SSL-Lanes: Self-Supervised Learning for Motion Forecasting in Autonomous Driving

Anonymous Author(s)
Affiliation
Address
`email`

Abstract: Self-supervised learning (SSL) is an emerging technique that has been successfully employed to train convolutional neural networks (CNNs) and graph neural networks (GNNs) for more transferable, generalizable, and robust representation learning. However its potential in motion forecasting for autonomous driving has rarely been explored. In this study, we report the first systematic exploration and assessment of incorporating self-supervision into motion forecasting. We first propose to investigate four novel self-supervised learning tasks for motion forecasting with theoretical rationale and quantitative and qualitative comparisons on the challenging large-scale Argoverse dataset. Secondly, we point out that our auxiliary SSL-based learning setup not only outperforms forecasting methods which use transformers, complicated fusion mechanisms and sophisticated online dense goal candidate optimization algorithms in terms of performance accuracy, but also has low inference time and architectural complexity. Lastly, we conduct several experiments to understand why SSL improves motion forecasting.

Keywords: Motion Forecasting, Autonomous Driving, Self-Supervised Learning

1 Introduction

Motion forecasting in a real-world urban environment is an important task for autonomous robots. It involves predicting the future trajectories of traffic agents including vehicles and pedestrians. This is absolutely crucial in the self-driving domain for safe, comfortable and efficient operation. However, this is a very challenging problem. Difficulties include inherent stochasticity and multimodality of driving behaviors, and that future motion can involve complicated maneuvers such as yielding, nudging, lane-changing, turning and acceleration or deceleration.

The motion prediction task has traditionally been based on kinematic constraints and road map information with handcrafted rules. These approaches however fail to capture long-term behavior and interactions with map structure and other traffic agents in complex scenarios. Tremendous progress has been made with data-driven methods in motion forecasting [3, 4, 5, 6, 7, 8, 9, 10]. Recent methods use a vector representation for HD maps and agent trajectories, including approaches like Lane-GCN [2], Lane-RCNN [11], Vector-Net [12], TNT [5] and Dense-TNT [6]. More recently, the enormous success of transformers [13] has been leveraged for forecasting in mm-Transformer [9], Scene transformer [8], Multimodal transformer [14] and Latent Variable Sequential Transformers [15]. Most of these methods however are extremely complex in terms of architecture and have low inference speeds, which makes them unsuitable for real-world settings.

In this work, we extend ideas from self-supervised learning (SSL) to the motion forecasting task. Self-supervision has seen huge interest in both natural language processing and computer vision [16] to make use of freely available data without the need for annotations. It aims to assist the model to learn more transferable and generalized representation from pseudo-labels via pretext tasks. Given the recent success of self-supervision with CNNs, transformers, and GNNs, we are naturally motivated to ask the question: *Can self-supervised learning improve accuracy and generalizability of motion forecasting, without sacrificing inference speed or architectural simplicity?*

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

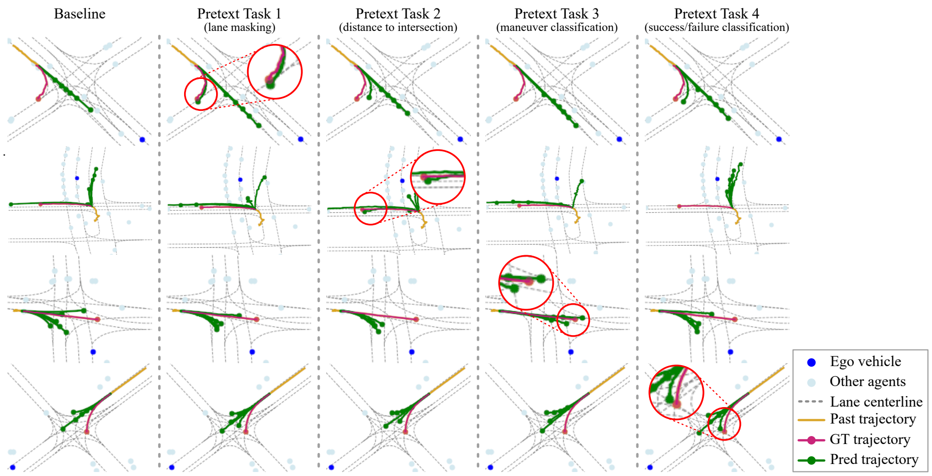

Figure 1: Motion forecasting on Argoverse [1] validation. We show four challenging scenarios at intersections. The baseline [2] misses all the predictions. In the first row, our proposed lane masking successfully captures the right-turn. For the second row, predicting distance to intersection helps the most in capturing the left turn. In the third row, acceleration at an intersection is best captured by the model that is made to classify maneuvers of traffic agents. Finally, in the fourth row, classifying successful final goal states is the most effective at capturing the left turn. These tasks are trained with pseudo-labels which are obtained for free from data. Please refer to Sec. 6.2 for details.

Contributions: Our work, SSL-Lanes, presents the first systematic study on how to incorporate self-supervision in a standard data-driven motion forecasting model. Our contributions are: (a) We demonstrate the effectiveness of incorporating self-supervised learning in motion forecasting. Since this does not add extra parameters or compute during inference, SSL-Lanes achieves the best accuracy-simplicity-efficiency trade-off on the challenging large-scale Argoverse [1] benchmark. (b) We propose four self-supervised tasks based on the nature of the motion forecasting problem. The key idea is to leverage easily accessible map/agent-level information to define domain-specific pretext tasks that encourage the standard model to capture more superior and generalizable representations for forecasting in comparison to pure supervised learning. (c) We further design experiments to explore why forecasting benefits from SSL. We provide extensive results to hypothesize that SSL-Lanes learns richer features from the SSL training as compared to a model trained with vanilla supervised learning.

2 Related Work

Motion Forecasting: Traditional methods for motion forecasting primarily use Kalman filtering [17] with a prior from HD-maps to predict future motion states [18, 19]. With the huge success of deep learning, recent works use data-driven approaches for motion forecasting. These methods explore different architectures involving rasterized images and CNNs [3, 20, 21], vectorized representations and GNNs [12, 11, 22, 4, 7], point-cloud representations [23], transformers [8, 9, 15, 14] and sophisticated fusion mechanisms [2], to generate features that predict final output trajectories. While the focus of these works is to find more effective ways of feature extraction from HD-maps and interacting agents, they need huge model capacity, heavy parameterization, and extensive augmentations or large amounts of data to converge to a general solution. Other works [5, 10, 24, 25] build on them to incorporate prior knowledge in the form of predefined candidate trajectories obtained from sampling or clustering strategies from training data. However the disadvantage of these methods is that their performance is highly related to the quality of the trajectory proposals, which becomes an extra dependency. End-to-end solutions for optimizing end-points of these candidates trajectories are proposed by Dense-TNT [6] and HOME [26]. Dense-TNT has state-of-the-art accuracy with a reasonable parameter budget, but its online dense goal candidate optimization strategy is computationally very expensive, which is unrealistic for real-time operations like autonomous driv-

