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

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

Abstract

Predicting diverse yet admissible trajectories that adhere to the map constraints is challenging. Graph-based scene encoders have been proven effective for preserving local structures of maps by defining lane-level connections. However, such encoders do not capture more complex patterns emerging from long-range heterogeneous connections between nonadjacent interacting lanes. To this end, we shed new light on learning common driving patterns by introducing meTA ROad paTh (TAROT) to formulate combinations of various relations between lanes on the road topology. Intuitively, this can be viewed as finding feasible routes. Furthermore, we propose MEta-road NeTwORk (MENTOR) that helps trajectory prediction by providing it with TAROT as navigation tips. More specifically, 1) we define TAROT prediction as a novel self-supervised proxy task to identify the complex heterogeneous structure of the map. 2) For typical driving actions, we establish several TAROTs that result in multiple Heterogeneous Structure Learning (HSL) tasks. These tasks are used in MENTOR, which performs meta-learning by simultaneously predicting trajectories along with proxy tasks, identifying an optimal combination of them, and automatically balancing them to improve the primary task. We show that our model achieves state-of-the-art performance on the Argoverse dataset, especially on diversity and admissibility metrics, achieving up to 20% improvements in challenging scenarios. We further investigate the contribution of proposed modules in ablation studies.

1. Introduction

Vehicle trajectory prediction is one of the main building blocks of autonomous driving. Prediction demonstrates how the future might unfold based on the road structure and the behavior of road users. To operate safely, self-driving cars must accurately perceive the geometric and semantic information in the driving scene and predict diverse yet scene-compliant trajectories. High-definition maps provide useful cues since the behaviors of agents and the interactions among them are largely influenced by the road topology and governed by the map constraints. For example, a vehicle is unlikely to change to a lane that runs in the opposite direction. In this light, learning the structure of the map efficiently is essential to predict a variety of feasible trajectories.

Inspired by the structural nature of the road, map vectorization has become prevalent in recent works, in which lanes are converted into vectors [15, 56, 18]. Consequently, graph structures can be leveraged to learn the relationships between vectorized entities [28, 54, 11, 16]. In this way, connected lanes in various formations, e.g. successor, predecessor, right, and left, represent admissible ways to traverse the terrain. For example, two lanes positioned consecutively in space and connected by a sequential link can be driven successively, whereas two lanes connected by a lateral link enable a lane change maneuver. Hence, the HD map lane graph has edges with different semantic meanings, indicating its heterogeneous nature.

Although existing lane graphs are shown to be effective in learning local structures through the adjacent lanes, they lack the ability to model more complex patterns imposed by long-range heterogeneous connections between nonadjacent yet interacting lanes. These connections are important to capture high-level intentions, such as overtaking, merging, and double turns, and can potentially represent the constraints and rules of the road. To model such connections, some methods propose adopting self-supervised tasks on top of the lane graph [15, 4] to encourage the model to learn the semantics associated with lane connectivities. Since scene structure contains various semantics, having only one proxy task at a time is insufficient to provide diverse traffic patterns. Alternatively, anchor-based models incorporate prior knowledge by identifying candidate trajectories based on sampling or clustering of training data [6, 43]. However, these methods are less robust and generalizable due to their dependency on the distribution of data.

Motivated by this insight, we introduce a novel self-supervised task for trajectory prediction, termed Heteroge-

neous Structure Learning (HSL), to formulate traffic patterns imposed by road constraints. In particular, we formulate the road structure by meTA ROad paTh (TAROT) as a composite relation of heterogeneous edges in the scene graph that can model diverse abstractions from the map without additional data and labels. Therefore, by predicting the presence of TAROT in the driving scene as an HSL task, the model gets a sense of feasible transitions on the map as navigation tips for traversing the road. Furthermore, we propose MEta-road NeTwORk (MENTOR), a novel trajectory prediction framework that provides insight into the structural perspective of the map by simultaneously predicting a set of HSL tasks. In detail, inspired by the meta-learning concept, MENTOR is learning to learn road topology by automatically selecting the HSL tasks that assist the target trajectory prediction.

The **contributions** of this paper are as follows: (1) For the first time, to the best of our knowledge, we investigate the idea of learning to learn in trajectory prediction through the proposed MEta-road NeTwORk (MENTOR), which determines an optimal combination of Heterogeneous Structure Learning (HSL) tasks and automatically balances them to assist the primary task of predicting diverse and admissible trajectories. (2) We propose meTA ROad paTh (TAROT) prediction as a novel HSL task capable of modeling high-level semantics within the map structure. Thus, diverse semantic aspects of road configuration can be captured by defining various TAROTs, resulting in multiple HSL tasks that are learned simultaneously. (3) We conduct extensive experimental evaluations to show our model's effectiveness in predicting not only precise but also diverse and admissible trajectories.

2. Related Work

Trajectory Prediction has been the subject of interest in a large body of literature. Existing methods explore different representations and methods to learn scene context, including rasterized images [6, 38, 16] and point-clouds [49] with CNNs, and vectorized representations with GNNs [15, 32, 18, 54], or transformers [21, 58, 17, 30], which are often associated with sophisticated fusion mechanisms [28]. These models focus on finding more powerful ways of extracting features from HD maps and modeling agents' interactions. They, however, require substantial model capacity, heavy parameterization, extensive augmentations, or a large amount of data to produce accurate results.

Some methods investigate the ways to improve prediction precision by incorporating prior knowledge as predefined candidate trajectories generated using sampling or clustering techniques [32, 55]. Nevertheless, the main limitation of these methods is that their performance is highly dependent on the quality of trajectory proposals, which becomes a dependency. To resolve this issue, end-to-end solutions optimize the endpoints of candidate trajectories [18, 16] resulting in improved accuracy while maintaining a reasonable parameter budget. However, using an online goal candidate optimization strategy is computationally very expensive, therefore it is prohibitive for real-time applications, such as autonomous driving. More recently, ensemble techniques have become more prevalent and demonstrated good prediction performance [43, 50]. However, their major disadvantages are high memory cost for training and computationally expensive inference.

