STraj: Self-training for Bridging the Cross-Geography Gap in Trajectory Prediction

Zhanwei Zhang[1], Minghao Chen[2], Zhihong Gu[3], Xinkui Zhao[4], Zheng Yang[5]*, Binbin Lin[4], Deng Cai[1], Wenxiao Wang[4]

[1]State Key Lab of CAD&CG, Zhejiang University
[2]Hangzhou Dianzi University
[3]Beijing Automobile Works
[4]School of Software Technology, Zhejiang University
[5]FABU Inc.
{zhanweizhang, zhaoxinkui, binbinlin, wenxiaowang}@zju.edu.cn,
{minghaochen01, dengcai}@gmail.com, guzhihong@baw.com.cn, yangzheng@fabu.ai

Abstract

Accurate trajectory prediction has prominent significance in autonomous driving scenarios. Most existing methods predict the trajectory of an agent by learning its interaction with other agents and the map within the scenario. However, the heterogeneous distribution of these elements across different geographical scenarios is always ignored. Thus, trajectory predictors might struggle to generalize well when deployed in different geographical scenarios. To bridge the cross-geography gap, in this paper, we propose a plug-and-play self-training pipeline, termed STraj, for cross-geography trajectory prediction. STraj comprises three progressive steps: pseudo label (*i.e.*, time-series trajectory) generation, update, and utilization. First, to generate pseudo labels that generalize to the cross-geography scenarios, STraj pre-trains the predictor through the complementary agent and map augmentations. Second, to facilitate the stable training of the predictor, we design a specific pseudo label update strategy. This strategy selects high-consistency pseudo trajectories from the current and historical epochs to supervise the target domain samples. Third, with generated pseudo trajectories, we introduce trajectory-induced contrastive learning to mitigate the representation bias of cross-geography agents. Extensive experiment results on various cross-geography trajectory prediction benchmarks demonstrate the effectiveness of STraj.

Code — https://github.com/Zhanwei-Z/STraj

Introduction

Trajectory prediction is a critical task in self-driving scenarios, which is significant for autonomous vehicles to take safe and reliable actions (Liu et al. 2024; Li et al. 2024). This task involves predicting the future trajectories of agents (*i.e.*, traffic participants) in motion, considering their interactions within the scenarios. Specifically, lane segments incorporated in road maps, together with other agent states, are vital interactive clues (Liang et al. 2020; Liao et al. 2024; Gu et al. 2024). However, these components are distributed heterogeneously across different geographical scenarios. As a result, the generalization ability of most trajectory predictors might be limited when transferred to diverse geographical scenarios. For instance, as shown in Fig. 1, the distribution of agent and lane segment numbers in Pittsburgh exhibit notable dissimilarities compared to that in Miami (Chang et al. 2019). Consequently, as shown in Table 1, a well-known predictor LaneGCN (Liang et al. 2020) and a state-of-the-art (SOTA) predictor HPNet (Tang et al. 2024) trained in Pittsburgh both experience a substantial performance decline (of 44.0-61.3% and 53.0-68.6% in terms of the minADE and minFDE metrics) when directly applied to Miami scenarios. While the potential solution lies in the collection of additional data, the feasibility is unfortunately hindered due to diverse real-world scenarios and the substantial costs entailed in trajectory annotation. Thus, unsupervised domain adaptation (UDA) for trajectory prediction becomes a challenging but highly demanded topic, which aims to explore an effective way to reduce the annotated trajectories in the target domain while maintaining satisfactory generalization. In this field, existing methods (Xu et al. 2022; Liu et al. 2022; Bahari et al. 2022; Bhattacharyya, Huang, and Czarnecki 2023; Kothari et al. 2023; Qian et al. 2024; Park, Jeong, and Yoon 2024) commonly concentrate on addressing general generalization issues, while (Ye, Zhou, and Wang 2023) specifically addresses domain gaps arising from the road infrastructure. In contrast, we delve into the cross-geography domain gap and focus on alleviating this gap, since most trajectory predictors

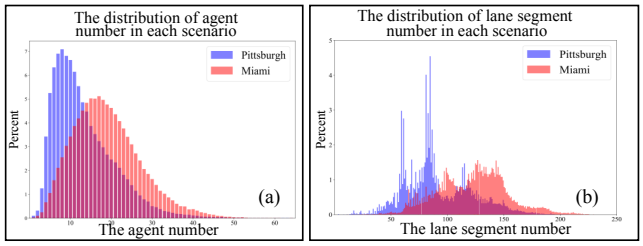

Figure 1: The cross-geography gap is a common issue in the trajectory prediction task. (a) and (b) present significant cross-geography distribution differences in terms of agent and lane segment numbers, respectively.

*Corresponding author

are sensitive to the varying agent and lane segment distribution across different geographical scenarios.

Recently, self-training methods employ pseudo labels to fill domain gaps, achieving noticeable developments across various image tasks (Yue, Sun, and Zhang 2023; Du et al. 2024). These methods generally leverage pseudo labels to provide supervision for unlabeled samples and iteratively update pseudo labels during the training process. Despite their effectiveness in image-related tasks, applying the SOTA self-training methods, originally designed for image tasks, directly to trajectory prediction might lead to a substantial drop in performance, such as ICON (Yue, Sun, and Zhang 2023) in Table 1. This drop can be attributed to the inherent data differences, as trajectory data consists of time-series agent coordinates, with varying numbers of agents and lane segments in different scenarios. These findings enlighten us to explore self-training based UDA for cross-geography trajectory prediction.