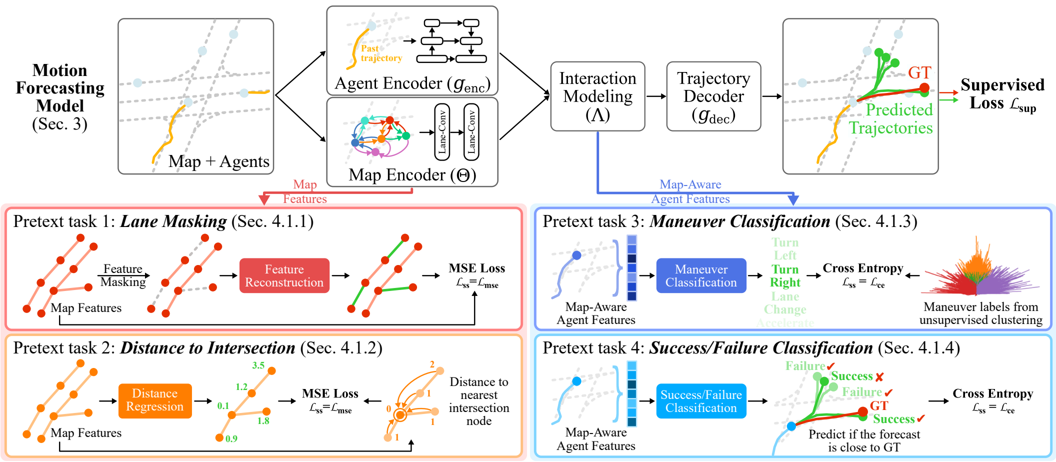

Figure 2: Illustration of the overall SSL-Lanes framework for self-supervision on motion forecasting through joint training. SSL-Lanes improves upon a standard-motion forecasting baseline, that consists of an agent encoder, map encoder, interaction model and a trajectory decoder, trained using a supervised loss $\mathcal{L}_{sup}$. SSL-Lanes proposes four pretext tasks: (1) Lane Masking: which recovers feature information from the perturbed lane graphs. (2) Distance to Intersection: which predicts the distance (in terms of shortest path length) from all lane nodes to intersection nodes. (3) Maneuver Classification: predicts the form of a 'maneuver' the agent-of-interest intends to execute (4) Success/Failure Classification: which trains an agent specialized at achieving end-point goals.

ing. Lately, ensembling techniques like MultiPath++ [27] and DCMS [28] have been proposed and while they have high forecasting performance, a major disadvantage is their high memory cost for training and heavy computational cost at inference. We also refer the reader to the supplementary for a detailed discussion of how SSL-Lanes differs from methods like Vector-Net [12], CS-LSTM [29] and MultiPath[3].

Self-supervised Learning: SSL is a rapidly emerging learning framework that generates additional supervised signals to train deep learning models through carefully designed pretext tasks. In the image domain, various self-supervised learning techniques have been developed for learning high-level image representations, including predicting the relative locations of image patches [30], jigsaw puzzle [31], image rotation [32], image clustering [33], image inpainting [34], image colorization [35] and segmentation prediction [36]. In the domain of graphs and graph neural networks, pretext tasks include graph partitioning, node clustering, context prediction and graph completion [37, 38, 39, 40]. To the best of our knowledge, this is the first principled approach that explores motion forecasting for autonomous driving with self-supervision.

3 Background

Problem Formulation: We are given the past motion of N actors. The i-th actor is denoted as a set of center locations over the past L time-steps. We pre-process it to represent each trajectory as a sequence of displacements $\mathcal{P}_i = \{\Delta \boldsymbol{p}_i^{-L+1}, ..., \Delta \boldsymbol{p}_i^{-1}, \Delta \boldsymbol{p}_i^{0}\}$, where $\boldsymbol{p}_i^l$ is the 2D displacement from time step $l-1$ to l. We are also given a high-definition (HD) map, which contains lanes and semantic attributes. Each lane is composed of many consecutive lane nodes, with a total of M nodes. $\boldsymbol{X} \in \mathbb{R}^{M \times F}$ denotes the lane node feature matrix, where $\boldsymbol{x}_j = \boldsymbol{X}[j,:]^T$ is the F-dimensional lane node vector. Following the connections between lane centerlines (i.e., predecessor, successor, left neighbour and right neighbour), we represent the connectivity within the lane nodes with 4 adjacency matrices $\{\boldsymbol{A}_f\}_{f \in \{\text{pre,suc,left,right}\}}$, with $\boldsymbol{A}_f \in \mathbb{R}^{M \times M}$. This implies that if $\boldsymbol{A}_{f,gh} = 1$, then node h is an f-type neighbor of node g. Our goal is to forecast the future motions of all actors in the scene $\mathcal{O}_{\text{GT}}^{1:T} = \{(x_i^1, y_i^1), ..., (x_i^T, y_i^T) | i = 1, ..., N\}$, where T is our prediction horizon.

SSL Task	Property Level	Primary Assumption	Type
Lane-Masking	Map features	Local map structure	Aux. auto-encoder
Distance to Intersection		Global map structure	Aux. regression
Maneuver Classification	Map-aware agent features	Agent feature similarity	Aux. classification
Success/Failure Classification		Distance to success state	

Table 1: Overview of our proposed self-supervised (SSL) tasks

Standard Motion Forecasting Model: We briefly introduce a standard data-driven motion forecasting framework, consisting of a feature encoder, interaction-modeler and prediction header.

Feature Encoding: We first encode the agent and map inputs similar to Lane-GCN [2]. The agent encoder includes a 1D convolution with a feature pyramid network, parameterized by g_{enc}, as given by Eq. (1). For map-encoding, we adopt two Lane-Conv residual blocks, parameterized by $\boldsymbol{\Theta} = \{\boldsymbol{W}_0, \boldsymbol{W}_{\text{left}}, \boldsymbol{W}_{\text{right}}, \boldsymbol{W}_{\text{pre},k}, \boldsymbol{W}_{\text{suc},k}\}$, where $k \in \{1, 2, 4, 8, 16, 32\}$, as given by Eq. (2).

$$\hat{\boldsymbol{p}}_i = g_{\text{enc}}(\mathcal{P}_i) \tag{1}$$

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}_0 + \sum_{j\in\{\text{left,right}\}} \boldsymbol{A}_j\boldsymbol{X}\boldsymbol{W}_j + \sum_k \boldsymbol{A}^k_{\text{pre}}\boldsymbol{X}\boldsymbol{W}_{\text{pre},k} + \boldsymbol{A}^k_{\text{suc}}\boldsymbol{X}\boldsymbol{W}_{\text{suc},k} \tag{2}$$

