TOWARDS SINGLE SOURCE DOMAIN GENERALISATION IN TRAJECTORY PREDICTION: A MOTION PRIOR BASED APPROACH

Renhao Huang, Anthony Tompkins, Maurice Magnucco, Yang Song
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
{renhao.huang,anthony.tompkins,morri,yang.song1}@unsw.edu.au

ABSTRACT

Trajectory prediction is an important task in many real-world applications. However, data-driven approaches typically suffer from dramatic performance degradation when applied to unseen environments due to the inevitable domain shift brought by changes in factors such as pedestrian walking speed and the geometry of the environment. In particular, when a dataset does not contain sufficient samples to determine prediction rules, the trained model can easily consider some important features as domain variant. We propose a framework that integrates a simple motion prior with deep learning to achieve, for the first time, exceptional single-source domain generalisation for trajectory prediction, in which deep learning models are only trained using a single domain and then applied to multiple novel domains. Instead of predicting the exact future positions directly from the model, we first assign a constant velocity motion prior to each pedestrian and then learn a conditional trajectory prediction model to predict residuals to the motion prior using auxiliary information from the surrounding environment. This strategy combines deep learning models with knowledge priors to simultaneously simplify training and enhance generalisation, allowing the model to focus on disentangling data-driven spatio-temporal factors while not overfitting to individual motions. We also propose a novel Train-on-Best-Motion strategy that can alleviate the adverse effects of domain shift, brought on by changes in environment, by exploiting invariances inherent to the choice of motion prior. Experiments across multiple datasets of different domains demonstrate that our approach reduces the influence of domain shift and also generalizes better to unseen environments.

1 INTRODUCTION

Trajectory prediction in autonomous systems involves predicting the future movement of agents such as vehicles and pedestrians. Predicting human trajectories is challenging due to the complex social "forces" (Helbing & Molnár, 1995) among agents and the high requirement for terrain compliance. Many previous works capture the interactions between agents using pooling (Alahi et al., 2016; Gupta et al., 2018). More advanced approaches use attention mechanisms (Sadeghian et al., 2019; Kosaraju et al., 2019; Vemula et al., 2018; Amirian et al., 2019) that assign adaptive attention weights to neighbours and aggregate them together as new features. Some approaches (Mohamed et al., 2020; Shi et al., 2021) build graphs as their inputs and manually define connections according to the distances between each pair of trajectories.

A single trajectory prediction dataset may be biased to certain behaviours, which would lead to incorrect priors that are implicitly learned when a model overfits a strongly biased training dataset. For example, if all trajectories in a dataset follow a similar displacement change, the model would have difficulty in learning the speed relationship between the observed and ground truth paths which can cause significant performance drops in different environments. To tackle this problem, some recent works attempt to enhance model robustness by obtaining domain-invariant features from observations. For example, Chen et al. (2021) indicate that performance drops are brought by environmental biases among different locations and proposes counterfactual inference to alleviate it. Liu et al. (2022) further disentangle this problem into eliminating spurious features via invariant risk minimisation (Arjovsky et al., 2019) and learning awareness of style shifts among different datasets using contrastive learning. These methods belong to the problem class of *multi-source domain generalisation*, which requires more than one domain during training and hence increases the complexity of data collection and annotation. Another work (Xu et al., 2022b) proposes an unsupervised *domain adaptation* method via the use of a transferable Graph Neural Network (T-GNN) which aligns attention weights between

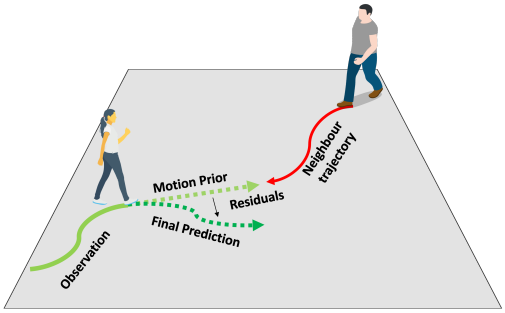


Figure 1: Overview of our motion prior approach. We assign a physical motion prior to the pedestrian and predict the residuals according to the surrounding environment.

source and target domains. Ivanovic et al. (2022) further propose a test-time adaptation strategy by using adaptive meta-learning (Harrison et al., 2018). However, these methods require the test domain information during training which is not practical in real settings.