Besides designing high-precision models, researchers have focused on diversity, admissibility, and map conformity of the predicted trajectories. To generate diverse trajectories, state-of-the-art models use Generative Adversarial Networks (GANs) [25, 37, 57], and Conditional Variational Autoencoders (CVAEs) [38, 42, 53, 26], sophisticated sampling functions, such as diversity sampling [52] and latent semantic sampling [20]. However, most works sacrifice map conformity for prediction diversity. To improve conformity, map data can be incorporated as a part of the loss function [44, 9], or as a post-processing step [48]. To better evaluate the diversity and admissibility of predicted trajectories, new metrics, such as RF and DAO, have been proposed [34]. In this paper, we use these as well as common metrics to further assess our model's performance.

Self-Supervised Learning (SSL) is a learning method that uses appropriately designed proxy tasks to generate additional supervised signals to train models. In trajectory prediction, SSL has been utilized by defining various proxy tasks laid on top of the agent/map information [15, 4]. For instance, a recent work, SSL-Lane [4], uses four different tasks and shows how they can be incorporated to improve performance. However, this model uses only one proxy task at a time and does not explore combinations of different tasks. Additionally, SSL-Lane employs different losses in its formulation only in a 1:1 ratio without tuning them. These shortcomings motivated our proposed framework for simultaneously learning proxy tasks without letting them compete with the target trajectory prediction task.

Meta-learning originates from the concept of learning to learn, a topic of continued interest in machine-learning research. Meta-learning has been used for finding optimal hyper-parameters and good initialization [27, 13], metric-learning, and non-parametric methods for few-shot learning [40, 41]. This paper uses meta-learning to learn the road topology across several tasks by transferring the knowledge from self-supervised tasks, defined on the map structure, to the primary trajectory prediction task. In a closely related category of works, sample weighting, a meta-learning mechanism is used for adaptively extracting sample weights to guarantee robust deep learning in the presence of training data bias [23, 39]. In practice, instead of manually setting the form of weight functions, these methods learn an ex-

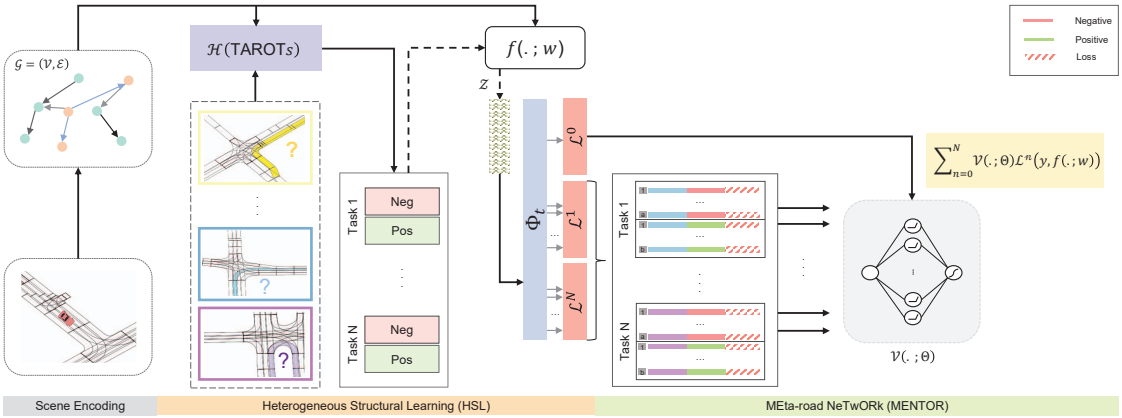

Figure 1: Overview of the proposed approach. Given directed HIN $\mathcal{G}$ of the scene and N predefined TAROTs, the HSL engine $\mathcal{H}$(TAROTs) defines N HSL tasks each predicting the presence of a specific TAROT. Within each TAROT task p_n, positive samples are two arbitrary nodes from $\mathcal{G}$ that can be reached by p_n, and negative samples are those that cannot. For the HSL tasks and the primary task, we find learned representations $\mathcal{Z}$ from GNN model f. To find the loss values, samples are fed through the task-specific transformations Φ_t followed by task-specific loss functions $\mathcal{L}^n$. Lastly, by Providing loss values of the primary task's samples and the HSL tasks' samples (as a concatenation of [task id, label(Pos/Neg), loss value]), we train MENTOR to learn weight function $\mathcal{V}$ in a way to find an optimal combination of the HSL tasks in a meta-learning manner.

plicit weighting function directly from the data.

Self-supervised learning and meta-learning both contribute to improving model generalization capability. To the best of our knowledge, this is the first study in trajectory prediction to exploit the benefit of scene-based self-supervised tasks through a meta-learning paradigm.

3. Proposed Method

Traffic transitions are complex and cannot simply be modeled by learning the local relationships. What required is modeling patterns of traveling through the road as a combination of relationships, not only in the adjacent lanes but also between nonadjacent yet interacting lanes. This can, for example, refer to a merge pattern in the road structure that involves both lateral and sequential relations.

The main idea behind our proposed model is to provide high-order combinations of basic adjacent-adjacent patterns as the full gamut of lane interactions when predicting vehicle trajectories. To this end, we first discuss the procedure for generating a Heterogeneous Information Network (HIN) from scenes, followed by the introduction of the Heterogeneous Structural Learning (HSL) tasks designed to capture relations between lanes. Next, we describe our novel MEta-road NeTwORk (MENTOR) that learns an explicit weighting function to softly select HSL tasks and balance them to improve the performance of the primary task via meta-learning (see Fig. 1).

3.1. Scene Encoding

Contrary to the common approach that treats the scene as a homogeneous graph [54], we model the scene as a Heterogeneous Information Network (HIN) [24].

Definition 1. Heterogeneous Information Network (HIN) a HIN is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges, each representing a binary relation between two nodes in $\mathcal{V}$. $\mathcal{G}$ is associated with two mappings: (1) node type mapping function $\phi : \mathcal{V} \to \mathcal{T}$ and (2) edge type mapping function $\psi : \mathcal{E} \to \mathcal{R}$, where $\mathcal{T}$ and $\mathcal{R}$ denote sets of node and edge types, respectively. If $|\mathcal{T}| + |\mathcal{R}| > 2$, network $\mathcal{G}$ is an HIN, otherwise it is homogeneous.