Modeling Interactions: Since the behavior of agents depends on map topology and social consistency, each encoded agent i subsequently aggregates context from the surrounding map features and its neighboring agent features, via spatial attention [41] as given by Eq. (3):

$$\begin{aligned}\tilde{\boldsymbol{p}}_i &= \hat{\boldsymbol{p}}_i\boldsymbol{W}_{\text{M2A}} + \sum_j \phi(\text{concat}(\hat{\boldsymbol{p}}_i, \Delta_{i,j}, \boldsymbol{y}_j)\boldsymbol{W}_1)\boldsymbol{W}_2 \\ \acute{\boldsymbol{p}}_i &= \tilde{\boldsymbol{p}}_i\boldsymbol{W}_{\text{A2A}} + \sum_j \phi(\text{concat}(\tilde{\boldsymbol{p}}_i, \Delta_{i,j}, \tilde{\boldsymbol{p}}_j)\boldsymbol{W}_3)\boldsymbol{W}_4\end{aligned} \tag{3}$$

Here, $\boldsymbol{y}_j$ is the feature of the j-th node, $\hat{\boldsymbol{p}}_i$ is the feature of the i-th agent, ϕ the composition of layer normalization and ReLU, and $\Delta_{ij} = \text{MLP}(\boldsymbol{v}_j - \boldsymbol{v}_i)$, where $\boldsymbol{v}$ denotes the (x, y) 2-D bird's-eye-view (BEV) location of the agent or the lane node. The parameters for map and agent feature aggregation is represented by $\boldsymbol{\Lambda} = \{\boldsymbol{W}_{\text{M2A}}, \boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_{\text{A2A}}, \boldsymbol{W}_3, \boldsymbol{W}_4\}$.

Trajectory Prediction: Finally, we decode the future trajectories from the features $\acute{\boldsymbol{p}}_i$ corresponding to the agents of interest as given by: $\mathcal{O}^{1:T}_{\text{pred}} = \{g_{\text{dec}}(\acute{\boldsymbol{p}}_i)|i = 1, ..., N\}$, where g_{dec} is the parameterized trajectory decoder. The parameters for the motion forecasting model are learned by minimizing the supervised loss ($\mathcal{L}_{\text{sup}}$) calculated between the predicted output and the ground-truth future trajectories ($\mathcal{O}^{1:T}_{\text{GT}}$), as given by Eq. (4):

$$g^\star_{\text{enc}}, \boldsymbol{\Theta}^\star, \boldsymbol{\Lambda}^\star, g^\star_{\text{dec}} = \underset{g_{\text{enc}},\boldsymbol{\Theta},\boldsymbol{\Lambda},g_{\text{dec}}}{\arg\min} \; \mathcal{L}_{\text{sup}}(\mathcal{O}^{1:T}_{\text{pred}}, \mathcal{O}^{1:T}_{\text{GT}}) \tag{4}$$

4 SSL-Lanes

The goal of our proposed SSL-Lanes framework is to improve the performance of the primary motion forecasting baseline by learning simultaneously with various self-supervised tasks. Fig. 2 shows the pipeline of our proposed approach, and Tab. 1 summarizes the self-supervised tasks.

Self-Supervision meets Motion Forecasting: Considering our motion forecasting task and a self-supervised task, the output and the training process can be formulated as:

$$\boldsymbol{\Psi}^\star, \boldsymbol{\Omega}^\star, \boldsymbol{\Theta}^\star_{\text{ss}} = \underset{\boldsymbol{\Psi},\boldsymbol{\Omega},\boldsymbol{\Theta}_{\text{ss}}}{\arg\min} \; \alpha_1\mathcal{L}_{\text{sup}}(\boldsymbol{\Psi}, \boldsymbol{\Omega}) + \alpha_2\mathcal{L}_{\text{ss}}(\boldsymbol{\Psi}, \boldsymbol{\Theta}_{\text{ss}}) \tag{5}$$

where, $\mathcal{L}_{\text{ss}}(\cdot, \cdot)$ is the loss function of the self-supervised task, $\boldsymbol{\Theta}_{\text{ss}}$ parameterizes the corresponding task-specific layers, and $\alpha_1, \alpha_2 \in \mathrm{R}_{>0}$ are the weights for the supervised and self-supervised losses. If the pretext task only focuses on the map encoder, then $\boldsymbol{\Psi} = \{\boldsymbol{\Theta}\}$ and $\boldsymbol{\Omega} = \{g_{\text{enc}}, \boldsymbol{\Lambda}, g_{\text{dec}}\}$. Otherwise, $\boldsymbol{\Psi} = \{g_{\text{enc}}, \boldsymbol{\Theta}, \boldsymbol{\Lambda}\}$ and $\boldsymbol{\Omega} = \{g_{\text{dec}}\}$. Henceforth, we also define the following representations. We will represent the primary task encoder as function $f_{\boldsymbol{\Psi}}$, parameterized by $\boldsymbol{\Psi}$. Furthermore, given a pretext task, which we will design in the next section, the pretext decoder $p_{\boldsymbol{\Theta}_{\text{ss}}}$ is a function that predicts pseudo-labels and is parameterized by $\boldsymbol{\Theta}_{\text{ss}}$.

Method	$\textbf{minADE}_1$	$\textbf{minFDE}_1$	$\textbf{MR}_1$	$\textbf{minADE}_6$	$\textbf{minFDE}_6$	$\textbf{MR}_6$
Baseline	1.42	3.18	51.35	0.73	1.12	11.07
Lane-Masking	1.36	2.96	49.45	**0.70**	1.02	8.82
Distance to Intersection	1.38	3.02	49.53	0.71	1.04	8.93
Maneuver Classification	**1.33**	**2.90**	49.26	0.72	1.05	9.36
Success/Failure Classification	1.35	2.93	**48.54**	**0.70**	**1.01**	**8.59**

Table 2: Motion forecasting performance on Argoverse validation with our proposed pretext tasks

4.1 Pretext tasks for Motion Forecasting

At the core of our SSL-Lanes approach is defining pretext tasks based upon self-supervised information from the underlying map structure *and* the overall temporal prediction problem itself (Tab. 1).