In this paper, we propose a framework named **S**elf-training for Bridging the Cross-Geography Gap in **Traj**ectory Prediction (STraj). Specifically, STraj consists of three progressive steps: pseudo trajectory generation, update, and utilization. These steps are particularly designed to resolve three challenges commonly encountered in cross-geography scenarios. ***First***, the source domain data is used to pre-train the predictor. It generally encompasses a limited range of agents and lane segments. Their number distribution varies from that in the target domain. This limits the generalization on the cross-geography target domain data. To expand the diversity of training data for adaptive trajectory prediction, (Bhattacharyya, Huang, and Czarnecki 2023) leverages agent or map augmentations, which mask a portion of agent trajectories or lane segments. However, this method faces the issues of potentially masking excessive or insufficient agent and map information simultaneously. The former issue can cause significant information loss and complicate the reconstruction of original features, while the latter can make the reconstruction process too easy, resulting in overfitting the existing data. To avoid these problems, during the pre-training process, we propose a complementary agent and map augmentations. Subsequently, we leverage the mean squared error (MSE) loss to reconstruct all masked agent, map, and fusion features. Through pre-training, we aim to enhance the quality of preliminary pseudo labels for the subsequent training process. ***Second***, ensuring consistent pseudo labels throughout the training process is crucial for maintaining the stability of the predictor. However, given an agent, its time-series pseudo trajectories across different epochs may not completely overlap with each other. To facilitate the stable training of the predictor, we introduce a specific update strategy in the field of trajectory prediction. This strategy involves iteratively updating pseudo labels by selecting high-consistency pseudo trajectories from the current and historical epochs. Subsequently, the updated pseudo labels are utilized to supervise target domain samples, with weighted losses. Concretely, the loss weight is calculated using an exponential function based on the consistency level. This function allows the predictor to focus more on reliable and consistent labels, thereby improving the training stability. ***Third***, ideally, agents with different future trajectories (*e.g.*, go-straight, turn-left and turn-right) should induce distinct representations, whereas agents with similar trajectories should generate similar representations. Indeed, the representation of an agent integrates the information of other agents and lane segments. However, as shown in Fig. 1, cross-geography scenarios exhibit significant differences in the distribution of other agents and lane segments, thus leading to the agent representation bias between different geographical scenarios. To alleviate the agent representation bias, with the help of generated pseudo trajectories, we propose trajectory-induced contrastive learning. This is achieved by implementing inter-domain and intra-domain contrastive losses. More concretely, we concentrate on pulling agent representations of similar trajectories closer while pushing apart those that are dissimilar, regardless of whether they belong to the same domain or different domains. Notably, as a plug-and-play module, STraj can be easily integrated into existing trajectory predictors without introducing additional computation burdens in the inference process.

Our contributions can be summarized as follows:

- We investigate the cross-geography domain gap and propose a self-training based UDA pipeline for cross-geography trajectory prediction.
- To generate and update pseudo labels, we introduce the complementary agent and map augmentations through weighted random masking during pre-training, and formulate a particular update strategy for stable training.
- With generated pseudo labels, we devise trajectory-induced contrastive learning to attenuate the cross-geography bias of agent representations.
- Our STraj is evaluated across various cross-geography trajectory prediction tasks. Extensive experiment results validate its effectiveness and generalization ability.

Related Work

Trajectory Prediction. Trajectory Prediction is a significant task for autonomous vehicles (Shen et al. 2023; Liu et al. 2024; Li et al. 2024; Zhang et al. 2024b). With the prominent success of deep neural networks, existing methods generally utilize data-driven manners to derive features from map and agent perspectives, and predict future trajectories for interested agents (Liang et al. 2020; Liao et al. 2024; Gu et al. 2024; Xu and Fu 2024). These approaches introduce various architectures, which mainly leverage convolutional neural networks to handle rasterized images (Park et al. 2020; Casas, Sadat, and Urtasun 2021; Chai et al. 2019), graph neural networks to tackle elaborate vectorized representations (Ha and Jeong 2023), point-cloud representations to encode relevant trajectory information (Ye, Cao, and Chen 2021), and transformers (Vaswani et al. 2017) to generate attention weights of lane segments and agents (Liu et al. 2024). Recently, several self-supervised learning methods (Bhattacharyya, Huang, and Czarnecki 2023; Cheng, Mei, and Liu 2023; Chen et al. 2023; Lan et al. 2023) commonly employ data augmentation to boost prediction performance. Another branch focuses on leveraging contrastive learning (Wang et al. 2023; Zhao et al. 2024) to achieve improved prediction outcomes. However, facing domain gaps in trajectory prediction, few predictors

could generalize well when directly transferred to the unlabeled target domain (Kothari et al. 2023; Qian et al. 2024).

Unsupervised Domain Adaptation endeavors to develop the generalization of models when transferred from labeled source domains to unlabeled target domains (Long et al. 2015; Sun et al. 2024). Some works (Chhabra, Venkateswara, and Li 2023; Wu et al. 2024) extend GAN-based (Goodfellow et al. 2020) adversarial learning to various computer vision tasks by aligning implicit representation distributions in the common cross-domain space. (Benigmim et al. 2023; Tsai et al. 2024; Peng et al. 2024) resort to diffusion models for achieving cross-domain feature alignment. In recent years, pseudo-label based self-training UDA has been extensively investigated for a variety of deep learning tasks, such as classification, recognition, detection and segmentation (Yue, Sun, and Zhang 2023; Du et al. 2024; Zhang et al. 2024a; Feng et al. 2024). While these methods excel in generating pseudo labels for specific tasks, they might significantly underperform when directly applied to trajectory prediction. This limitation stems from overlooking the inherent model and data differences between image tasks and trajectory prediction.

Model Generalization for Trajectory Prediction. Recently, a few works have endeavored to improve model generalization for trajectory prediction. (Liu et al. 2022) learns an adaptive human trajectory prediction mechanism from a causal representation perspective. (Lu et al. 2022) analyses the generalization bottlenecks of graph-based trajectory predictors. (Hu et al. 2022) proposes a structural causal model to learn adaptive representations to alleviate domain gaps. Furthermore, several studies (Bhattacharyya, Huang, and Czarnecki 2023) (Kothari et al. 2023) employ semi-supervised domain adaptation to develop model generalization. Additionally, (Liang, Jiang, and Hauptmann 2020; Bahari et al. 2022) improve the generalization performance of existing models by generating realistic scenes. (Xu et al. 2022) resorts to a transferable graph neural network for aligning the source and target trajectories. (Ye, Zhou, and Wang 2023) specifically addresses domain gaps caused by road geometry and topology. (Qian et al. 2024) introduces a multi-source domain generalization pipeline designed for multi-agent trajectory prediction. (Park, Jeong, and Yoon 2024) employs continuous and stochastic representations to mitigate cross-domain discrepancies. Despite significant progress, few approaches have specifically explored the cross-geography domain gap. In contrast, we introduce a self-training framework, specially designed to mitigate this gap in trajectory prediction.