We encode the scene as a directed HIN with node types $\mathcal{T} = \{\text{lane}, \text{agent}\}$ and $\mathcal{R} = \{\text{left}, \text{right}, \text{successor}, \text{predecessor}\}$ as basic relations between adjacent lanes. To initialize node features in the HIN, we use the simplified PointNet [36] with multi-layer perception (MLP) to process polyline features and a 1D convolution with a feature pyramid network [29] to process agents' observations similar to [28]. One advantage of our method is that any type of GNN can be used to process the HIN. Hence, we employ Heterogeneous Graph Transformer (HGT) as a simple yet effective model similar to [19].

3.2. Heterogeneous Structural Learning

Traffic transitions follow patterns governed by high-level constraints stemming from long-range structural connections between lanes of the road. For learning the map structure, heterogeneity helps interpret a series of relations,

which our model learns with meTA ROad paTh (TAROT).

Definition 2. Meta-Path. A meta-path $\mathcal{P} = v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} \ldots \xrightarrow{r_\ell} v_{\ell+1}$ expresses a composite relation $R = r_1 \circ r_2 \ldots \circ r_\ell$ between nodes v_1 and $v_{\ell+1}$, where $\circ$ denotes the composition operator on relations. If two nodes v_i and v_j are related by the composite relation R, then there exists a path that connects v_i to v_j in $\mathcal{G}$, denoted by $\mathcal{P}_{v_i \rightsquigarrow v_j}$. Moreover, the nodes and edges types in $\mathcal{P}_{x_i \rightsquigarrow x_j}$ match with types in $\mathcal{T}$ and $\mathcal{R}$, respectively.

The features obtained from the meta-path analysis in HINs are notably useful in improving graph-based models since they encode indirect semantic relations between nodes that are not directly connected. These features also aid in establishing new relations between vertices. As such, meta-path analysis has been frequently used in graph domains to enhance graph representation learning power and improve downstream tasks, such as node classification and link prediction [51, 46, 22, 8]. This means meta-paths can help find new patterns and relations in HINs.

By focusing on the long-range relations that impose transitions on the road, we define TAROT p though the HIN as a path instance of $p_{v_i \rightsquigarrow v_j} \vdash \mathcal{P}$ where node types are the same. Thus, the TAROT can formulate a combination of basic road relations corresponding to transition patterns imposed by map constraints. Predicting the presence of a TAROT between two nodes v_i and v_j is similar to link prediction. Here, the link means heterogeneous composite relations between nodes. Contrary to link prediction problems, the proposed TAROT prediction can be treated as a self-supervised task. In detail, we follow a simple rule to provide the additional supervised signals for proxy tasks: *If node v is reachable from node u by TAROT p, then $y^p_{u,v} = 1$, otherwise $y^p_{u,v} = 0$*. This means the HSL task, associated with the TAROT will be formed as a simple link prediction task without the need for any additional data or manual labeling. Hence, by obtaining the hidden representations of two arbitrary nodes learned by a GNN and using a simple operator $\hat{y}^p_{u,v} = \sigma\left(\Phi_t\left(z_u\right)^\top \Phi_t\left(z_v\right)\right)$ we can predict whether a TAROT between two nodes is present or absent. Here, Φ_t is a type-specific transformation $t \in \mathcal{T}$ and z_u and z_v are the node embeddings of nodes u and v.

Using the proposed TAROT as a type of meta-path we can model a common pattern of driving that adhere to the map constraints. Hence, we define several TAROTs to formulate diverse traffic patterns within the driving scene, resulting in several HSL tasks that can be used to predict diverse and admissible trajectories. We provided a complimentary figure in the supplementary material to further show TAROT's motivation.

3.3. Meta Road Network

The proposed HSL task is designed to identify diverse patterns within the road structure, for which we explore the idea of defining various TAROTs resulting in several self-supervised HSL tasks to be learned simultaneously with the primary trajectory prediction task. However, as with any multi-task learning method, the HSL tasks should be carefully chosen and properly weighted so that learning the map structure does not compete with the primary task, especially when the capacity of graph neural networks (GNNs) is limited. To this end, we propose MEta-road NeTwORk (MENTOR), a learning framework for trajectory prediction that offers the possibility of learning to learn the road configuration while learning the main task. Below, we discuss a naive multi-task combination of self-supervision tasks to explain the motivation behind the design of MENTOR, followed by describing the meta-learning objective using MENTOR, and a detailed look at the learning procedure.

3.3.1 Naive Multi-task Combination of HSL Tasks

Self-supervised learning has already been explored in trajectory prediction for autonomous driving and has proven beneficial [15, 4]. Adopting a combination of Self-Supervised Learning (SSL) tasks, namely proxy tasks, to improve the primary task has not yet been addressed, even though it has been shown to be effective in visual learning [12]. Here, we formulate a naive multi-task combination of HSL tasks, as SSL tasks, with a shared backbone network with parameters $\mathbf{w}$, between the primary and all proxy tasks. Each task has a specific loss function $\mathcal{L}^n$ incorporated by a task-specific parameter λ_n. So the optimal parameter $\mathbf{w}^*$ can be extracted by the following objective function:

$$\min_{\mathbf{w}} \quad \mathbb{E}\left[\lambda_0 \mathcal{L}^0\left(.;\mathbf{w}\right) + \sum_{n=1}^{N} \lambda_n \mathcal{L}^n\left(.;\mathbf{w}\right)\right] \tag{1}$$

where index zero stands for the primary task $n = 0$, and index $n|_1^N$ denotes n^{th} task of N proxy tasks.