In this work, we explore a more practical problem: *single-source domain generalisation*. We follow a similar setting to the *single-source domain adaptation* study (Xu et al., 2022b) but explore a harder problem in which only a *single* training domain is provided but evaluated on *multiple* testing domains. This thus becomes a domain generalisation problem with a single source and multiple target domains. This problem becomes more challenging since no other domains can be used to find domain invariant features. To tackle this problem, we emphasise the role of *handcrafted motion priors* when training on a single source where the dataset has an absence of informative data to train the model, which is neglected in current adaptation works. Firstly, a motion prior can contain basic social rules that may not be represented in raw observational datasets. Consequently, some domain variant features can be directly handled by the motion prior and thus reduce the risk of domain shift. Second, using a motion prior constrains the model to update gradients only when the handcrafted motion rules are not effective which simplifies the training and encourages the models to search for useful features from the surrounding environment. Third, this strategy can be simply integrated with any trajectory prediction model and can consistently improve their performance without extra complexity.

We propose a robust trajectory prediction framework that can exploit knowledge from a *motion prior* model alongside a learnable data-driven model. Our framework consists of a physical motion model that can predict trajectories using theoretically justified social rules and a data-driven trajectory prediction model to learn high-level spatio-temporal interactions and predict residual corrections to the motion prior. A simple physical model for the motion prior is sufficient in most cases and in this work, we choose the constant velocity motion (CVM) (Schöller et al., 2020), which encodes the prior knowledge that the future path of a pedestrian follows the same speed and direction of the previous steps. Then, we use a deep learning model (e.g., Social-STGCNN (Mohamed et al., 2020)) to extract features from the environment to model the scene and social interactions and predict residuals that can be aggregated on the motion prior for the prediction. We also find that CVM may not handle the static scene interaction and the scene features become domain-specific prior. Therefore, we further propose a training strategy called "Train-on-Best-Motion", which assigns the motion prior closest to the ground truth as our motion prior during training with the hypothesis that the trajectory direction can be directly affected by the scene terrains. This strategy successfully alleviates domain shifts due to the environment without using a scene interaction module, which is useful when no scene image is provided or the scene interaction module cannot work well.

In our experiments, our method outperforms current state-of-the-art models (Xu et al., 2022b; Ivanovic et al., 2022; Chen et al., 2021) on multiple pedestrian trajectory prediction datasets, indicating better generalisation than those models specifically designed for robustness. Furthermore, we build a synthetic dataset containing constant velocity motions with controlled speed and further demonstrate that our model can be generalisable to environments with different speeds of motion and environmental bias. Finally, we find that our method is also useful for autonomous driving datasets (NuScenes $\rightarrow$ Lyft) and dealing with domain shifts due to different sampling frequencies. We summarise our contributions as follows:

- We propose a single source domain generalisation framework based on physical motion priors, which largely enhance the model's generalisation ability. Our method is simple and effective and can be generally integrated with existing trajectory prediction methods.

- We propose a "Train-on-Best-Motion" training strategy that alleviates the domain shifts due to scene constraints.
- Comprehensive experiments show that our model outperforms many existing trajectory prediction models and can generalise well in unseen environments even when training on only a single source domain.