4.1.1 Lane-Masking

The goal of the *Lane-Masking* pretext task is to encourage the map encoder $\mathbf{\Psi} = \{\mathbf{\Theta}\}$ to learn local structure information in addition to the forecasting task that is being optimized. Specifically, we randomly mask (i.e., set equal to zero) the features of m_a percent of nodes per lane and then ask the self-supervised decoder to reconstruct these features.

$$\mathbf{\Psi}^\star, \mathbf{\Theta}_{\text{ss}}^\star = \underset{\mathbf{\Psi}, \mathbf{\Theta}_{\text{ss}}}{\arg\min} \frac{1}{m_a} \sum_{i=1}^{m_a} \mathcal{L}_{\text{mse}}\Big(p_{\mathbf{\Theta}_{\text{ss}}}([f_{\mathbf{\Psi}}(\tilde{\boldsymbol{X}}, \boldsymbol{A}_f)]_{\boldsymbol{v}_i}), \boldsymbol{X}_i\Big) \tag{6}$$

Here, $\tilde{\boldsymbol{X}}$ is the node feature matrix corrupted with random masking, i.e., some rows of $\boldsymbol{X}$ corresponding to nodes $\boldsymbol{v}_i$ are set to zero. $p_{\mathbf{\Theta}_{\text{ss}}}$ is a fully connected network that maps the representations to the reconstructed features. $\mathcal{L}_{\text{mse}}$ is the mean squared error (MSE) loss function penalizing the distance between the reconstructed map features $p_{\mathbf{\Theta}_{\text{ss}}}([f_{\mathbf{\Psi}}(\tilde{\boldsymbol{X}}, \boldsymbol{A}_f)]_{\boldsymbol{v}_i})$ for node $\boldsymbol{v}_i$ and its actual features $\boldsymbol{X}_i$.

4.1.2 Distance to Intersection

Distance-to-Intersection pretext task is proposed to guide the map-encoder, $\mathbf{\Psi} = \{\mathbf{\Theta}\}$, to maintain global topology information by predicting the distance (in terms of shortest path length) from all lane nodes to intersection nodes. We aim to regress the distances from each lane node to pre-labeled intersection nodes annotated as part of the dataset. Given K labeled intersection nodes $\mathcal{V}_{\text{intersection}} = \{\boldsymbol{v}_{\text{intersection},k} | k = 1, ...K\}$, we first generate reliable pseudo labels using breadth-first search (BFS). Specifically, BFS calculates the shortest distance $d_i \in \mathbb{R}$ for every lane node $\boldsymbol{v}_i$ from the given set $\mathcal{V}_{\text{intersection}}$. The target of this task is to predict the pseudo-labeled distances using a pretext decoder. If $p_{\mathbf{\Theta}_{\text{ss}}}([f_{\mathbf{\Psi}}(\boldsymbol{X}, \boldsymbol{A}_f)]_{\boldsymbol{v}_i})$ is the prediction of node $\boldsymbol{v}_i$, and $\mathcal{L}_{\text{mse}}$ is the mean-squared error loss function for regression, then the loss formulation for this SSL pretext task is as follows:

$$\mathbf{\Psi}^\star, \mathbf{\Theta}_{\text{ss}}^\star = \underset{\mathbf{\Psi}, \mathbf{\Theta}_{\text{ss}}}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{\text{mse}}\Big(p_{\mathbf{\Theta}_{\text{ss}}}([f_{\mathbf{\Psi}}(\boldsymbol{X}, \boldsymbol{A}_f)]_{\boldsymbol{v}_i}), d_i\Big) \tag{7}$$

4.1.3 Maneuver Classification

We propose *Maneuver Classification*, and we expect it to provide prior regularization to $\mathbf{\Psi} = \{g_{\text{enc}}, \mathbf{\Theta}, \mathbf{\Lambda}\}$, based on driving modes of agents. We aim to construct pseudo label to divide agents into different clusters according to their driving behavior and thus explore unsupervised clustering algorithms to acquire the maneuver for each agent. We find that using naive k-Means (on agent end-points) or DBSCAN (on Hausdorff distance between entire trajectories [42]) leads to noisy clustering. We find that constrained k-means [43] on agent end-points works best to divide trajectory samples into C clusters equally. We define $C = \{\text{maintain-speed, accelerate, decelerate, turn-left, turn-right, lane-change}\}$ and the clustering function as ρ. If $p_{\mathbf{\Theta}_{\text{ss}}}(f_{\mathbf{\Psi}}(\mathcal{P}_i, \boldsymbol{X}, \boldsymbol{A}_f))$ is the prediction of agent i's intention and $E_i = (x^T_{i,\text{GT}}, y^T_{i,\text{GT}})$

Method	minADE_1	minFDE_1	MR_1	minADE_6	minFDE_6	MR_6	b-FDE_6
NN + Map [1]	3.65	8.12	94.0	2.08	4.02	58.0	-
Jean [4]	1.74	4.24	68.56	0.98	1.42	13.08	2.12
Lane-GCN [2]	1.71	3.78	58.77	0.87	1.36	16.20	2.05
LaneRCNN [11]	1.68	3.69	56.85	0.90	1.45	12.32	2.15
TNT [5]	1.77	3.91	59.70	0.94	1.54	13.30	2.14
DenseTNT [6]	1.68	3.63	58.43	0.88	1.28	12.58	1.97
PRIME [24]	1.91	3.82	58.67	1.22	1.55	11.50	2.09
WIMP [7]	1.82	4.03	62.88	0.90	1.42	16.69	2.11
TPCN [23]	1.66	3.69	58.80	0.87	1.38	15.80	1.92
HOME [26]	1.70	3.68	57.23	0.89	1.29	**8.46**	**1.86**
mmTransformer [9]	1.77	4.00	61.78	0.87	1.34	15.40	2.03
MultiModalTransformer [14]	1.74	3.90	60.23	0.84	1.29	14.29	1.94
LatentVariableTransformer [15]	-	-	-	0.89	1.41	16.00	-
SceneTransformer [8]	1.81	4.06	59.21	**0.80**	**1.23**	12.55	1.88
Success/Failure Classification (Ours)	**1.63**	**3.56**	**56.71**	0.84	1.25	13.26	1.94

Table 3: Comparison of our (best) proposed model and top approaches on the Argoverse Test. The best results are in bold and underlined, and the second best is also underlined.

is its ground-truth end-point, then the learning objective is to classify each agent maneuver into its corresponding cluster using cross-entropy loss $\mathcal{L}_{ce}$ as:

$$\boldsymbol{\Psi}^\star, \boldsymbol{\Theta}_{\text{ss}}^\star = \underset{\boldsymbol{\Psi}, \boldsymbol{\Theta}_{\text{ss}}}{\arg\min}\, \mathcal{L}_{\text{ce}}\Big(p_{\boldsymbol{\Theta}_{\text{ss}}}(f_{\boldsymbol{\Psi}}(\mathcal{P}_i, \boldsymbol{X}, \boldsymbol{A}_f)), \rho(E_i)\Big) \tag{8}$$