Methodology

Problem Formulation

In the traditional setup of trajectory prediction, we are given the past trajectory coordinates $P_a = \{(x_k^a, y_k^a)\}_{k=-L_p+1}^{0}$ of an interested agent a, where L_p represents the length of past steps. The objective is to predict its future trajectory coordinates $F_a = \{(x_k^a, y_k^a)\}_{k=1}^{L_f}$, where L_f is the length of future steps. Naturally, in the given scenario, a interacts with other agents and lane segments on the map, and we respectively denote them as P_o and M for the sake of brevity. Our work builds upon an existing trajectory predictor $\mathcal{P}$, whose pipeline can generally be illustrated by

$$\begin{aligned} \mathcal{A} &= \phi_a\left(P_a, P_o\right), \quad \mathcal{M} = \phi_m\left(P_a, M\right), \\ \mathcal{F} &= \phi_f\left(\mathcal{A}, \mathcal{M}\right), \quad \hat{F}_a, s(\hat{F}_a) = \varphi\left(\mathcal{F}\right), \end{aligned} \tag{1}$$

where ϕ_a, ϕ_m and ϕ_f are feature encoders of $\mathcal{P}$, φ is a prediction decoder. $\mathcal{A}$, $\mathcal{M}$, and $\mathcal{F}$ denote agent features, map features and fusion features, respectively. $\hat{F}_a$ denotes the predicted most likely trajectory, under the agent-centric coordinate system (Zhao et al. 2021; Varadarajan et al. 2022; Liang et al. 2020), while $s(\hat{F}_a)$ denotes its corresponding confidence score. If agent a belongs to the source domain $\mathbb{D}_s$, $\mathcal{P}$ is trained in a supervised manner by minimizing the prediction error between $\hat{F}_a$ and the ground truth F_a, utilizing the inherent loss $\mathcal{L}_{pred}^s$ of $\mathcal{P}$. Otherwise, if agent a belongs to the target domain $\mathbb{D}_t$, the ground truth F_a is unknown. The UDA for trajectory prediction aims to train $\mathcal{P}$ based on the labeled $\mathbb{D}_s$ and the unlabeled $\mathbb{D}_t$, and maximize $\mathcal{P}$'s generalization on $\mathbb{D}_t$ (Xu et al. 2022). Notably, this paper focuses on the cross-geography UDA. In other words, $\mathbb{D}_s$ and $\mathbb{D}_t$ are derived from distinct geographical scenarios.

Framework Overview

The overview framework of our STraj is depicted in Fig. 2. It consists of three progressive steps: pseudo label generation, update, and utilization. First, to generate the initial pseudo labels that are adaptable to the cross-geography scenarios, STraj pre-trains the predictor $\mathcal{P}$ using complementary agent and map augmentations. Second, to enable the stable training of the predictor, we propose a specific pseudo label update strategy during training. Third, by utilizing the generated pseudo trajectories, we introduce trajectory-induced contrastive learning to alleviate the representation bias of cross-geography agents.

Model Pre-training with Complementary Augmentation

The data from the source domain is used for pre-training $\mathcal{P}$. However, as depicted in Fig. 1, this data typically covers a limited range of scenarios, where the agent and lane segment distribution varies from that in the target domain. Consequently, when $\mathcal{P}$ generates initial pseudo labels from the cross-geography target domain data, these labels exhibit limited generalization. To this end, we propose complementary augmentations from the agent and map perspectives.

Concretely, as for agent a (from $\mathbb{D}_s$ or $\mathbb{D}_t$), we calculate the current distance between a and another agent $j \in P_o$ by

$$w_{aj} = \sqrt{|x_0^a - x_0^j|^2 + |y_0^a - y_0^j|^2}/\rho_a, \tag{2}$$

where ρ_a is a scaling parameter. Then, the masking probability of agent j for agent augmentation is normalized by a softmax, which can be written as

$$p_j = \begin{cases} \frac{\exp\left(w_{aj}\right)}{\sum_{i \in P_o} \exp\left(w_{ai}\right)}, & \text{Weak Augmentation}, \\ \frac{\exp\left(-w_{aj}\right)}{\sum_{i \in P_o} \exp\left(-w_{ai}\right)}, & \text{Strong Augmentation}. \end{cases} \tag{3}$$

Here, with weak augmentation, the farther agents are prone to be masked, while the closer agents are inclined to be

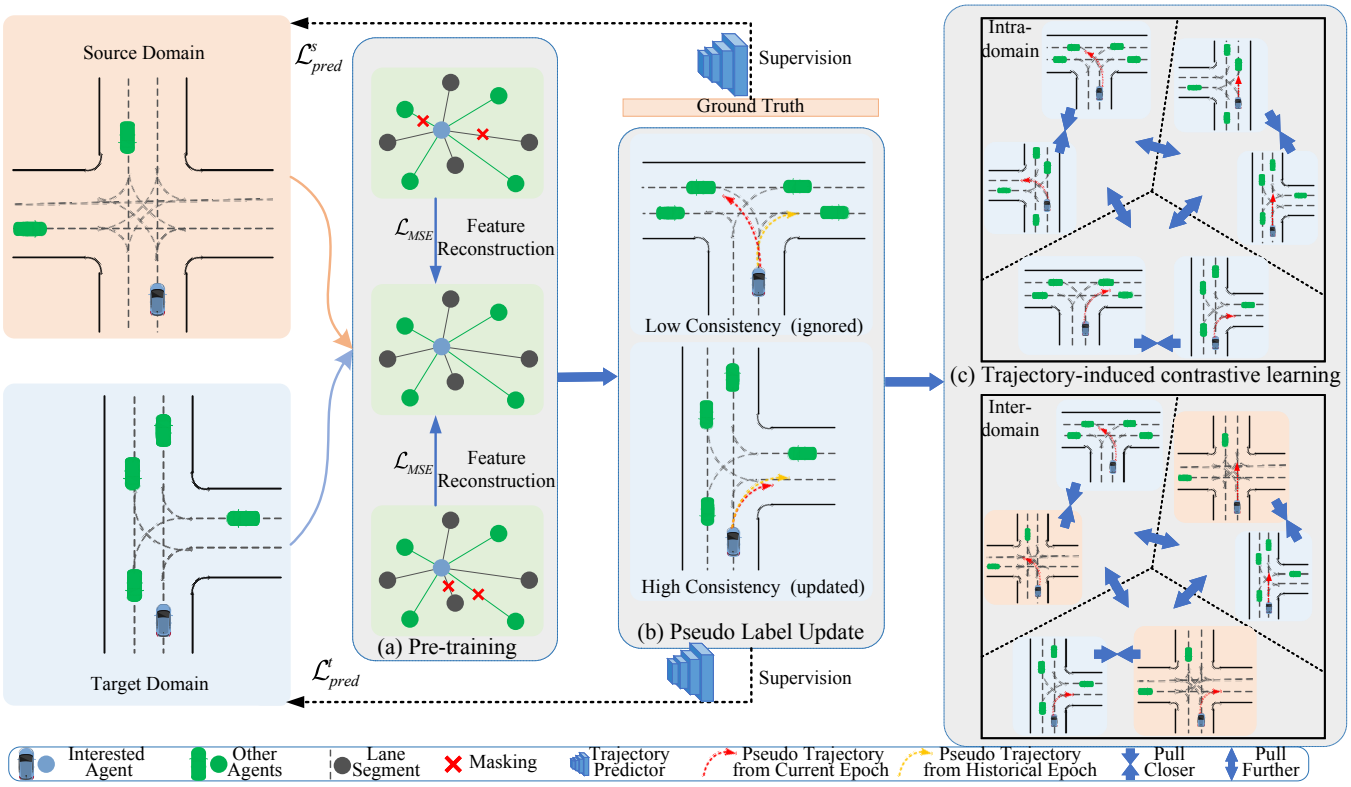

Figure 2: The overall framework of our STraj. It is composed of three progressive steps: pseudo label generation, update, and utilization. (a) To generate the initial pseudo labels that generalize to the cross-geography scenarios, STraj pre-trains the predictor $\mathcal{P}$ using complementary agent and map augmentations through weighted random masking. (b) To enable the stable training of $\mathcal{P}$, we devise a particular pseudo label update strategy during training. (c) By utilizing the generated pseudo trajectories, we propose trajectory-induced contrastive learning to alleviate the representation bias of cross-geography agents.