2 Related Work

Human Trajectory Prediction. To model human walking behaviour, many approaches have been proposed throughout the years. Prior to deep learning approaches, the social force model (Helbing & Molnár, 1995) was the main paradigm in modelling human walking behaviour. It describes social rules as "forces" that make pedestrians change their routes. Following this, many deep learning-based approaches (Bierlaire, 1998; Lerner et al., 2007; Yang & Peters, 2019; Yamaguchi et al., 2011) have been proposed to model human motion using public datasets. Social-LSTM (Alahi et al., 2016) is a Long Short-Term Memory (LSTM) structure with grid-based pooling mechanism to model the interaction information. Gupta et al. (2018) improve the pooling mechanism using a max operation. Further, SoPhie Sadeghian et al. (2019) describes the environmental constraints as a combination of scene and social constraints and applies soft attention to dynamically extract features. Following this strategy, many models (Amirian et al., 2019; Huang et al., 2019; Kosaraju et al., 2019; Vemula et al., 2018) propose different attention-based mechanisms to learn to rank the important neighbours. Similarly, Mohamed et al. (2020); Shi et al. (2021) propose a spatial-temporal graph convolution neural network (STGCN) to model the social interactions directly on graphs. Recent work such as (Sun et al., 2020; Xu et al., 2022a) design more complex graph models to enhance relationship learning. Meanwhile, many methods are proposed to model the uncertainty of trajectory prediction. For example, Gupta et al. (2018); Amirian et al. (2019); Kosaraju et al. (2019) propose Generative Adversarial Network (Goodfellow et al., 2014) based generative models to propose multiple trajectories while Mangalam et al. (2020; 2021) propose endpoint conditioned predictions. Dendorfer et al. (2021) use multiple decoders to model discontinued manifolds in trajectory prediction.

Robust Trajectory Prediction. In trajectory prediction tasks, we usually follow the "leave-one-out" strategy that uses multiple datasets for training and an unseen dataset for evaluation. Therefore, most works assume that the distribution of the training data is consistent with the target domain. However, recent work (Xu et al., 2022b; Chen et al., 2021; Liu et al., 2022; Ivanovic et al., 2022) demonstrate that the domain difference between the training and testing datasets can lead to a severe performance drop due to over-fitting on the domain variant features. These works are thus designed to improve the robustness of the methods and generally belong to two types: domain generalisation and domain adaptation, differentiated by the usage of testing data.

The domain generalisation approaches include those using causal inference to reduce spurious features. For example, Chen et al. (2021) suggest that models learn spurious features in trajectory prediction datasets due to terrain constraints. Therefore, they use counterfactual learning to eliminate those features independent of the observed trajectories. A further work (Liu et al., 2022) reformulates the causal relationship by reducing the spurious features using invariance risk minimisation (Arjovsky et al., 2019) and then transferring their styles into target domains. Another approach called SimAug (Liang et al., 2020a) increases model robustness under different views using simulated scenes (Liang et al., 2020b) with adversarial data augmentation. Meanwhile, domain adaptation-based methods focus on aligning the features between source and target domains given target domain data. For example, Transferable GNN (Xu et al., 2022b) is proposed to align the features while Huang et al. (2021) use adversarial domain adaptation to discriminate domain variant features. Ivanovic et al. (2022) tackle test-time domain adaptation in trajectory prediction using adaptive meta-learning and achieves state-of-the-art results in the cross-domain evaluation.

Our work focuses on single-source domain generalisation where only one domain can be seen during training. This is more challenging than domain adaptation and multi-source generalisation. Also, using only one domain makes (Liu et al., 2022) not suitable for this task. Counterfactual learning (Chen et al., 2021) can be a solution but our experiments show that it cannot achieve the desired results. Our solution injects a physically informed motion prior to a deep learning model and achieves the best result.

Prior Knowledge in Trajectory Prediction. Injecting handcrafted knowledge is an effective way to enhance model performance and robustness. For example, some models such as (Bansal et al., 2018; Park et al., 2020; van der Heiden et al., 2019) propose loss functions to constrain the model prediction to follow social or traffic rules. Chai et al. (2020); Phan-Minh et al. (2020); Zhao & Wildes (2021) inject the knowledge by building a fixed set of trajectories sampled from the training dataset, named "anchor sets". Then, they learn to rank top K "anchors" from this set and regress residuals on each of them along with uncertainties. In human trajectory prediction, Kothari et al. (2021) explicitly define the social interactions as group following, collision avoidance and leader following and learns to regress residuals via a discrete choice model (Bierlaire, 1998) on trajectories predicted from (Gupta et al., 2018). However, we suggest that knowledge priors in the above methods highly depend on the training set where the domain