3.3.2 Meta-learning Objective

Learning road topology should enhance rather than compete with trajectory prediction, hence determining an effective combination of λ parameters is crucial. To achieve this, we adopt MENTOR to parameterize model $\mathbf{w}(\mathbf{\Theta})$ to determine an optimal combination of HSL tasks and automatically balance them for improved prediction as shown below,

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{\Theta}} \quad \mathbb{E}\left[\mathcal{L}^0\left(\mathbf{w}^*(\mathbf{\Theta})\right)\right] \\ &\text{s.t. } \mathbf{w}^*(\mathbf{\Theta}) = \underset{\mathbf{w}}{\operatorname{argmin}} \quad \mathbb{E}\left[\mathcal{L}^{n|_0^N}\left(\mathbf{w};\mathbf{\Theta}\right)\right]. \end{aligned} \tag{2}$$

Specifically, to parametrize the model parameters $\mathbf{w}$, we can impose a learnable weighting function $\mathcal{V}(\cdot,\mathbf{\Theta})$ to the objective function in (2) as MENTOR similar to [39]. Thus, the

Algorithm 1: MEta-road NeTwORk (MENTOR)

Input : primary data D^{pr}, HSL proxy tasks data D^{hsl}, batch size n, m, max iteration T, folds for cross-validation K, model parameters $\mathbf{w}$, MENTOR parameters Θ

Output : model parameter $\hat{\mathbf{w}}^T$

Initialize $\mathbf{w}, \mathbf{\Theta}$
for $t = 0$ *to* $T-1$ **do**
  $\left\{x^{(\text{pr})}, y^{(\text{pr})}\right\} \leftarrow \text{SampleMiniBatch}\left(D^{pr}, n\right)$;
  $\left\{x^{(\text{hsl})}, y^{(\text{hsl})}\right\} \leftarrow \text{SampleMiniBatch}\left(D^{\text{hsl}}, m\right)$;
  // cross validation
  for $k = 1$ *to* K **do**
    $\left\{x^{(\text{meta})}, y^{(\text{meta})}\right\}, \left\{x^{(\text{train})}, y^{(\text{train})}\right\} \leftarrow \text{CVSplit}\left(D^{pr}, k\right)$;
    Formulate model parameter $\mathbf{w}^t$ by Eq 6
    Keep $\mathbf{\Theta}_k^{t+1}$ on fold $= k$ by Eq. 7
  end
  Update $\mathbf{\Theta}^{t+1}$ with $\frac{1}{K}\sum_k^K \mathbf{\Theta}_k^{t+1}$
  Update $\mathbf{w}^{t+1}$by Eq 8
end

objective function can be written as follows:

$$\mathbf{w}^*(\mathbf{\Theta}) = \underset{\mathbf{w}}{\text{argmin}} \sum_{n=0}^{N}\sum_{i=1}^{M_n} \frac{1}{M_n}\mathcal{V}\left(\Psi_i^n; \mathbf{\Theta}\right)\mathcal{L}^n\left(y_i^n, f^n\left(x_i^n; \mathbf{w}\right)\right) \tag{3}$$

where each task has M_n samples and f^n is the model for task n. For i^{th} sample of task n, y_i^n is the label, and Ψ_i^n is an embedding vector expressing as the concatenation of one-hot vector of task types, the label (positive/negative), and the loss value of the sample. Therefore, following Eq. (3), the model learns how to assist the primary task by optimizing MENTOR's parameters $\mathbf{\Theta}$. These parameters can be optimized using the meta-learning idea [47, 1, 10, 14]. Specifically, using a small amount of meta-data D^{meta}, representing the meta-knowledge of ground-truth D^{gt}, $D^{\text{meta}} \cup D^{\text{train}} = D^{\text{gt}}$, the optimal parameter $\mathbf{\Theta}^*$ can be obtained by minimizing the following loss:

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmin}}\ \mathcal{L}^{\text{meta}}\left(\mathbf{w}^*(\mathbf{\Theta})\right) \triangleq \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_i^{\text{meta}}\left(\mathbf{w}^*(\mathbf{\Theta})\right) \tag{4}$$

where m and r, $m \ll r$, are the number of meta-samples and training samples, respectively, and meta-loss is,

$$\mathcal{L}_i^{\text{meta}}(\mathbf{w}) = \mathcal{L}\left(y_i^{\text{meta}}, f\left(x_i^{\text{meta}}, \mathbf{w}\right)\right) \tag{5}$$

where $(x, y) \sim D^{\text{meta}}$.

3.3.3 Meta-road Network Learning Procedure

To overcome the complexity of the bi-level optimization, following [13, 39], we use an online strategy to approximate $\mathbf{w}^*$ and $\mathbf{\Theta}^*$ with the updated parameters $\hat{\mathbf{w}}$ and $\hat{\mathbf{\Theta}}$, respectively through a single optimization loop. In each iteration of training, given training data D^{train}, HSL proxy tasks data D^{hsl}, and meta-data D^{meta}, we follow three steps:

Formulating Model Parameter: We formulate the updating equation of model parameter $\mathbf{w}$ by moving the current $\mathbf{w}^t$ along the descent direction of the loss in Eq. (3),

$$\underset{D^{\text{train}}\cup D^{\text{hsl}}}{\hat{\mathbf{w}}^{(t)}}(\mathbf{\Theta}) = \mathbf{w}^{(t)} - \alpha\frac{1}{N}\sum_{n=0}^{N}\mathcal{V}\left(\cdot; \mathbf{\Theta}\right)\nabla_{\mathbf{w}}\mathcal{L}^n(\mathbf{w})\Big|_{\mathbf{w}^{(t)}} \tag{6}$$

where α is the learning rate for $\mathbf{w}$ and $\mathcal{L}^n$ denotes the loss function for the task number n. To avoid cluttered notation, the summation of task samples is omitted.