4.1.4 Forecasting Success/Failure Classification

We propose a pretext task called *Success/Failure Classification*, which trains an agent specialized at achieving end-point goals and thus links directly to the forecasting task. We expect this to constrain $\boldsymbol{\Psi} = \{g_{\text{enc}}, \boldsymbol{\Theta}, \boldsymbol{\Lambda}\}$ to predict trajectories ϵ distance away from the correct final end-point. Similar to maneuver classification, we wish to create pseudo-labels for our data samples. We label trajectory predictions as successful ($c = 1$) if the final prediction $(x^T_{i,\text{pred}}, y^T_{i,\text{pred}})$ is within $\epsilon < 2\,m$ of the final end-point E_i, and as failure ($c = 0$) otherwise. We choose $2\,m$ as our ϵ threshold because it is also used for miss-rate calculation (Sec. 5). If the pretext decoder predicts agent i's final-endpoint as $p_{\boldsymbol{\Theta}_{\text{ss}}}(f_{\boldsymbol{\Psi}}(\mathcal{P}_i, \boldsymbol{X}, \boldsymbol{A}_f))$ and, given the ground-truth end-point E_i, its success or failure label is c_i, then the pretext loss can be formulated as:

$$\boldsymbol{\Psi}^\star, \boldsymbol{\Theta}_{\text{ss}}^\star = \underset{\boldsymbol{\Psi}, \boldsymbol{\Theta}_{\text{ss}}}{\arg\min}\, \mathcal{L}_{\text{ce}}\Big(p_{\boldsymbol{\Theta}_{\text{ss}}}(f_{\boldsymbol{\Psi}}(\mathcal{P}_i, \boldsymbol{X}, \boldsymbol{A}_f)), c_i\Big) \tag{9}$$

4.2 Learning

As all the modules are differentiable, we can train the model in an end-to-end way. We use the sum of classification, regression and self-supervised losses to train the model. Specifically, we use:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{terminal}} + \mathcal{L}_{\text{ss}} \tag{10}$$

For classification and regression loss design, we adopt the formulation proposed in [2]. $\mathcal{L}_{\text{terminal}} = \frac{1}{N}\sum_{i=1}^{N} L2\Big((x^T_{i,\text{pred}}, y^T_{i,\text{pred}}), (x^T_{i,\text{GT}}, y^T_{i,\text{GT}})\Big)$ is a simple L2 loss that minimizes the distance between predicted final-endpoints and the ground-truth. This is because $\mathcal{L}_{\text{reg}}$ is averaged across all time-points $1 : T$, and from a practical end user perspective, minimizing the endpoint loss is much more important than weighting loss from all time-steps equally. Our proposed pretext tasks contribute to $\mathcal{L}_{\text{ss}}$. During evaluation, we study each pretext task separately, and their corresponding loss formulations defined in Eq. (6), Eq. (7), Eq. (8), Eq. (9) are used for joint training.

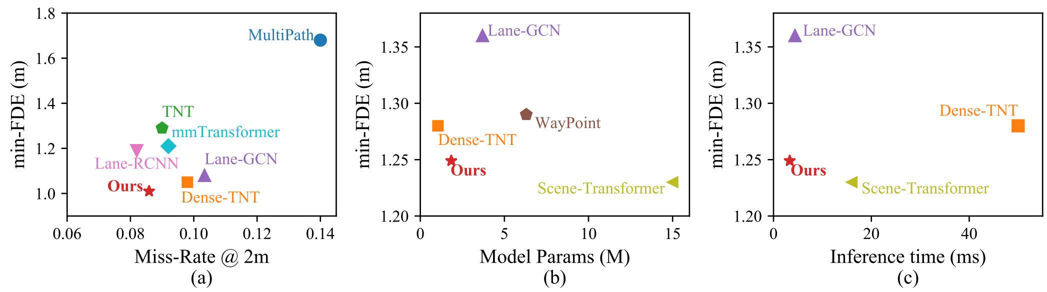

Figure 3: (a) min-FDE_6 - Miss-Rate_6 trade-off on Argoverse Validation. Lower-left is better. We optimize both successfully in comparison to other popular approaches. (b) and (c) We plot min-FDE on Argoverse Test against number of model parameters (in millions) and inference time (in milli-seconds). We find that there is a trade-off between min-FDE performance, architectural complexity (as measured by number of parameters) and computational efficiency (as measured by inference time). Our work achieves the best trade-off (lower-left).

5 Experiments

Dataset: Argoverse provides a large-scale dataset, where the task is to forecast 3 seconds of future motions, given 2 seconds of past observations. It has more than 300K real-world driving sequences collected in Miami (MIA) and Pittsburgh (PIT). Those sequences are further split into train, validation, and test sets, without any geographical overlap. Each of them has 205,942, 39,472, and 78,143 sequences respectively. In particular, each sequence contains the positions of all actors in a scene within the past 2 seconds history, annotated at 10Hz. It also specifies one actor of interest in the scene, with type 'agent', whose future 3 seconds of motion are used for the evaluation. The train and validation splits additionally provide future locations of all actors within 3 second horizon labeled at 10Hz, while annotations for test sequences are withheld from the public and used for the leaderboard evaluation. HD map information is available for all sequences. We have two main requirements for the dataset: (a) **Scale of Data:** Modern motion forecasting methods and self-supervised learning systems require a large amount of training data to imitate human maneuvers in complex real-world scenarios. Thus, the dataset should be large-scale and diverse, such that it has a wide range of behaviors and trajectory shapes across different geometries represented in the data. (b) **Interesting Scenarios for Forecasting Evaluation:** The dataset should be collected for interesting behaviours by biasing sampling towards complex observed behaviours (e.g., lane changes, turns) and road features (e.g., intersections), since we wish to focus on these cases. We find that on the basis of these requirements, as well as its popularity in the the motion forecasting community, Argoverse [1] is the best candidate to showcase our method. Please refer to the supplementary for more details regarding why we choose to focus on it in comparison to other motion forecasting benchmarks.

Metrics: ADE is defined as the average displacement error between ground-truth trajectories and predicted trajectories over all time steps. FDE is defined as displacement error between ground-truth trajectories and predicted trajectories at the final time step. We compute K likely trajectories for each scenario with the ground truth label, where $K = 1$ and $K = 6$ are used. Therefore, minADE and minFDE are minimum ADE and FDE over the top K predictions, respectively. Miss rate (MR) is defined as the percentage of the best-predicted trajectories whose FDE is within a threshold (2 m). Brier-minFDE is the minFDE plus $(1 - p)^2$, where p is the corresponding trajectory probability.

Experimental Details: To normalize the data, we translate and rotate the coordinate system of each sequence so that the origin is at current position $t = 0$ of 'agent' actor and x-axis is aligned with its current direction, i.e., orientation from the agent location at $t = -1$ to the agent location at $t = 0$ is the positive x axis. We use all actors and lanes whose distance from the agent is smaller than 100 meters as the input. We train the model on 4 TITAN-X GPUs using a batch size of 128 with the Adam [44] optimizer with an initial learning rate of 1×10^{-3}, which is decayed to 1×10^{-4} at

100,000 steps. The training process finishes at 128,000 steps and takes about 10 hours to complete. We provide more implementation details in the supplementary.