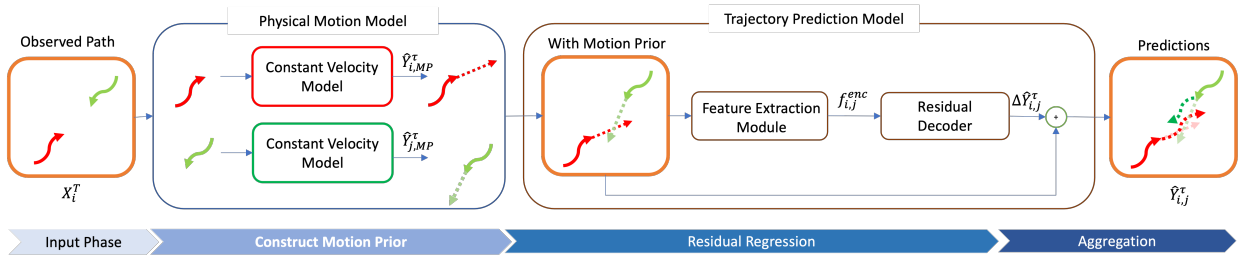


Figure 2: An overview of our framework. Given pedestrians' observed trajectories, we first assign a motion prior using a **physical motion model**. Then we send them together with observed trajectories into a deep learning based **trajectory prediction model** to predict the residuals. The final prediction is the aggregation between the motion prior and the residuals.

shift problem cannot be solved. Therefore, our knowledge prior is generated by a pure physical model which is not tied to the training set. A work similar to ours is (Bahari et al., 2021) which uses the lanes of the road to form the scene-compliant trajectory. However, human trajectory prediction does not have such information and hence we choose the constant velocity motion as our motion prior which is suitable for pedestrians and can successfully increase the robustness of the model across different domains.

3 Methods

Our approach is a motion prior based deep learning framework that follows two core assumptions for future path prediction: (1) *If there is no information available from either the surrounding environment or pedestrians' social cues, pedestrians should follow a motion prior (e.g., constant velocity) in physical space.* This is a reasonable assumption for pedestrian walking behaviour and essential for a model where only the past trajectory is provided, which is also suggested by Monti et al. (2022). (2) *Pedestrians change their routes due to the environment.* This assumption is consistent with the concept of social force (Helbing & Molnár, 1995) and suggests that environmental influence should directly contribute to the route changing on a motion prior rather than predicting the entire trajectory. We describe the details in the following sections.

3.1 Problem Definition

Recall that our goal is to generate future paths $\hat{Y}_i^\tau$ for agent i using the observed sequence X_i^τ. Mathematically, we formulate our problem as observing the motion states for N agents in a scene from the timestep $t = 1$ to t_{obs} as $X_i^\tau \in \{X_1^t, ..., X_N^t | t = 1, 2, ..., t_{obs}\}$.

We follow previous works (Gupta et al., 2018; Huang et al., 2019; Mohamed et al., 2020; Shi et al., 2021) in using the vectors at each timestep to normalise the trajectory, each with the source originated at the position at the previous time-step. Then, we predict the motion states from t_{obs} to a future time step t_{pred} as $\hat{Y}_i^\tau \in \{\hat{Y}_1^T, ..., \hat{Y}_N^T | T = t_{obs} + 1, ..., t_{pred}\}$ Finally, the sequence is evaluated with the ground truth as $Y_i^\tau \in \{Y_1^T, ..., Y_N^T | T = t_{obs} + 1, ..., t_{pred}\}$. Both X_i^t and Y_i^t are 2D vectors related to the previous positions. $\left(x_i^t - x_i^{t-1}, y_i^t - y_i^{t-1}\right)$.

To avoid ambiguity, we follow Sadeghian et al. (2019) in using *social interactions* and *scene interactions* to describe interactions between dynamic agents and the static objects or terrain in this work. We also use environmental bias in (Chen et al., 2021) to describe the domain shift problem brought by different environments.