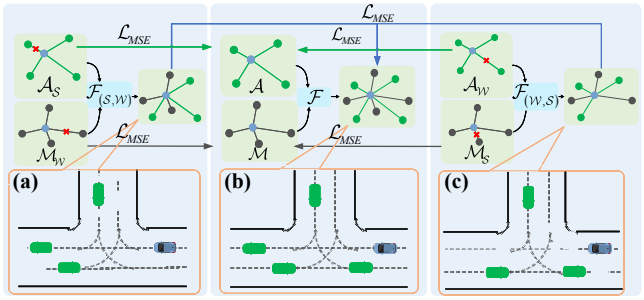

Figure 3: Agent and map augmentations through weighted random masking. (a) With $\mathcal{A}_{\mathcal{S}}$ and $\mathcal{M}_{\mathcal{W}}$, the closer agents and the farther lane segments are prone to be masked. (b) Original features. (c) With $\mathcal{A}_{\mathcal{W}}$ and $\mathcal{M}_{\mathcal{S}}$, the farther agents and the closer lane segments are inclined to be masked.

masked with strong augmentation. With a larger ρ_a, the gaps in masking probability between far agents and close agents will be smaller. As $\rho_a \to +\infty$, the masking probability for each agent becomes the same, and Eq. (3) is equivalent to random masking. Similarly, for a given lane segment $m \in M$, we calculate the average distance between a and m to obtain the masking probability of m, where $\left(x_0^j, y_0^j\right)$ in Eq. (2) is replaced by the average coordinates of m. Then, with the masking probability, we mask r percent agents and lane segments to implement weak and strong augmentations, and the masked features are written as

$$\begin{aligned} \mathcal{A}_{\mathcal{W}} &= \phi_a\left(P_a, \mathcal{W}\left(P_o\right)\right), \quad \mathcal{A}_{\mathcal{S}} = \phi_a\left(P_a, \mathcal{S}\left(P_o\right)\right), \\ \mathcal{M}_{\mathcal{W}} &= \phi_m\left(P_a, \mathcal{W}\left(M\right)\right), \ \mathcal{M}_{\mathcal{S}} = \phi_m\left(P_a, \mathcal{S}\left(M\right)\right), \end{aligned} \tag{4}$$

where $\mathcal{W}$ and $\mathcal{S}$ denote weak and strong augmentations, respectively. Intuitively, the closest agents lane segments contain more useful information for predicting a's future trajectories than the farthest ones. Distinguished from random masking, which might mask excessive or insufficient agent and map information simultaneously (Bhattacharyya, Huang, and Czarnecki 2023), we derive masked fusion features by a trade-off strategy as

$$\begin{aligned} \mathcal{F}_{(\mathcal{W},\mathcal{S})} &= \phi_f\left(\mathcal{A}_{\mathcal{W}}, \mathcal{M}_{\mathcal{S}}\right), \\ \mathcal{F}_{(\mathcal{S},\mathcal{W})} &= \phi_f\left(\mathcal{A}_{\mathcal{S}}, \mathcal{M}_{\mathcal{W}}\right). \end{aligned} \tag{5}$$

As shown in Fig. 3, we then employ the mean squared error (MSE) loss $\mathcal{L}_{MSE}$ to reconstruct all masked agent, map and fusion features from original features in Eq. (1). Notably, if $a \in \mathbb{D}_s$, all masked and original samples are also supervised

by $\mathcal{L}_{pred}$ along with the ground truths in the pre-training state. Albeit simple, our augmentation approach simulates diverse self-driving scenarios to improve the quality of preliminary pseudo labels for the subsequent refinery.

Pseudo Label Update Strategy

After acquiring the pre-trained $\mathcal{P}$, the next step is to refine pseudo labels by updating them during each epoch. Different from image tasks, effectively updating pseudo trajectories presents great challenges. First, cross-geography gaps might induce low-consistency pseudo trajectories across epochs, as shown in the upper part of Fig. 2(b). Second, as for the high-consistency pseudo trajectories, they might not completely overlap each other, as shown in the lower part of Fig. 2(b).

To address these problems, we leverage the trajectory consistency to select useful information. Concretely, given an agent a from $\mathbb{D}_t$, let $\hat{F}^i$ denote its i-th pseudo trajectory derived after training epoch i for brevity, and let $\mathbb{H}^{i-1}$ denote a set incorporating pseudo trajectories from historical epochs. Notably, $\mathbb{H}^0$ is generated based on the pre-trained $\mathcal{P}$. We then measure the consistency $\mathcal{U}$ between $\hat{F}^i$ and each pseudo trajectory in $\mathbb{H}^{i-1}$, by formulating the cosine similarity of their entire L_f time-series coordinates, rather than solely focusing on their endpoints. Obviously, the greater the cosine similarity, the higher the consistency $\mathcal{U}$. Consequently, we derive the most matched pseudo trajectory of $\hat{F}^i$ by

$$\hat{h} = \text{argmax}_{h \in \mathbb{H}^{i-1}} \left\{ \mathcal{U}(\hat{F}^i, h) \right\}. \tag{6}$$

Algorithm 1: Algorithm of the Pseudo Label Update Strategy

Input: The pseudo trajectory $\hat{F}^i$ of agent a during epoch i, the historical pseudo trajectory set $\mathbb{H}^{i-1}$, the confidence score s for each trajectory mentioned in Eq. (1).
Output: An update pseudo trajectory used for supervising agent a in epoch $i+1$.

