML (w/ ensemble). Through the adoption of an ensemble strategy involving 6 variants of our framework (*e.g.*, different masking ratios, encoder depth), our ensemble model achieves the best performance among all methods across six metrics. In particular, our ensemble model outperforms the second-best (GANet) by 7.5% in terms of minFDE$_1$.

Comparison with training from scratch. The comparison results between the fine-tuned model and the model trained from scratch are presented in Table 3. It is noteworthy that the vanilla model, despite its simplicity, serves as a strong baseline. However, our fine-tuned model outperforms the baseline in all metrics, exhibiting improvements of 5.1% on minADE$_1$, 5.7% on minFDE$_1$, 2.4% on minADE$_6$, and minFDE$_6$, without the utilization of additional data or a more complex model.

As we incorporate agents' future trajectories as inputs during the pre-training, a plausible concern is that the fine-tuned model may benefit from additional training iterations. To address this, we conduct further training of the vanilla model with more epochs using cosine learning rate decay.

[1] https://github.com/AutoVision-cloud/SSL-Lanes

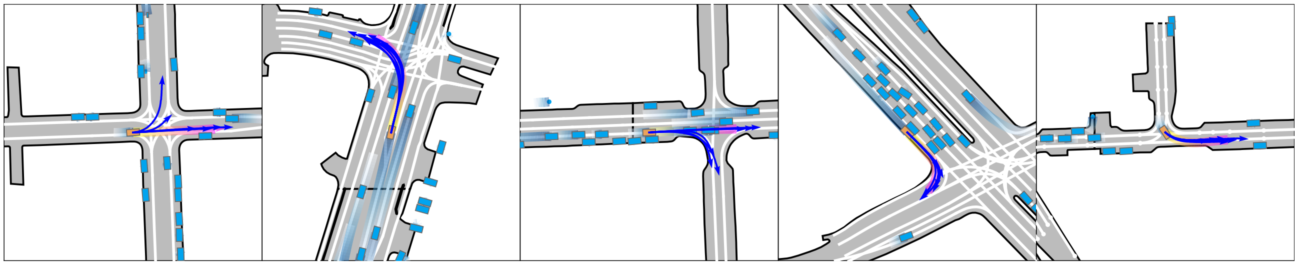

Figure 4. **Qualitative results of Forecast-MAE** The predicted trajectories (K=6) are in blue and the ground truth is in gradient pink. The bounding box in orange indicates the focal agent.

	minADE	minFDE	MR
Scratch	0.9098	1.645	0.2346
Fine-tune	**0.8968**	**1.613**	**0.2164**

Table 5. **Evaluation results on different data distribution.**. Models are trained or pre-trained on scenarios in *Miami, Pittsburgh* and *Austin* and validated in *Dearborn, Palo-alto* and *Washington-DC*.

The results presented in Table 4 indicate that continuing to increase the training iterations fails to enhance the performance of the model trained from scratch, underscoring the importance of pre-training.

Generalization ability. Our method demonstrates strong generalization ability in the AV2 benchmark, as evidenced by the results in Table 3. To further investigate this point, we design an experiment where training and testing employ different data distributions. Specifically, we partition all scenarios involving six cities in the AV2 dataset into two distinct and non-overlapping groups. We then train or pre-train the models solely on scenarios in *Miami, Pittsburgh, and Austin*, and evaluate them on *Dearborn, Palo-Alto, and Washington-DC*. Results presented in Table 5 indicate that the fine-tuned model surpasses the baseline in all metrics, signifying that self-supervised pre-training enables learning of more generalizable features.

Qualitative Results. We visualize the qualitative results of our fine-tuned model on the AV2 validation set, as shown in Figure 4. We leave more results to the supplementary, due to the limited space.

5. Conclusion

We present Forecast-MAE, a simple and neat framework for self-supervised pre-training on the motion forecasting task. Based on the asymmetric architecture of MAE, we devise a scene reconstruction pretext task that utilizes a novel masking strategy. By leveraging the complementary masking of the agents' trajectories and the random masking of lane segments during the pre-training process, the model acquires the ability to capture the bidirectional agent motion features, road geometry features, and cross-modal interconnections jointly. Our experiments on the challenging Argoverse 2 benchmark demonstrate that our Forecast-MAE surpasses supervised learning methods and previous self-supervised learning works, especially in terms of minADE_1 and minFDE_1, indicating its superior ability to predict the most likely future.