3.2 Framework Overview

Figure 2 shows the overall structure of our approach, which consists of two parts: (1) a **physical motion prior** F_{MP}, and (2) a deep learning based **human trajectory prediction model** F_{HTP}. Given the observed trajectory X_i^τ, we first use the physical motion model to assign motion priors $\hat{Y}_{i,MP}^\tau$ as initial predictions to each pedestrian according to their current speed and direction. Then, we concatenate the motion prior with X_i^τ and feed them into F_{HTP}. F_{HTP} is expected to extract motion and interaction features from the surrounding agents and the static scene denoted as f_i^{enc}. The feature extraction process can be defined as follows:

$$f_i^{enc} = F_{HTP}([X_{i=1..N}^\tau, \hat{Y}_{i,MP}^\tau], I), \tag{1}$$

where I denotes the scene image if provided and $[\cdot,\cdot]$ is the concatenation operation. Finally, a trajectory decoder D takes f_i^{enc} to predict the residuals $\Delta\hat{Y}_{i,MP}^{\tau}$ which are then aggregated with the assigned motion prior to give the predicted trajectory:

$$\hat{Y}_i^{\tau} = \Delta\hat{Y}_{i,MP}^{\tau} + \hat{Y}_{i,MP}^{\tau} = D(f_i^{enc}) + \hat{Y}_{i,MP}^{\tau}, \tag{2}$$

where $\Delta\hat{Y}_{i,MP}^{\tau}$ indicates that the residuals can be predicted by conditioning on the input motion prior. We incorporate this data-driven learning process so that different types of motions can be accommodated during inference. Furthermore, we propose a "Train-on-Best-Motion" training strategy to mitigate any environmental biases.

3.3 Method Components

Physical Motion Model. Previous works such as (Chai et al., 2020) assume that the training and testing datasets share similar patterns of motion. Thus, they can cluster the training trajectories as their prototypes and use them as priors during the inference. However, they suffer from domain shifts from out-of-domain distributions where inconsistent motion (e.g., speed) can be observed across different environments. Therefore, we require that *our motion prior should estimate such motion shifts under different domains*. In this work, we resort to a well-studied constant velocity model (CVM) (Schöller et al., 2020) which hypothesises that the speed of the predicted trajectory is consistent with its historical one. Moreover, we suggest that the fundamental assumption taken by the CVM is reasonable as it is invariant to social cues and environmental factors, which also satisfies our first assumption that our framework follows and Fig. 3 demonstrates the effectiveness of our proposed CVM-based motion prior. Specifically, given an observed trajectory X_i^{τ}, we expand the last observed motion $X_i^{t_{obs}}$ through all future timesteps and our motion prior becomes $\hat{Y}_{i,MP}^{\tau} = \{X_i^{t_{obs}}, \cdots, X_i^{t_{obs}}\}$. Alternatively, as a potential future work, we can explore more advanced motion priors that can further supplement required rules in trajectory prediction.

Trajectory Prediction Model. Our method presents a generalized framework that is suitable to integrate with most trajectory prediction models. In this work, our trajectory prediction model builds upon Social-STGCNN (Mohamed et al., 2020). Social-STGCNN is an advanced trajectory prediction model that can effectively model the social interactions using its spatial temporal GCN module and is a strong baseline used in (Xu et al., 2022b; Shi et al., 2021). Therefore, given the observed trajectories and corresponding motion priors, we follow Mohamed et al. (2020) to build the adjacency matrix sequence at each timestep using relative positions and preprocess it using Laplacian normalisation. Then, Social-STGCNN extracts social features as f_i^{enc} via a spatial temporal graph convolutional neural network, which are the node embeddings in the output graph.