Let $i \leftarrow 1$.
while $i \leq \text{Epoch}_{max}$ **do**
  Calculate $\hat{h}$ by Eq. (6).
  if $\mathcal{U}(\hat{F}^i, \hat{h}) < T_{\mathcal{U}}$ or $\max\left(s(\hat{F}^i), s(\hat{h})\right) < T_c$ **then**
    $\mathbb{H}^i \leftarrow \mathbb{H}^{i-1} \cup \hat{F}^i$;
    ▷ *No pseudo trajectory is selected for supervising agent a in the next epoch.*
  else
    $\mathbb{H}^i \leftarrow \hat{h} \cup \hat{F}^i$
    if $s(\hat{F}^i) > s(\hat{h})$ **then**
      $b^i \leftarrow \hat{F}^i$.
    else
      $b^i \leftarrow \hat{h}$.
    end if
    ▷ b^i *is the updated pseudo trajectory for supervising $\hat{F}^{i+1}$ of agent a in the next epoch.*
  end if
  $i \leftarrow i + 1$;
end while

As shown in Algorithm 1, we set a consistency threshold $T_{\mathcal{U}}$ and a confidence threshold T_c to ignore low-consistency and low-confidence pseudo trajectories and leverage the update strategy to obtain the updated pseudo trajectory b^i, which is subsequently utilized to supervise the predicted trajectory $\hat{F}^{i+1}$ of agent a during epoch $i+1$. Considering the incomplete overlap issue, we assign the loss weight in the target domain using an exponential function as

$$\mathcal{L}_{pred}^{tar} = \exp\left(\mathcal{U}(\hat{F}^i, \hat{h})/\rho_t\right) \mathcal{L}_{pred}\left(\hat{F}^{i+1}, b^i\right), \tag{7}$$

where ρ_t is a scaling parameter, which controls the overall weight range of target domain samples. Notably, the weight increases with higher consistency values. By assigning the loss weight, we further strengthen the utilization of high-consistency pseudo trajectories.

Trajectory-induced Contrastive Learning

Ideally, agents with different future trajectories (*e.g.*, go-straight, turn-left and turn-right) should induce distinct representations, whereas agents with the similar trajectories should generate similar representations. As shown in Eq. (1), the fusion representation of an agent integrates the representations of other agents and lane segments. However, as shown in Fig. 1, cross-geography scenarios exhibit significant differences in the numbers of other agents and lane segments, leading to the agent representation bias. To tackle this issue, (Hu et al. 2022) considers category-level contrastive learning. Yet, agents within the same categories might present distinct trajectories. Thus, the core idea of our trajectory-induced contrastive learning is to enhance the cross-geography representation compactness in similar trajectories and strengthen the representation separability in divergent trajectories. Specifically, given the predicted trajectory $\hat{F}_a$ of agent a in domain d_1, we select its corresponding positive index p and the negative index set S_n from domain d_2, where $d_1, d_2 \in \{\mathbb{D}_s, \mathbb{D}_t\}$. These indices are determined based on the above trajectory consistency and can be calculated by

$$p = \text{argmax}_{j \in J, \mathcal{U}(\hat{F}_a, \hat{F}_j) \geq T_{\mathcal{U}}} \left(\mathcal{U}(\hat{F}_a, \hat{F}_j)\right), \tag{8}$$

$$S_n = \left\{\mathcal{U}(\hat{F}_a, \hat{F}_j)\right\}_{j \in J, \mathcal{U}(\hat{F}_a, \hat{F}_j) < T_{\mathcal{U}}}, \tag{9}$$

where J denotes the total agent representations within the same category as agent a (*e.g.*, vehicle, pedestrian and cyclist) in domain d_2 at the current batch. Notably, we eliminate the low-confidence agents using the aforementioned confidence threshold T_c. Moreover, if agent $a \in \mathbb{D}_s$, its ground truth trajectory F_a is available. Following the teacher forcing technique (Bengio et al. 2015), we exploit the ground truth F_a to replace $\hat{F}_a$. In this way, we aim to mitigate the error accumulation during this alignment process. The same strategy is applied to agent j as well. Inspired by InfoNCE (Oord, Li, and Vinyals 2018), if both p and S_n exist, the trajectory-induced contrastive loss can be formulated as

$$\mathcal{L}_{(d_1,d_2)} = -\log \frac{\exp\left(\text{sim}(\mathcal{F}_\text{a}, \mathcal{F}_\text{p})/\rho_\text{c}\right)}{\sum_{j \in S_n \cup p} \exp\left(\text{sim}(\mathcal{F}_\text{a}, \mathcal{F}_\text{j})/\rho_\text{c}\right)} \tag{10}$$

where $\mathrm{sim}(x, y) = \frac{x \cdot y}{||x||||y||}$ denotes the cosine similarity, ρ_c denotes the temperature parameter, and $\mathcal{F}$ denotes agent fusion features. Obviously, as ρ_c decreases, the gradient of $\mathcal{L}_{(d_1,d_2)}$ becomes larger. As shown in Fig. 2(c), we implement intra-domain and inter-domain contrastive learning to mitigate cross-geography domain shifts. Specifically, As $d_1 = d_2$, $\mathcal{L}_{d_1,d_2}$ is the intra-domain contrastive loss, denoted as $\mathcal{L}_{(\mathbb{D}_s,\mathbb{D}_s)}$ and $\mathcal{L}_{(\mathbb{D}_t,\mathbb{D}_t)}$; as $d_1 \neq d_2$, $\mathcal{L}_{d_1,d_2}$ is the inter-domain contrastive loss, represented as $\mathcal{L}_{(\mathbb{D}_s,\mathbb{D}_t)}$ and $\mathcal{L}_{(\mathbb{D}_t,\mathbb{D}_s)}$. The total contrastive loss is written as

$$\mathcal{L}_{con} = \mathcal{L}_{(\mathbb{D}_s,\mathbb{D}_s)} + \mathcal{L}_{(\mathbb{D}_t,\mathbb{D}_t)} + \mathcal{L}_{(\mathbb{D}_s,\mathbb{D}_t)} + \mathcal{L}_{(\mathbb{D}_t,\mathbb{D}_s)}. \quad (11)$$

The total loss in the update process can be written as

$$\mathcal{L}_{final} = \mathcal{L}^{src}_{pred} + \mathcal{L}^{tar}_{pred} + \eta\mathcal{L}_{con}, \quad (12)$$

where $\eta \in (0, 1)$ is a trade-off hyper-parameter, $\mathcal{L}^{src}_{pred}$ denotes the application of $\mathcal{L}_{pred}$ on the source domain samples.

As a plug-and-play module, our STraj builds upon the existing predictors. During the inference process, STraj does not introduce extra computational complexity.

Experiments

Dataset. We evaluate our proposed STraj on the widely used trajectory prediction datasets Argoverse 1 (Chang et al. 2019). Argoverse 1 comprises more than 300K real-world driving sequences collected in two geographically diverse cities, *i.e.*, Miami (MIA) and Pittsburgh (PIT). The task of trajectory prediction in Argoverse 1 is to predict 3 seconds of future trajectories, given 2 seconds of previously observed motions and High-Definition (HD) maps. **Notably, we conduct cross-geography UDA experiments within the same dataset to eliminate interference from other domain gaps across different datasets.** Since each city has distinctive geographical idiosyncrasies, we denote each city as a separate domain. The detailed cross-geography UDA experiments on Argoverse 1 are as follows: MIA $\rightarrow$ PIT and PIT $\rightarrow$ MIA. We split half of the validation sets as the test sets for the convenience of separately evaluating each domain.