6 Results

6.1 Ablation Studies

We first examine the effect of incorporating our proposed pretext tasks (Sec. 4) with the standard data-driven motion forecasting baseline (Sec. 3). While evaluating the importance of our proposed pretext tasks, we wish to underline that motion prediction for autonomous driving is a safety-critical task, especially at intersections where most of our data is collected, and most accidents also happen. We thus posit that in this situation, even a small error in predicting final locations (FDE) for a given agent can lead to dangerous potential collision scenarios. Results in Tab. 2 show that all proposed pretext tasks improve motion forecasting performance for Argoverse. Specifically, the Lane Mask pretext task improves min-FDE by 8.9% and MR@2m by 20.3%. Distance to Intersection improves min-FDE by 7.1% and 19.3%. Maneuver classification improves min-FDE by 6.3% and MR@2m by 15.4%. We expect that improving the quality of clustering for maneuvers and thus creating better pseudo-labels will improve this further. Finally, Success/Failure classification improves min-FDE by 9.8% and, perhaps expectedly, MR@2m by 22.4%. Moreover, since pretext tasks are not used for inference and only for training, they also do not add any extra parameters or FLOPs to the baseline, thereby increasing accuracy but at no cost to computational efficiency or architectural complexity.

6.2 Comparison with State-of-the-Art

Performance: We compare our approach with top entries on Argoverse [1] in Tab. 3. SSL-Lanes improves the metrics for $K = 1$ convincingly and outperforms existing approaches w.r.t. *min-ADE*$_1$, *min-FDE*$_1$ and *MR*$_1$. We are strongly competitive w.r.t. *min-ADE*$_6$, *min-FDE*$_6$ and *MR*$_6$. with a relatively simple architecture.

Trade-off between min-FDE and Miss-Rate: *min-FDE*$_6$ and *MR*$_6$ are both important for autonomous robots to optimize. Ideally we wish for both of these metrics to be low. However, there exists a frequent trade-off between them. We compare this trade-off in Fig. 3(a) w.r.t 6 other popular motion forecasting models (in terms of citations and GitHub stars), namely: Lane-GCN [2], Lane-RCNN [2], MultiPath [3], mm-Transformer [9], TNT [5] and Dense-TNT [6] on the Argoverse Validation Set. We are on the lowest-left of Fig. 3(a), meaning we optimize both *min-FDE*$_6$ and *MR*$_6$ successfully in comparison to other top models.

Trade-off between accuracy, efficiency and complexity: We are the first to point out a trade-off that exists for current state-of-the-art motion forecasting models between forecasting performance, architectural complexity and inference speed, in this work. This is illustrated in Fig. 3(b, c). In contrast to the popular models, our approach has high accuracy (*min-FDE*$_6$: 1.25m, *MR*$_6$: 13.3%), while also having low architectural complexity (1.84M parameters) and high inference speed (3.3 ms). Thus it provides a great balance for application to real-time safety-critical autonomous robots.

Qualitative Results: We present some multi-modal prediction trajectories on several hard cases shown in Fig. 1. The yellow trajectory represents the observed 2s. Red represents ground truth for the next 3s and green represents the multiple forecasted trajectories for those 3s. In Row 1, the agent turns right at the intersection. The baseline misses this mode completely, despite having access to the map. The model trained with lane-masking successfully predicts this right turn within 2m of the ground-truth end-point. In Row 2, the agent has a noisy past history and accelerates while turning left at the intersection. The pretext task distance-to-intersection can correctly capture this, while the baseline has only one trajectory covering this mode but vastly overshoots the ground-truth. Interestingly, we note that the success/failure pretext task is unable to capture this mode. We believe this is due to a stronger prior imposed by the model during learning. In Row 3, we have an agent accelerating while going straight at an intersection. We find that the maneuver classification pretext task is the only model that correctly predicts trajectories aligned with the ground-truth. In Row 4,

Description	Experimental Setup: Training	Experimental Setup: Validation	Method	minADE$_6$	minFDE$_6$	MR$_6$
Effects of limited training data	25% of train	All	Baseline Ours	0.82 **0.78**	1.33 **1.22**	14.66 **12.63**
Effects of new domain	100% PIT + 20% MIA	MIA val	Baseline Ours	0.88 **0.85**	1.46 **1.34**	17.21 **14.96**
Performance on difficult maneuvers	All	Turning & lane changing	Baseline Ours	0.90 **0.84**	1.53 **1.34**	19.90 **14.93**
Effects of imbalanced data	2x straight 1x other maneuvers	Turning & lane changing	Baseline Ours	0.94 **0.90**	1.65 **1.49**	21.53 **17.97**
Effects of noisy data	All	Gaussian noise ($\sigma = 0.2$) with $p = 0.25$	Baseline Ours	1.01 **0.96**	1.37 **1.24**	15.59 **11.98**
Effects of noisy data	All	Gaussian noise ($\sigma = 0.2$) with $p = 0.5$	Baseline Ours	1.19 **1.13**	1.56 **1.40**	20.64 **15.65**

Table 4: Different experimental settings for SSL-based training

we have an agent turning left at an intersection. Most of the predictions of other models predicts that the agent will go straight. The success/failure pretext task however picks up on the left-turn, possibly due to the priors imposed upon it by end-point conditioning.

Overall, SSL-Lanes can capture left and right turns better, while also being able to discern acceleration at intersections. Our pretext tasks provide priors for the model and provides data-regularization for free. We believe this can improve forecasting through better understanding of map topology, agent context with respect to the map, and generalization with respect to imbalance implicitly present in data.

6.3 When does SSL help Motion Forecasting?

We design 6 different training and testing setups as shown in Tab. 4. We use Success/Failure classification as the pretext task, and all models are trained for 50,000 steps. We initialize the map-encoder with the parameters from a model trained with the lane-masking pretext task.

We hypothesize that training with SSL pretext tasks helps motion forecasting in the following ways: (a) Topology-based context prediction assumes feature similarity or smoothness in small neighborhoods of maps, and the resulting feature representation may improve prediction performance. This is mainly expected to help in the first and second settings, which requires generalizing to new topologies. (b) Clustering and classification assumes that feature similarity implies target-label similarity and can group distant nodes with similar features together, leading to better generalization. This is mainly expected to help with dataset imbalance and performance on difficult maneuvers, which requires generalizing to hard cases. (c) Supervised learning with imbalanced datasets sees significant degradation in performance. Although most of the data samples in Argoverse are at an intersection, a significantly large number involve driving straight while maintaining speed. Recent studies [45] have shown that SSL tends to learn richer features from more frequent classes, which also allows it to generalize to other classes better. We expect this to help with imbalanced data, limited training data and noisy data.