Updating Parameters of MENTOR: After updating model parameter $\hat{\mathbf{w}}^{(t)}(\mathbf{\Theta})$, parameter $\mathbf{\Theta}$ of MENTOR can be updated guided by Eq. (4), i.e., moving the current parameter $\mathbf{\Theta}^{(t)}$ along the objective gradient of Eq. (4) calculated on the meta-data,

$$\underset{D^{\text{meta}}}{\mathbf{\Theta}^{(t+1)}} = \mathbf{\Theta}^{(t)} - \beta\mathcal{V}(\cdot; \mathbf{\Theta})\mathcal{L}^0\left(\hat{\mathbf{w}}^{(t)}(\mathbf{\Theta})\right)\Big|_{\mathbf{\Theta}^{(t)}} \tag{7}$$

where β is the learning rate for $\mathbf{\Theta}$. This update allows the soft selection of useful HSL proxy tasks and balances them with the main motion prediction task to improve the performance of the main task. Without balancing tasks with the weighting function $\mathcal{V}(\cdot; \mathbf{\Theta})$, proxy tasks can dominate training and degrade the performance of the primary task.

Updating Model Parameters: Model parameters $\mathbf{w}$ for tasks can be updated with optimized $\mathbf{\Theta}^{t+1}$ in Eq. (7) as,

$$\underset{D^{\text{train}}\cup D^{\text{hsl}}}{\hat{\mathbf{w}}^{(t+1)}}(\mathbf{\Theta}) = \mathbf{w}^{(t)} - \alpha\frac{1}{N}\sum_{n=1}^{N}\mathcal{V}\left(\cdot; \mathbf{\Theta}^{(t+1)}\right)\nabla_{\mathbf{w}}\mathcal{L}^n(\mathbf{w})\Big|_{\mathbf{w}^{(t)}}. \tag{8}$$

Lastly, to circumvent the problem of meta-overfitting, meaning that parameters $\mathbf{\Theta}$ overfit to the small meta-dataset, we follow the meta-validation solution as in [2, 59]. In particular, to make $\mathbf{\Theta}$ generalizable across meta-training sets, we optimize it using K different meta-datasets with k-fold cross-validation. The gradients of $\mathbf{\Theta}$ from different meta-datasets are then averaged to update $\mathbf{\Theta}^{(t)}$. The MENTOR learning procedure is summarized in Algorithm 1.

4. Experiments

Our goal is to investigate ways for improving map representational learning power to predict diverse and admissible trajectories. For this purpose, we seek to answer the following questions: **Q1:** Are predicted trajectories diverse and admissible? **Q2:** Is MEta-road NeTwORk (MENTOR) effective regardless of the type of HSL task? **Q3:** How much

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

	Turns					Stationary				
Method	minADE(↓)	minFDE(↓)	DAO(↑)	RF(↑)	MR(↓)	minADE(↓)	minFDE(↓)	DAO(↑)	RF(↑)	MR(↓)
LaneGCN [28]	1.08	2.18	67.63	2.89	35.76	1.23	2.52	61.69	2.69	36.80
MMTransformer[21]	1.05	2.07	69.76	3.35	32.00	**1.13**	2.28	64.85	3.21	35.69
FTGN [3]	1.06	2.04	70.72	3.25	34.06	1.16	2.22	63.95	3.06	**32.49**
HiVT [58]	1.06	2.23	66.84	2.79	35.45	1.25	2.74	58.43	2.51	39.61
SSL-Lane [4]	1.07	2.11	69.65	3.21	33.34	1.16	2.34	62.45	3.22	34.54
MENTOR (Ours)	**1.03**	**2.00**	**79.21**	**3.76**	**31.30**	**1.13**	**2.19**	**69.23**	**3.87**	32.76

does our proposed HSL task contribute to diversity and admissibility?

4.1. Experimental Setup

Dataset: We evaluated our method on the Argoverse benchmark dataset [7]. Here the task is to predict 3 seconds of future trajectories given 2 seconds of past observations. This dataset consists of more than $300K$ real-world driving sequences, which are split into the train (205K), validation (39K), and test (78K) sets without geographical overlap along with corresponding HD maps.

Metrics: We selected a set of metrics to measure the precision, diversity, and admissibility of predictions. For precision, we use minADE (Average Displacement Error), minFDE (Final Displacement Error), and MR (Miss Rate). For diversity, we use the Ratio of avgFDE to minFDE (RF) [34], which measures the spread of predictions in Euclidean distance. We use Drivable Area Occupancy (DAO) [34] to determine the proportion of pixels that predicted trajectories occupy in the drivable area. RF and DAO work in a complementary way to measure diversity along with admissibility.

Implementation Details: To normalize the data, we translate and rotate the coordinate system of each sequence so that the origin is at the agent's current location at $t = 0$. We use the orientation from the agent's location at $t = -1$ to its location at $t = 0$ as the positive x axis. We use all actors and lanes whose distance from the agent is smaller than 100 meters as the input. As a model architecture, we use three layers of HGT with the hidden dimension of size 128. For the HSL tasks, we define five TAROTs with variable lengths of 4 to 7. The intuition behind the number of TAROTs is to model turning right/left, lane changing (right-to-left and left-to-right), and going straight. We formulate Mentor as an MLP with two hidden layers with 100 nodes each, with ReLU activations for hidden nodes and sigmoid output to ensure it is in range $[0, 1]$. For the loss functions, we follow the widely used regression-MLP plus classification-MLP combination [28, 24, 33] for trajectory prediction and cross-entropy for the HSL task. For the training, we use the AdamW [31] optimizer with an initial learning rate of 5e-4, weight decay 1e-4, and batch size of 64.

4.2. Analysis on Challenging Scenarios

Existing prediction datasets are highly imbalanced in terms of scenario diversity, with the majority of traffic actors simply driving straight [35]. Calculating metrics by aggregating over scenarios is therefore influenced by the most common maneuvers, where prediction models often do well. In practice, scenarios, such as turning at intersections could be of greater interest to practitioners and accurately predicting these scenarios are even more crucial for the safe operation of autonomous vehicles. Hence, to verify the effectiveness of our model, we extracted challenging scenarios including left/right/blind turns and stationary scenarios where the vehicle's observed trajectory is very short, thus the model has minimal motion information from the vehicle. By blind turns, we refer to the cases where the observed trajectory is straight and the turn occurs during the unobserved portion of the trajectory.