Competitive Methods. We compare STraj with five models: **Source Only** means directly applying the original model trained in the source domain to the target domain. **Oracle** represents that the predictor is supervised by the labeled target domain data with available ground truths. **ICON** (Yue, Sun, and Zhang 2023) is a SOTA UDA approach in the image fields through self-training. For UDA on trajectory prediction, **S-attack** (Bahari et al. 2022) boosts the UDA performance by generating realistic scenes, while **Frenét** (Ye, Zhou, and Wang 2023) specifically addresses domain shifts stemming from road geometry and topology. All methods build upon the same backbone predictors for a fair comparison.

Metrics. We evaluate all experimental results by the official Argoverse criteria, incorporating minimum average displacement error (minADE), minimum final displacement error (minFDE), and miss rate (MR). minADE is the minimum ADE of the predicted $\mathcal{K}$ trajectories, where ADE denotes the average L2 distance between each coordinate of the predicted trajectory and the ground truth. minFDE is the minimum FDE of the predicted $\mathcal{K}$ trajectories, where FDE denotes the L2 distance between the endpoint of the predicted trajectory and the ground truth. Miss rate (MR) calculates the ratio of scenarios where none of the predicted trajectories are within 2 meters of the ground truth according to FDE.

Implementation details. In the pre-training process, We set ρ_a, r and the weight of $\mathcal{L}_{MSE}$ as 1, 10 and 0.01, respectively. As for the update strategy, $T_{\mathcal{U}}$ and ρ_t are set as $\frac{\sqrt{3}}{2}$ and 2. We set T_c as a dynamic threshold, which exceeds half confidence scores of all target domain samples in the current epoch. In the trajectory-induced contrastive learning module, we set ρ_c of inter-domain and intra-domain ρ_c as 1 and 2, respectively. The trade-off parameter η is set as 0.1. Our STraj builds upon a popular predictor LaneGCN (Liang et al. 2020) and a state-of-the-art (SOTA) predictor HPNet(Tang et al. 2024) for Argoverse 1, following their default model parameters. In the pre-training and training process, we exploit Adam (Kingma and Ba 2014) with learning rate 1.5×10^{-3} for 30 epochs, and train the model on four A6000 GPUs.

Performance Comparison of Main Results. As shown in Table 1, our STraj surpasses all competitive predictors by convincing margins across various cross-geography tasks in most cases. For instance, in the MIA $\rightarrow$ PIT task, conducted on LaneGCN evaluated with $\mathcal{K}$=1, STraj decreases the MR from 0.755 to 0.685, and markedly outperforms the Source Only by around 15.5% and 18.5% in terms of minADE and minFDE. Notably, The generalization performance is generally higher when MIA is set as the source domain compared to PIT. Indeed, as depicted in Fig. 1 (a) and (b), due to the larger numbers of agents and lane segments in MIA as well as their more uniform distribution, the model trained in MIA is more adaptive than that trained in PIT. The above results validate the effectiveness of our STraj across different cross-geography UDA tasks in trajectory prediction.

Ablation Studies

In this section, to further investigate the influence of individual modules in STraj, we conduct several ablation studies, which are conducted on the MIA $\rightarrow$ PIT task with LaneGCN (Liang et al. 2020) and evaluated with $\mathcal{K}$=1.

Architecture Designs. As shown in Table 2, we compare the results of using different components. Generally, each component boosts the performance. Specifically, compared with Source Only, Augmentation, PLUS and TCL provide 2.6%, 7.3% and 5.6% (2.9%, 8.1% and 7.5%) performance gains in terms of minADE (minFDE). PLUS contributes the most to model performance, which indicates the superiority of our pseudo label update strategy.

Analysis of Augmentation. We investigate the effectiveness of each module in complementary augmentations. As shown in Table 3, when only implementing agent or map augmentation, the performance slightly drops. Without weight, random masking achieves the worst performance. It indicates that agent augmentation, map augmentation and weighted random masking all enhance the performance. Notably, weighted random masking plays a more vital role than the others.

Analysis of Consistency Variants. To evaluate the superiority of our measurement method for trajectory consistency, we propose four additional variants. As shown in Table 4, the

Task	Method	LaneGCN		HPNet	
		$\mathcal{K}$=1	$\mathcal{K}$=6	$\mathcal{K}$=1	$\mathcal{K}$=6
		minADE↓/minFDE↓/MR↓	minADE↓/minFDE↓/MR↓	minADE↓/minFDE↓/MR↓	minADE↓/minFDE↓/MR↓
MIA → PIT	Source Only	2.32 / 5.46 / 0.755	1.34 / 2.67 / 0.455	2.05 / 5.21 / 0.593	1.09 / 2.28 / 0.281
	ICON	2.51 / 5.62 / 0.764	1.30 / **2.41** / 0.429	1.94 / 4.92 / 0.575	**0.947** / 2.09 / 0.227
	S-attack	2.19 / 4.98 / 0.749	1.33 / 2.63 / 0.442	1.76 / 3.89 / 0.488	1.12 / 2.23 / 0.258
	Frenét	2.23 / 4.84 / 0.707	1.36 / 2.58 / 0.426	1.79 / 4.18 / 0.552	1.02 / 2.39 / 0.244
	STraj (ours)	**1.96** / **4.45** / **0.685**	**1.26** / 2.59 / **0.415**	**1.55** / **3.72** / **0.524**	0.960 / **1.84** / **0.193**
	Oracle	1.50 / 3.40 / 0.574	0.872 / 1.49 / 0.172	1.40 / 3.30 / 0.529	0.758 / 1.22 / 0.167
PIT → MIA	Source Only	2.68 / 7.24 / 0.850	2.08 / 4.82 / 0.695	2.74 / 5.82 / 0.823	1.96 / 3.89 / 0.359
	ICON	**2.62** / 7.06 / **0.825**	1.89 / 4.31 / 0.532	2.61 / 5.33 / 0.783	1.77 / 3.19 / 0.339
	S-attack	2.96 / 7.05 / 0.859	1.73 / 3.79 / 0.471	2.71 / 5.43 / 0.816	1.51 / 2.90 / 0.342
	Frenét	2.74 / 7.17 / 0.841	1.55 / 3.68 / 0.459	2.63 / 5.29 / 0.763	1.68 / 3.12 / 0.348
	STraj (ours)	2.89 / **6.90** / 0.836	**1.41** / **2.79** / **0.404**	**2.56** / **4.87** / **0.715**	**1.26** / **2.25** / **0.323**
	Oracle	1.72 / 3.84 / 0.611	0.873 / 1.42 / 0.173	1.64 / 3.41 / 0.580	0.787 / 1.35 / 0.168

Table 1: Performance comparison on test sets. We use minADE, minFDE and MR for $\mathcal{K}$=1 and $\mathcal{K}$=6 as the main metrics. The gray values of the Oracle model are obtained in a supervised manner. We obtain the performance values of ICON, S-attack and Frenét based on their official codes. Best performances are in **bold**. Unit for ADE and FDE: meter.