SSL leads to better generalization compared to pure supervised learning: To provide evidence for our hypotheses, we design 6 different training and testing setups as shown in Tab. 4. We use Success/Failure classification as the pretext task, and all models are trained for 500,000 steps. We initialize the map-encoder with the parameters from a model trained with the lane mask pretext task. Our *first* setting is to train with 25% of the total data available for training and testing on the full validation set. Our *second* setting assumes that SSL also generalizes to topology from different cities and trains on 100% of data from Pittsburgh (PIT) but only 20% of data from Miami (MIA). For evaluation, we only test on data examples taken from the city of MIA. For our *third* setting, we assume that SSL learns superior features and can thus perform better in difficult cases like lane-changes and turning cases. For evaluation, we only test on data examples which involves these difficult cases. In our *fourth* setting, we choose to explicitly train with data that contains $2\times$ 'straight-with-same-speed' maneuver and $1\times$ all other maneuvers. We test only on lane-changes and turning cases from validation. Finally in order to test the effect of noise on motion forecasting performance, we take two models already trained on full data. We now take the full validation set, randomly select agent trajectories or map nodes with probability $p = 0.25$ and $p = 0.5$, and then add Gaussian noise with zero mean and 0.2 variance to their features. There is strong evidence from our experiments that SSL-based tasks provide better generalization and can thus be more effective than pure supervised training.

7 Discussion: Potential of this Work

We expect this work to influence real world deployment of SSL forecasting methods for autonomous driving. Another use case for this work is realistic behavior generation in traffic simulation. The general construction of the prediction problem, inspired by [2], enables a generic understanding of how an object moves in a given environment without memorizing the training data. A neural network may learn to associate particular areas of a scene with certain motion patterns. To prevent this, we centre around the agent of interest and normalize all other trajectory and map coordinates with respect to it. We predict relative motion as opposed to absolute motion for the future trajectory. This helps to learn general motion patterns. Reconstructing the map or predicting distances from map elements are conducted in a frame-of-reference relative to the agent of interest. This helps in learning general map connectivity. Following work in pedestrian trajectory prediction, we also additionally add random rotations to the training trajectories to reduce directional bias. Furthermore, we provide strong evidence that SSL-based tasks provide better generalization compared to pure supervised training, thereby having the ability to effectively reuse the same prediction model across different scenarios.

8 Conclusion

We propose SSL-Lanes to leverage supervisory signals generated from data for free in the form of pseudo-labels and integrate it with a standard motion forecasting model. We design four pretext tasks that can take advantage of map-structure and similarities between agent dynamics to generate these pseudo-labels, namely: lane masking, distance to intersection prediction, maneuver classification and success/failure classification. We validate our proposed approach by achieving competitive results on the challenging large-scale Argoverse benchmark. The main advantage of SSL-Lanes is that it has high accuracy combined with low architectural complexity and high inference speed. We further demonstrate that each proposed SSL pretext task improves upon the baseline, especially in difficult cases like left/right turns and acceleration/deceleration. We also provide hypotheses and experiments on why SSL-Lanes can improve motion forecasting.

Limitations: A limitation of our framework is that it uses the different losses for our formulation only in a 1:1 ratio without tuning them. We also use only one pretext task at a time and do not explore the combination of these different tasks. For our future work, we plan to incorporate meta-learning [46] to identify an effective combination of pretext tasks and automatically balance them—we expect that this will lead to more gains in terms of forecasting performance. Another limitation is that we report improvements with SSL-pretext tasks in scenarios without specifically considering multiple heavily interacting agents. In the future we would like to explore how the interactions between road agents can influence our SSL losses on the interaction split of the Waymo Open Motion dataset (WOMD) [47]. Finally, we explore generalization in terms of implicit data imbalance only in comparison to pure supervised training on the same dataset from which training samples are derived. We would like to study the generalization of our work to other datasets without re-training.

References

[1] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8748–8757. Computer Vision Foundation / IEEE, 2019. doi:10.1109/CVPR.2019.00895. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Chang_Argoverse_3D_Tracking_and_Forecasting_With_Rich_Maps_CVPR_2019_paper.html. 2, 6, 7, 8

[2] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 1, 2, 4, 6, 8, 10

[3] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 86–99. PMLR, 2019. URL http://proceedings.mlr.press/v100/chai20a.html. 1, 2, 3, 8

[4] J. Mercat, T. Gilles, N. E. Zoghby, G. Sandou, D. Beauvois, and G. P. Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 9638–9644. IEEE, 2020. doi:10.1109/ICRA40945.2020.9197340. URL https://doi.org/10.1109/ICRA40945.2020.9197340. 1, 2, 6

[5] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov. TNT: target-driven trajectory prediction. In J. Kober, F. Ramos, and C. J. Tomlin, editors, *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pages 895–904. PMLR, 2020. URL https://proceedings.mlr.press/v155/zhao21b.html. 1, 2, 6, 8

[6] J. Gu, C. Sun, and H. Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15283–15292. IEEE, 2021. doi:10.1109/ICCV48922.2021.01502. URL https://doi.org/10.1109/ICCV48922.2021.01502. 1, 2, 6, 8

[7] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan. What-if motion prediction for autonomous driving. *CoRR*, abs/2008.10587, 2020. URL https://arxiv.org/abs/2008.10587. 1, 2, 6

[8] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. Weiss, B. Sapp, Z. Chen, and J. Shlens. Scene transformer: A unified multi-task model for behavior prediction and planning. *CoRR*, abs/2106.08417, 2021. URL https://arxiv.org/abs/2106.08417. 1, 2, 6

[9] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou. Multimodal motion prediction with stacked transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7577–7586. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Liu_Multimodal_Motion_Prediction_With_Stacked_Transformers_CVPR_2021_paper.html. 1, 2, 6, 8

[10] S. Casas, W. Luo, and R. Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, volume 87 of *Proceedings of Machine Learning Research*, pages 947–956. PMLR, 2018. URL http://proceedings.mlr.press/v87/casas18a.html. 1, 2

[11] W. Zeng, M. Liang, R. Liao, and R. Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pages 532–539. IEEE, 2021. doi:10.1109/IROS51168.2021.9636035. URL https://doi.org/10.1109/IROS51168.2021.9636035. 1, 2, 6

[12] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. Vectornet: Encoding HD maps and agent dynamics from vectorized representation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11522–11530. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01154. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Gao_VectorNet_Encoding_HD_Maps_and_Agent_Dynamics_From_Vectorized_Representation_CVPR_2020_paper.html. 1, 2, 3

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. 1

[14] Z. Huang, X. Mo, and C. Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. *CoRR*, abs/2109.06446, 2021. URL https://arxiv.org/abs/2109.06446. 1, 2, 6