We compare our method against several vectorized-based SOTA models with public implementations, including two graph-based models, LaneGCN [28] and FTGN [3], two transformer-based models, MMTransformer [21] and HiVT [58], and SSL-Lane [4], which extends the lane graph by adding four different self-supervised tasks based on agent/map information. As shown in Table 1, our method significantly improves upon the state-of-the-art models on most metrics in both scenarios. Particularly, our method, MENTOR achieves notable improvements in diversity and admissibility metrics, namely DAO and RF by up to 13% and 20% respectively in the challenging scenarios.

Qualitative analysis. Fig. 2 depicts a number of qualitative examples to better highlight our method's performance. The first two rows represent turn-right scenarios, and the third and fourth rows show left-turn and stationary scenarios. As can be seen in the last column, our method consistently generates diverse predictions that are also admissible by correctly aligning with the road topology. The behavior of competing methods, however, varies often resulting in the generation of infeasible trajectories. In the supplementary materials, we discuss the algorithm used to find challenging scenarios and present additional per-scenario metrics to further elaborate on the correlation of metrics.

4.3. Ablation Studies

4.3.1 MENTOR Acting as a Mentor

To assess the ability of MENTOR, we analyze the weighting function $\mathcal{V}(\Psi_i^n; \mathbf{\Theta})$ for i^{th} sample of task n. Particularly, we present the learned weight function at the last epoch of

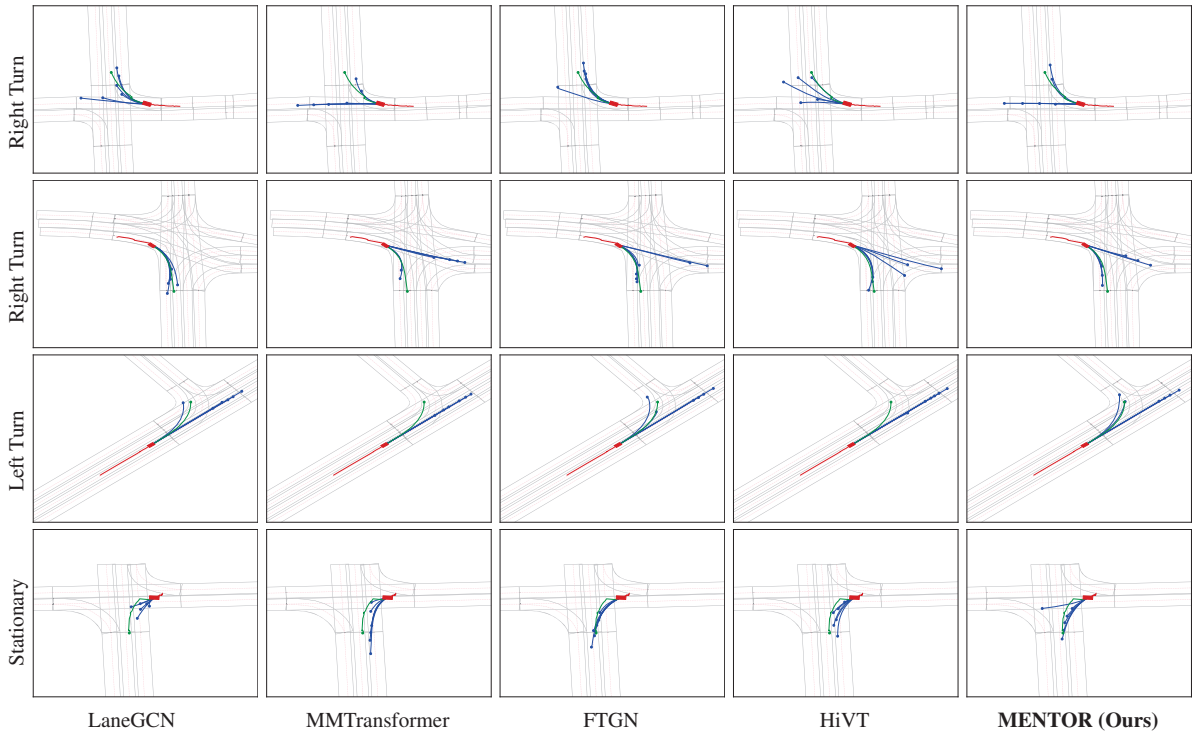

Figure 2: Qualitative examples on Argoverse showing diversity and admissibility of the predicted trajectories by MENTOR. Red, green, and blue trajectories represent observation, ground truth, and 6 predicted trajectories, respectively.

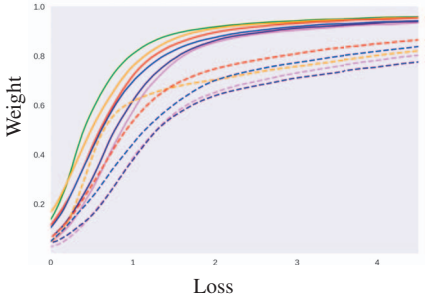

Figure 3: The weighting function in MENTOR. The green line is the primary task, and solid and dashed color lines represent positive and negative HSL samples, respectively. Each color corresponds to a specific HSL task.

the training. As shown in Fig. 3, MENTOR has two main benefits. Firstly, even when the model's loss is low, as the model is learning less challenging scenarios, trajectory prediction task (green line) samples have a higher weight than HSL tasks. Hence, MENTOR is more focused on learning the primary trajectory prediction task as there is no competition between HSL tasks and the primary task. Secondly, in all tasks, hard samples with higher losses are given more weight than easy samples with lower losses, indicating that MENTOR is mining hard samples. Here, MENTOR acts as a mentor by learning TAROTs and transferring that knowledge to the primary task.

Effectiveness of MENTOR: To analyze the effectiveness of MENTOR, we borrow a lane-graph-based baseline and four self-supervised learning (SSL) tasks introduced in [4] and use them in MENTOR to improve the baseline. The SSL tasks are as follows: A) Lane masking, which recovers feature information from the perturbed lane graphs; B) Predicting distance to the intersection in terms of the shortest path length from all lane nodes to the intersection nodes; C) Maneuver classification, which predicts the form of a maneuver that the agent of interest intends to execute; And D) success/failure classification, which trains an agent specialized in achieving end-points.