Method	minADE↓	minFDE↓	MR↓
Source Only	2.32	5.46	0.755
Augmentation	2.26	5.30	0.712
Augmentation + PLUS	2.09	4.86	0.694
Augmentation + PLUS + TCL	**1.96**	**4.45**	**0.685**

Table 2: Ablation studies of different components. Augmentation denotes the complementary augmentations during pre-training. PLUS denotes the pseudo label update strategy, and TCL denotes the trajectory-induced contrastive learning.

Method	minADE↓	minFDE↓	MR↓
Agent augmentation	2.11	4.93	0.699
Map augmentation	2.05	4.79	0.690
Random masking	2.19	5.16	0.710
Original augmentation	**1.96**	**4.45**	**0.685**

Table 3: Effectiveness analysis of the Augmentation module.

Method	minADE↓	minFDE↓	MR↓
$\mathbf{V_4}$	2.16	5.13	0.710
$\mathbf{V_3}$	2.21	5.19	0.706
$\mathbf{V_2}$	2.20	5.09	0.701
$\mathbf{V_1}$	2.02	4.68	0.692
Ours	**1.96**	**4.45**	**0.685**

Table 4: Analysis of consistency variants. Distinguished from our approach, which calculates the cosine similarity of the entire pseudo trajectories, $\mathbf{V_1}$ calculates the Euclidean distance, while $\mathbf{V_2}$ takes into account both the Euclidean distance and the cosine similarity. $\mathbf{V_3}$ and $\mathbf{V_4}$ calculate the cosine similarity and the Euclidean distance of the endpoints, respectively.

cosine similarity (ours) outperforms the Euclidean distance ($\mathbf{V_1}$). It demonstrates that the direction is more important than the size in this regard, as illustrated in Fig. 4 (b) and (c). $\mathbf{V_2}$ yields inferior results, due to the elimination of excessive pseudo labels that only possess similar directions or sizes. $\mathbf{V_3}$ and $\mathbf{V_4}$ achieve worse performance as they ignore

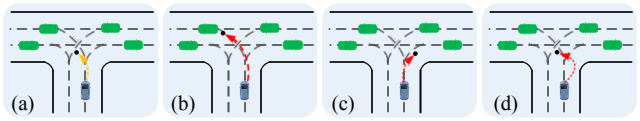

Figure 4: (a) is the pseudo trajectory from the historical epoch. (b), (c), and (d) represent the most consistent trajectories with (a) at the current epoch, as measured by the cosine similarity, the Euclidean distance, and the endpoints, respectively. As shown in (b), the cosine similarity mainly considers the trajectory direction while ignoring the size, which is contrary to the Euclidean distance as shown in (c).

Intra-domain	Inter-domain	minADE↓	minFDE↓	MR↓
		2.15	4.94	0.705
✓		2.06	4.73	0.690
	✓	2.09	4.82	0.693
✓	✓	**1.96**	**4.45**	**0.685**

Table 5: Effectiveness analysis of contrastive loss.

intermediate points as shown in Fig. 4(d).

Analysis of Trajectory-induced Contrastive Loss. As presented in Table 5, both the inter-domain and intra-domain contrastive losses contribute to the performance gains. The combination of intra-domain contrastive loss with inter-domain contrastive loss achieves the best results, thereby validating the efficacy of trajectory-induced contrastive learning.

Conclusion

In this paper, we have presented a new self-training pipeline for cross-geography trajectory prediction, named STraj. STraj incorporates agent and map augmentations during the pre-training process, an update strategy for pseudo labels, and trajectory-induced contrastive learning. Extensive experiments across various adaptation tasks along with comprehensive ablation analysis validate the effectiveness and generalization of our STraj.

Acknowledgements

This work was supported in part by The National Nature Science Foundation of China (Grant Nos.: 62303406, 62273302, 62036009, 61936006, 62273303), in part by Yongjiang Talent Introduction Programme (Grant Nos.: 2023A-194-G, 2022A-240-G), in part by Ningbo Key R&D Program (Nos.: 2023Z231, 2023Z229).

References

Bahari, M.; Saadatnejad, S.; Rahimi, A.; Shaverdikondori, M.; Shahidzadeh, A. H.; Moosavi-Dezfooli, S.-M.; and Alahi, A. 2022. Vehicle trajectory prediction works, but not everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17123–17133.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, volume 28.

Benigmim, Y.; Roy, S.; Essid, S.; Kalogeiton, V.; and Lathuilière, S. 2023. One-shot unsupervised domain adaptation with personalized diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 698–708.

Bhattacharyya, P.; Huang, C.; and Czarnecki, K. 2023. Ssllanes: Self-supervised learning for motion forecasting in autonomous driving. In *Conference on Robot Learning*, 1793–1805. PMLR.

Casas, S.; Sadat, A.; and Urtasun, R. 2021. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14403–14412.

Chai, Y.; Sapp, B.; Bansal, M.; and Anguelov, D. 2019. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*.

Chang, M.-F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. 2019. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8748–8757.

Chen, H.; Wang, J.; Shao, K.; Liu, F.; Hao, J.; Guan, C.; Chen, G.; and Heng, P.-A. 2023. Traj-MAE: Masked Autoencoders for Trajectory Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8351–8362.

Cheng, J.; Mei, X.; and Liu, M. 2023. Forecast-MAE: Self-supervised Pre-training for Motion Forecasting with Masked Autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8679–8689.

Chhabra, S.; Venkateswara, H.; and Li, B. 2023. Generative Alignment of Posterior Probabilities for Source-free Domain Adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4125–4134.

Du, Z.; Li, X.; Li, F.; Lu, K.; Zhu, L.; and Li, J. 2024. Domain-agnostic mutual prompting for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23375–23384.

Feng, T.; Shi, H.; Liu, X.; Feng, W.; Wan, L.; Zhou, Y.; and Lin, D. 2024. Open compound domain adaptation with object style compensation for semantic segmentation. *Advances in Neural Information Processing Systems*, 36.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Gu, X.; Song, G.; Gilitschenski, I.; Pavone, M.; and Ivanovic, B. 2024. Producing and leveraging online map uncertainty in trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14521–14530.