Limitaion and Dicussion. One constraint of our work is the lack of exploration of transfer learning or few-shot learning for the proposed method (*e.g.*, pre-training on WMOD and fine-tuning on AV2). Such exploration is hindered by the different problem settings, namely observation/prediction horizons, of different datasets. Besides, due to the relatively limited size of publicly available motion forecasting datasets compared to those in computer vision or natural language processing, we are unable to determine whether the performance of Forecast-MAE will scale up with increased training data and model capacity. However, we are positive about this point by drawing intuition from MAE and our minimal inductive bias design. Our approach could be advantageous for autonomous driving companies with large-scale internal datasets. Although Forecast-MAE already achieves strong performance while designed to be simple, we anticipate it can be further improved. Drawing inspiration from the development of techniques such as ViT [11] to Swin-Trainsformer [30], properly incorporating inductive bias such as relative position design [52, 8, 51] or local attention [36] may further boost Forecast-MAE in terms of performance and efficiency. Another possible direction is to generate realistic traffic scenarios building upon this work. These possibilities are left for future works.

Acknowledgement This work was supported by Guangdong Basic and Applied Basic Research Foundation, under project 2021B1515120032, Jiangsu Province Science and Technology Project BZ2021056, and Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone (HZQB-KCZYB-2020083), awarded to Prof. Ming Liu.

References

[1] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. Ssl-lanes: Self-supervised learning for motion

forecasting in autonomous driving. In *6th Annual Conference on Robot Learning*. 2, 7

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 3

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021. 1, 2

[4] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, pages 86–99. PMLR, 2020. 2, 3

[5] 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. 2, 5

[6] Yuying Chen, Congcong Liu, Xiaodong Mei, Bertram E. Shi, and Ming Liu. Hgcn-gjs: Hierarchical graph convolutional network with groupwise joint sampling for trajectory prediction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13400–13405, 2022. 3

[7] Yuying Chen, Congcong Liu, Bertram Shi, and Ming Liu. Comogcn: Coherent motion aware trajectory prediction with graph representation. *arXiv preprint arXiv:2005.00754*, 2020. 3

[8] Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, and Raquel Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. *arXiv preprint arXiv:2211.02545*, 2022. 7, 8

[9] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *Conference on Robot Learning*, pages 203–212. PMLR, 2022. 2, 3

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2, 3

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 8

[12] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 2, 5

[13] 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. 2, 3

[14] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Thomas: Trajectory heatmap output with learned multi-agent sampling. In *International Conference on Learning Representations*. 7

[15] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Home: Heatmap output for future motion estimation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 500–507. IEEE, 2021. 3

[16] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9107–9114. IEEE, 2022. 3

[17] 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*. 3

[18] 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. 3

[19] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022. 4

[20] 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. 2, 3

[21] Po-Yao Huang, Hu Xu, Juncheng B Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. In *Advances in Neural Information Processing Systems*. 2

[22] Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 7(3):652–674, 2022. 1

[23] Xiaosong Jia, Liting Sun, Hang Zhao, Masayoshi Tomizuka, and Wei Zhan. Multi-agent trajectory prediction by combining egocentric and allocentric views. In *Conference on Robot Learning*, pages 1434–1443. PMLR, 2022. 3

[24] Xiaosong Jia, Penghao Wu, Li Chen, Hongyang Li, Yu Liu, and Junchi Yan. Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *arXiv preprint arXiv:2205.09753*, 2022. 3

[25] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction

for autonomous driving. *arXiv preprint arXiv:2008.10587*, 2020. 3

[26] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 541–556. Springer, 2020. 2, 3

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[28] Congcong Liu, Yuying Chen, Ming Liu, and Bertram E. Shi. Avgcn: Trajectory prediction using graph convolutional networks guided by human attention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14234–14240, 2021. 3

[29] 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. 2, 3

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 8

[31] 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 *International Conference on Learning Representations*, 2022. 3

[32] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022. 2

[33] 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. 3

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 4

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[36] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems*. 2, 3, 8

[37] Haoran Song, Di Luan, Wenchao Ding, Michael Y Wang, and Qifeng Chen. Learning to predict vehicle trajectories with model-based planning. In *Conference on Robot Learning*, pages 1035–1045. PMLR, 2022. 3

[38] Tong Su, Xishun Wang, and Xiaodong Yang. Qml for argoverse 2 motion forecasting challenge. *arXiv preprint arXiv:2207.06553*, 2022. 7

[39] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. *Advances in neural information processing systems*, 32, 2019. 2, 3

[40] 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. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821. IEEE, 2022. 3

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[42] 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. 7

[43] 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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2

[44] Chenfeng Xu, Tian Li, Chen Tang, Lingfeng Sun, Kurt Keutzer, Masayoshi Tomizuka, Alireza Fathi, and Wei Zhan. Pretram: Self-supervised pre-training via connecting trajectory and map. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 34–50. Springer, 2022. 2

[45] 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. 3

[46] 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. 3

[47] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 2, 4

[48] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Confer-

ence on Intelligent Robots and Systems (IROS), pages 532–539. IEEE, 2021. 3

[49] Chen Zhang, Honglin Sun, Chen Chen, and Yandong Guo. Banet: Motion forecasting with boundary aware network. 7

[50] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021. 2, 3

[51] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17863–17873, 2023. 3, 8

[52] 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. 2, 3, 8