We report the results in Tab. 2 where the SSL tasks are termed A-D and MENTOR$_{\sum \text{A, B, C, D}}$ denotes the proposed model. According to the table, among the proposed tasks, maneuver classification, task C, has the best DAO (admissibility), indicating that predicting the form of maneuver can reveal more information about the road structure. Additionally, predicting the distance to the intersection, task B, gives a better estimate of the agent to the intersection, and the possibility of branching roads, hence improving RF (diversity) metric. In summary, individual tasks are shown to be effective by providing auxiliary information that helps improve different aspects of prediction. Once combined together, SSL tasks through the MENTOR framework improve distance-based metrics, minADE and minFDE as well as admissibility and diversity metrics, DAO and RF. These results validate the effectiveness of MENTOR as a learning framework to enhance prediction models

Table 2: Results of the ablation studies. SSL_x indicates self-supervised tasks as in [4] and MENTOR$_{\sum \text{A,B,C,D}}$ indicates the original model augmented with MENTOR.

Model	minADE(↓)	minFDE(↓)	DAO(↑)	RF(↑)
Baseline	0.73	1.12	65.87	3.00
SSL$_A$	0.70	1.02	68.21	2.98
SSL$_B$	0.71	1.04	68.65	3.87
SSL$_C$	0.72	1.05	69.97	3.14
SSL$_D$	0.70	1.01	68.02	3.01
MENTOR$_{\sum \text{A, B, C, D}}$	**0.69**	**0.98**	**77.91**	**4.18**

Table 3: Results of the ablation studies. TAROT^x indicates the length of the TAROT through the HSL task and MENTOR represents our full model, which includes several HSL tasks.

	Turns			Validation		
Model	minFDE(↓)	DAO(↑)	RF(↑)	minFDE(↓)	DAO(↑)	RF(↑)
Baseline	2.372	68.45	2.98	1.132	74.02	4.21
TAROT^4	2.267	71.01	3.19	1.128	75.52	4.43
TAROT^5	2.208	73.67	3.23	1.107	77.11	4.48
TAROT^6	2.153	76.93	3.71	1.112	78.89	4.50
TAROT^7	2.082	76.17	3.56	1.121	78.00	4.51
TAROT^8	2.109	73.65	3.41	1.120	77.21	4.38
MENTOR	**2.003**	**79.21**	**3.76**	**0.974**	**85.31**	**4.65**

by providing additional information as SSL tasks.

Impact of TAROT Prediction as an HSL Task: for the purpose of analyzing the impact of the TAROT prediction, we conduct several experiments on the entire validation set and challenging turn scenarios by designing different models named TAROT^x where x refers to the length of TAROT through the HSL task. For example, TAROT^4 could be any of these patterns [S-S-S-R, S-L-S-S, S-L-P-P...] (S: Successor, L: Left, R: Right, P: Predecessor as the basic relation of adjacent lanes). We average the metrics over all of the members in the list. A baseline model does not have any HSL tasks, and therefore does not have MENTOR. The last model is our final model which is loaded by multiple HSL tasks through MENTOR. As shown in Tab. 3, augmenting the model with the HSL task improves the baseline model performance. Moreover, from the experiments, we found that by predicting a longer TAROT, corresponding to a longer pattern in the road configuration with a higher abstraction, we can improve the model in turns scenarios. However, the length of TAROTs should be set carefully, as if set too long, the model would have a tendency to predict a turn maneuver even when several alternatives, such as driving straight, are available. To take advantage of this phenomenon, we define multiple TAROTs with variable lengths representing both short and long patterns. Thus, once used through MENTOR we enable the model to identify the best combination of HSL tasks, as TAROT prediction, by weighting them properly. In this way, our full model outperforms other models significantly. Based on the obtained results, we can conclude that 1) TAROTs are capable of modeling patterns in the map structure, thus formulating multiple TAROTs and subsequently multiple HST tasks, can capture diverse lane connection patterns in the map topology. 2) MENTOR is good at adopting HSL tasks to transfer their knowledge to the primary task, and consequently resulting in significant improvement in terms of diverse (RF) and admissible (DAO) trajectory prediction.

Table 4: Comparisons with SOTA on Argoverse test set.

Method	minADE(↓)	minFDE(↓)	MR(↓)
TNT [56]	0.94	1.54	13.30
DenseTNT [18]	0.88	1.28	12.58
TPCN [49]	0.87	1.36	15.80
mmTransformer [30]	0.87	1.34	15.40
LaneRCNN [54]	0.90	1.45	**12.32**
LTP [45]	0.83	1.29	14.72
AutoBot [17]	0.89	1.41	16.00
SceneTransformer [33]	0.80	1.23	12.55
HiVT [58]	**0.77**	**1.16**	12.67
MENTOR (Ours)	0.79	1.21	13.01

4.4. Comparison to SOTA

We compare our method to SOTA models on the Argoverse benchmark. Even though the focus of our method was on improving diversity and admissibility of generated trajectories in challenging scenarios, as shown in Tab. 4, our model's performance is comparable to best models, on the distance-based metrics on the benchmark. Particularly, compared to HiVT [58], our model achieves the second rank on minADE and minFDE, while in challenging scenarios it outperforms HiVT on the diversity and admissibility metrics as well as the distance-based metrics with a large margin, as is evident in Tab. 1. We further verify the effectiveness of our approach on a smaller dataset, nuScenes [5], and report the results in the supplementary material.