Ha, S.; and Jeong, H. 2023. Learning Heterogeneous Interaction Strengths by Trajectory Prediction with Graph Neural Network. In *The Eleventh International Conference on Learning Representations*.

Hu, Y.; Jia, X.; Tomizuka, M.; and Zhan, W. 2022. Causal-based time series domain generalization for vehicle intention prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, 7806–7813. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kothari, P.; Li, D.; Liu, Y.; and Alahi, A. 2023. Motion style transfer: Modular low-rank adaptation for deep motion forecasting. In *Conference on Robot Learning*, 774–784. PMLR.

Lan, Z.; Jiang, Y.; Mu, Y.; Chen, C.; and Li, S. E. 2023. Sept: Towards efficient scene representation learning for motion prediction. In *The Twelfth International Conference on Learning Representations*.

Li, R.; Li, C.; Ren, D.; Chen, G.; Yuan, Y.; and Wang, G. 2024. Bcdiff: Bidirectional consistent diffusion for instantaneous trajectory prediction. *Advances in Neural Information Processing Systems*, 36.

Liang, J.; Jiang, L.; and Hauptmann, A. 2020. Simaug: Learning robust representations from simulation for trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, 275–292. Springer.

Liang, M.; Yang, B.; Hu, R.; Chen, Y.; Liao, R.; Feng, S.; and Urtasun, R. 2020. Learning lane graph representations for motion forecasting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 541–556. Springer.

Liao, H.; Li, Z.; Shen, H.; Zeng, W.; Liao, D.; Li, G.; and Xu, C. 2024. Bat: Behavior-aware human-like trajectory prediction for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10332–10340.

Liu, M.; Cheng, H.; Chen, L.; Broszio, H.; Li, J.; Zhao, R.; Sester, M.; and Yang, M. Y. 2024. Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2039–2049.

Liu, Y.; Cadei, R.; Schweizer, J.; Bahmani, S.; and Alahi, A. 2022. Towards robust and adaptive motion forecasting: A causal representation perspective. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 17081–17092.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, 97–105. PMLR.

Lu, J.; Zhan, W.; Tomizuka, M.; and Hu, Y. 2022. Generalizability analysis of graph-based trajectory predictor with vectorized representation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 13430–13437. IEEE.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Park, D.; Jeong, J.; and Yoon, K.-J. 2024. Improving transferability for cross-domain trajectory prediction via neural stochastic differential equation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10145–10154.

Park, S. H.; Lee, G.; Seo, J.; Bhat, M.; Kang, M.; Francis, J.; Jadhav, A.; Liang, P. P.; and Morency, L.-P. 2020. Diverse and admissible trajectory forecasting through multimodal context understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 282–298. Springer.

Peng, D.; Ke, Q.; Ambikapathi, A.; Yazici, Y.; Lei, Y.; and Liu, J. 2024. Unsupervised Domain Adaptation via Domain-Adaptive Diffusion. *IEEE Transactions on Image Processing*, 33: 4245–4260.

Qian, T.; Chen, Y.; Cong, G.; Xu, Y.; and Wang, F. 2024. AdapTraj: A Multi-Source Domain Generalization Framework for Multi-Agent Trajectory Prediction. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 5048–5060. IEEE.

Shen, F.; Xie, Y.; Zhu, J.; Zhu, X.; and Zeng, H. 2023. Git: Graph interactive transformer for vehicle re-identification. *IEEE Transactions on Image Processing*, 32: 1039–1051.

Sun, H.; Xu, L.; Jin, S.; Luo, P.; Qian, C.; and Liu, W. 2024. PROGRAM: PROtotype GRAph Model based Pseudo-Label Learning for Test-Time Adaptation. In *The Twelfth International Conference on Learning Representations*.

Tang, X.; Kan, M.; Shan, S.; Ji, Z.; Bai, J.; and Chen, X. 2024. Hpnet: Dynamic trajectory forecasting with historical prediction attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15261–15270.

Tsai, Y.-Y.; Chen, F.-C.; Chen, A. Y.; Yang, J.; Su, C.-C.; Sun, M.; and Kuo, C.-H. 2024. GDA: Generalized Diffusion for Robust Test-time Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23242–23251.

Varadarajan, B.; Hefny, A.; Srivastava, A.; Refaat, K. S.; Nayakanti, N.; Cornman, A.; Chen, K.; Douillard, B.; Lam, C. P.; Anguelov, D.; et al. 2022. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, 7814–7821. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Y.; Zhang, P.; Bai, L.; and Xue, J. 2023. FEND: A Future Enhanced Distribution-Aware Contrastive Learning Framework for Long-tail Trajectory Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1400–1409.

Wu, Y.; Li, Z.; Wang, C.; Zheng, H.; Zhao, S.; Li, B.; and Tao, D. 2024. Domain re-modulation for few-shot generative domain adaptation. *Advances in Neural Information Processing Systems*, 36.

Xu, Y.; and Fu, Y. 2024. Adapting to length shift: Flexilength network for trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15226–15237.

Xu, Y.; Wang, L.; Wang, Y.; and Fu, Y. 2022. Adaptive trajectory prediction via transferable gnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6520–6531.

Ye, L.; Zhou, Z.; and Wang, J. 2023. Improving the generalizability of trajectory prediction models with frenet-based domain normalization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11562–11568. IEEE.

Ye, M.; Cao, T.; and Chen, Q. 2021. Tpcn: Temporal point cloud networks for motion forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11318–11327.

Yue, Z.; Sun, Q.; and Zhang, H. 2023. Make the u in uda matter: Invariant consistency learning for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 36: 26991–27004.

Zhang, Z.; Chen, M.; Xiao, S.; Peng, L.; Li, H.; Lin, B.; Li, P.; Wang, W.; Wu, B.; and Cai, D. 2024a. Pseudo Label Refinery for Unsupervised Domain Adaptation on Cross-dataset 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15291–15300.

Zhang, Z.; Hua, Z.; Chen, M.; Lu, W.; Lin, B.; Cai, D.; and Wang, W. 2024b. G2LTraj: A Global-to-Local Generation Approach for Trajectory Prediction. *arXiv preprint arXiv:2404.19330*.

Zhao, H.; Gao, J.; Lan, T.; Sun, C.; Sapp, B.; Varadarajan, B.; Shen, Y.; Shen, Y.; Chai, Y.; Schmid, C.; et al. 2021. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, 895–904. PMLR.

Zhao, J.; Xu, J.; Xu, Y.; Fang, J.; Chao, P.; and Zhou, X. 2024. CCML: Curriculum and Contrastive Learning Enhanced Meta-Learner for Personalized Spatial Trajectory Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 36: 4499–4514.