[15] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *10th International Conference on Learning Representations, ICLR 2022, Conference Track Proceedings*, 2022. 1, 2, 6

[16] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021. doi:10.1109/ICCV48922.2021.00951. URL https://doi.org/10.1109/ICCV48922.2021.00951. 1

[17] R. E. Kalman and Others. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 2

[18] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang. Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. *IEEE Trans. Ind. Electron.*, 65(7):5999–6008, 2018. doi:10.1109/TIE.2017.2782236. URL https://doi.org/10.1109/TIE.2017.2782236. 2

[19] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 4363–4369. IEEE, 2013. doi:10.1109/IROS.2013.6696982. URL https://doi.org/10.1109/IROS.2013.6696982. 2

[20] M. Bansal, A. Krizhevsky, and A. S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In A. Bicchi, H. Kress-Gazit, and S. Hutchinson, editors, *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, 2019. doi:10.15607/RSS.2019.XV.031. URL https://doi.org/10.15607/RSS.2019.XV.031. 2

[21] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14062–14071. Computer Vision Foundation / IEEE, 2020. doi:10.1109/CVPR42600.2020.01408. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Phan-Minh_CoverNet_Multimodal_Behavior_Prediction_Using_Trajectory_Sets_CVPR_2020_paper.html. 2

[22] A. A. Mohamed, K. Qian, M. Elhoseiny, and C. G. Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14412–14420. Computer Vision Foundation / IEEE, 2020. doi:10.1109/CVPR42600.2020.01443. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Mohamed_Social-STGCNN_A_Social_Spatio-Temporal_Graph_Convolutional_Neural_Network_for_Human_CVPR_2020_paper.html. 2

[23] M. Ye, T. Cao, and Q. Chen. TPCN: temporal point cloud networks for motion forecasting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11318–11327. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Ye_TPCN_Temporal_Point_Cloud_Networks_for_Motion_Forecasting_CVPR_2021_paper.html. 2, 6

[24] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen. Learning to predict vehicle trajectories with model-based planning. In A. Faust, D. Hsu, and G. Neumann, editors, *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pages 1035–1045. PMLR, 2021. URL https://proceedings.mlr.press/v164/song22a.html. 2, 6

[25] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. End-to-end interpretable neural motion planner. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8660–8669. Computer Vision Foundation / IEEE, 2019. doi:10.1109/CVPR.2019.00886. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Zeng_End-To-End_Interpretable_Neural_Motion_Planner_CVPR_2019_paper.html. 2

[26] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde. HOME: heatmap output for future motion estimation. In *24th IEEE International Intelligent Transportation Systems Conference, ITSC 2021, Indianapolis, IN, USA, September 19-22, 2021*, pages 500–507. IEEE, 2021. doi:10.1109/ITSC48978.2021.9564944. URL https://doi.org/10.1109/ITSC48978.2021.9564944. 2, 6

[27] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. Lam, D. Anguelov, and B. Sapp. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. *CoRR*, abs/2111.14973, 2021. URL https://arxiv.org/abs/2111.14973. 3

[28] M. Ye, J. Xu, X. Xu, T. Cao, and Q. Chen. DCMS: motion forecasting with dual consistency and multi-pseudo-target supervision. *CoRR*, abs/2204.05859, 2022. doi:10.48550/arXiv.2204.05859. URL https://doi.org/10.48550/arXiv.2204.05859. 3

[29] N. Deo and M. M. Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*, pages 1179–1184. IEEE, 2018. doi:10.1109/IVS.2018.8500493. URL https://doi.org/10.1109/IVS.2018.8500493. 3

[30] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago,*

Chile, December 7-13, 2015, pages 1422–1430. IEEE Computer Society, 2015. doi:10.1109/ICCV.2015.167. URL `https://doi.org/10.1109/ICCV.2015.167`. 3

[31] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2016. doi:10.1007/978-3-319-46466-4_5. URL `https://doi.org/10.1007/978-3-319-46466-4_5`. 3

[32] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=S1v4N2l0-`. 3

[33] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, volume 11218 of *Lecture Notes in Computer Science*, pages 139–156. Springer, 2018. doi:10.1007/978-3-030-01264-9_9. URL `https://doi.org/10.1007/978-3-030-01264-9_9`. 3

[34] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2536–2544. IEEE Computer Society, 2016. doi:10.1109/CVPR.2016.278. URL `https://doi.org/10.1109/CVPR.2016.278`. 3

[35] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, pages 649–666. Springer, 2016. doi:10.1007/978-3-319-46487-9_40. URL `https://doi.org/10.1007/978-3-319-46487-9_40`. 3

[36] D. Pathak, R. B. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6024–6033. IEEE Computer Society, 2017. doi:10.1109/CVPR.2017.638. URL `https://doi.org/10.1109/CVPR.2017.638`. 3

[37] Y. You, T. Chen, Z. Wang, and Y. Shen. When does self-supervision help graph convolutional networks? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10871–10880. PMLR, 2020. URL `http://proceedings.mlr.press/v119/you20a.html`. 3

[38] W. Jin, T. Derr, H. Liu, Y. Wang, S. Wang, Z. Liu, and J. Tang. Self-supervised learning on graphs: Deep insights and new direction. *CoRR*, abs/2006.10141, 2020. URL `https://arxiv.org/abs/2006.10141`. 3

[39] Y. Liu, S. Pan, M. Jin, C. Zhou, F. Xia, and P. S. Yu. Graph self-supervised learning: A survey. *CoRR*, abs/2103.00111, 2021. URL `https://arxiv.org/abs/2103.00111`. 3

[40] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. S. Pande, and J. Leskovec. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=HJlWWJSFDH`. 3

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. 4

[42] E. A. Abolfathi, M. Rohani, E. Banijamali, J. Luo, and P. Poupart. Self-supervised simultaneous multi-step prediction of road dynamics and cost map. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8494–8503. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Amirloo_Self-Supervised_Simultaneous_Multi-Step_Prediction_of_Road_Dynamics_and_Cost_Map_CVPR_2021_paper.html. 5

[43] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In C. E. Brodley and A. P. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 577–584. Morgan Kaufmann, 2001. 5

[44] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980. 7

[45] H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma. Self-supervised learning is more robust to dataset imbalance. *CoRR*, abs/2110.05025, 2021. URL https://arxiv.org/abs/2110.05025. 9

[46] D. Hwang, J. Park, S. Kwon, K. Kim, J. Ha, and H. J. Kim. Self-supervised auxiliary learning with meta-paths for heterogeneous graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/74de5f915765ea59816e770a8e686f38-Abstract.html. 10

[47] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9690–9699. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00957. URL https://doi.org/10.1109/ICCV48922.2021.00957. 10