5. Conclusion

In this paper, we presented a novel trajectory prediction model based on the concept of meta-learning. We formulated various lane connection relations on the map topology as meTA ROad paTh (TAROT), and used a model, MEta-road NeTwORk (MENTOR), to supply TAROT tips in the form of Heterogeneous Structural Learning (HSL) tasks to improve trajectory prediction. Besides, focusing on enhancing the map representational learning power, MENTOR can mentor any proxy task to further improve the primary motion forecasting task. Via extensive experimental evaluations, we showed that HSL tasks are capable of modeling high-level semantics stemming from long-range heterogeneous connections between lanes. Additionally, we showed that our proposed MENTOR model acts as a mentor to gain the maximum benefit from the existing structural information of the map. The results also highlighted that, besides significantly improving diversity and admissibility of predictions, our method can obtain improved or comparable results on common distance-based metrics compared to the past arts with a larger number of parameters.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016. 5

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *ICLR*, 2019. 5

[3] Görkay Aydemir, Adil Kaan Akan, and Fatma Güney. Trajectory forecasting on temporal graphs. *arXiv:2207.00255*, 2022. 6

[4] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. SSL-Lanes: Self-supervised learning for motion forecasting in autonomous driving. In *CoRL*, 2022. 1, 2, 4, 6, 7, 8

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 8

[6] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2019. 1, 2

[7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 6

[8] Yaomin Chang, Chuan Chen, Weibo Hu, Zibin Zheng, Xiaocong Zhou, and Shouzhi Chen. MEGNN: Meta-path extracted graph neural network for heterogeneous graph representation learning. *Knowledge-Based Systems*, 235:107611, 2022. 4

[9] Henggang Cui, Hoda Shajari, Sai Yalamanchi, and Nemanja Djuric. Ellipse loss for scene-compliant motion prediction. In *ICRA*, 2021. 2

[10] Mostafa Dehghani, Aliaksei Severyn, Sascha Rothe, and Jaap Kamps. Learning to learn from weak supervision by full supervision. *arXiv:1711.11383*, 2017. 5

[11] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *CoRL*, 2022. 1

[12] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 4

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2, 5

[14] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, 2018. 5

[15] 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 *CVPR*, 2020. 1, 2, 4

[16] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. GOHOME: Graph-oriented heatmap output for future motion estimation. In *ICRA*, 2022. 1, 2

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

[18] Junru Gu, Chen Sun, and Hang Zhao. DenseTNT: End-to-end trajectory prediction from dense goal sets. In *ICCV*, 2021. 1, 2, 8

[19] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *The World Wide Web Conference*, 2020. 3

[20] Xin Huang, Stephen G McGill, Jonathan A DeCastro, Luke Fletcher, John J Leonard, Brian C Williams, and Guy Rosman. DiversityGAN: Diversity-aware vehicle motion prediction via latent semantic sampling. *RAL*, 5(4), 2020. 2

[21] Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *ICRA*, 2022. 2, 6

[22] Dasol Hwang, Jinyoung Park, Sunyoung Kwon, KyungMin Kim, Jung-Woo Ha, and Hyunwoo Kim. SELAR: Self-supervised auxiliary learning with meta-paths for heterogeneous graphs. In *NuerIPS*, 2020. 4

[23] Yunhun Jang, Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Learning what and where to transfer. In *ICML*, 2019. 2

[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:2205.09753*, 2022. 3, 6

[25] Parth Kothari and Alexandre Alahi. Safety-compliant generative adversarial networks for human trajectory forecasting. *Transactions on Intelligent Transportation Systems*, 2023. 2

[26] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. DESIRE: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017. 2

[27] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to learn quickly for few-shot learning. *arXiv:1707.09835*, 2017. 2

[28] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 1, 2, 3, 6

[29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[30] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 2, 8

[31] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in ADAM. *arXiv:1711.05101*, 2017. 6

[32] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *ICRA*, 2020. 2

[33] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al.

Scene Transformer: A unified architecture for predicting future trajectories of multiple agents. In *ICLR*, 2022. 6, 8

[34] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *ECCV*, 2020. 2, 6

[35] Derek J Phillips, Tim A Wheeler, and Mykel J Kochenderfer. Generalizable intention prediction of human drivers at intersections. In *IV*, 2017. 6

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3

[37] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. In *CVPR*, 2019. 2

[38] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 2

[39] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-Weight-Net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019. 2, 4, 5

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2

[41] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2

[42] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. In *NeurIPS*, 2019. 2

[43] 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 *ICRA*, 2022. 1, 2

[44] Eason Wang, Henggang Cui, Sai Yalamanchi, Mohana Moorthy, and Nemanja Djuric. Improving movement predictions of traffic actors in bird's-eye view models using gans and differentiable trajectory rasterization. In *International Conference on Knowledge Discovery and Data Mining*, 2020. 2

[45] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. LTP: Lane-based trajectory prediction for autonomous driving. In *CVPR*, 2022. 8

[46] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, 2019. 4

[47] Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Lai Jian-Huang, and Tie-Yan Liu. Learning to teach with dynamic loss functions. In *NeurIPS*, 2018. 5

[48] Sai Yalamanchi, Tzu-Kuo Huang, Galen Clark Haynes, and Nemanja Djuric. Long-term prediction of vehicle behavior using short-term uncertainty-aware trajectories and high-definition maps. In *ITSC*, 2020. 2

[49] Maosheng Ye, Tongyi Cao, and Qifeng Chen. TPCN: Temporal point cloud networks for motion forecasting. In *CVPR*, 2021. 2, 8

[50] Maosheng Ye, Jiamiao Xu, Xunnong Xu, Tongyi Cao, and Qifeng Chen. DCMS: Motion forecasting with dual consistency and multi-pseudo-target supervision. *arXiv:2204.05859*, 2022. 2

[51] Jianxiang Yu and Xiang Li. Heterogeneous graph contrastive learning with meta-path contexts and weighted negative samples. *arXiv:2212.13847*, 2022. 4

[52] Ye Yuan and Kris Kitani. Diverse trajectory forecasting with determinantal point processes. *arXiv:1907.04967*, 2019. 2

[53] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, 2021. 2

[54] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. LaneRCNN: Distributed representations for graph-centric motion forecasting. In *IROS*, 2021. 1, 2, 3, 8

[55] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019. 2

[56] 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 *CoRL*, 2021. 1, 8

[57] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, 2019. 2

[58] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. HiVT: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, 2022. 2, 6, 8

[59] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *ICML*, 2019. 5