Social-STGCNN predicts a sequence of bi-variate Gaussian distributions for each future positions, which can alleviate generation mode collapse as suggested by Mohamed et al. (2020); Chai et al. (2020). To integrate the motion prior, we build the Gaussian distribution as follows:

$$\hat{Y}_i^{\tau} \sim \mathcal{N}(\hat{Y}_{i,MP}^{\tau} + \Delta\mu_i^{\tau}, \sigma_i^{\tau}, \rho_i^{\tau}), \tag{3}$$

where $\hat{Y}_{i,MP}^{\tau} + \Delta\mu_i^{\tau}$, σ_i^{τ},ρ_i^{τ} are the mean, variance and correlation coefficient for bi-variate Gaussian distribution at timestep τ for agent i. Therefore, we expect the Social-STGCNN to predict the $\Delta\mu_i^{\tau}$ as the residuals.

To avoid the gradient vanishing problem caused by recurrent neural networks, Social-STGCNN (Mohamed et al., 2020) uses a time-extrapolator convolutional neural network (TXP-CNN) as its decoder which consists of multiple 3×3 Conv2d layers which use the time dimension as the channel dimension and slide kernels to fuse node feature embeddings from neighbouring agents. However, TXP-CNN suffers from a permutation variance problem where changing the order of the pedestrians may vary the results, which can be a potential risk for domain shift. We also argue that interactions among neighbours are not necessary during the decoding. Considering that Social-STGCNN is a non-autoregressive model that predicts for a very short horizon and its STGCN module already integrates the neighbour information, we believe that handling social interaction in the decoder is redundant and even increases the learning complexity. Therefore, we use a 3×1 Conv2d layer to slide the node feature embedding only:

$$\Delta\mu_i^{\tau}, \sigma_i^{\tau}, \rho_i^{\tau} = \text{TXP-CNN}_{3\times1}(f_i^{enc}). \tag{4}$$

Our experimental results also indicate that using a 3×1 can have a better performance. Finally, we learn the residual regression using the loss function as follows:

$$l(\theta) = -\log\mathcal{N}(\hat{Y}_{i,MP}^{\tau} + \Delta\mu_i^{\tau}, \sigma_i^{\tau}, \rho_i^{\tau}) + \lambda\log|\Sigma_i^{\tau}|, \tag{5}$$

where θ denotes the learnable weights in our revised Social-STGCNN model. The first part of this loss function is a standard negative Gaussian log-likelihood loss to search for the optimal Gaussian parameters for each predicted positions. We also expect the model to predict a low entropy distribution which leverages the term $\log|\Sigma_i^{\tau}|$ scaled by a coefficient λ, where $|\Sigma_i^{\tau}|$ is the determinant of the covariance matrix built by σ_i^{τ} and ρ_i^{τ}.

Method	Year	Performance (ADE_{20}) (Source2Target)																				Avg
		A2B	A2C	A2D	A2E	B2A	B2C	B2D	B2E	C2A	C2B	C2D	C2E	D2A	D2B	D2C	D2E	E2A	E2B	E2C	E2D	
Social-STGCNN (Mohamed et al., 2020)	2020	1.83	1.58	1.30	1.31	3.02	1.38	2.63	1.58	1.16	0.70	0.82	0.54	1.04	1.05	0.73	0.47	0.98	1.09	0.74	0.50	1.22
PECNet (Mangalam et al., 2020)	2020	1.97	1.68	1.24	1.35	3.11	1.35	2.69	1.62	1.39	0.82	0.93	0.57	1.10	1.17	0.92	0.52	1.01	1.25	0.83	0.61	1.31
RSBG (Sun et al., 2020)	2020	2.21	1.59	1.48	1.42	3.18	1.49	2.72	1.73	1.23	0.87	1.04	0.60	1.19	1.21	0.80	0.49	1.09	1.37	1.03	0.78	1.38
Tra2Tra (Xu et al., 2021)	2021	1.72	1.58	1.27	1.37	3.32	1.36	2.67	1.58	1.16	0.70	0.85	0.60	1.09	1.07	0.81	0.52	1.03	1.10	0.75	0.52	1.25
SGCN (Shi et al., 2021)	2021	1.68	1.54	1.26	1.28	3.22	1.38	2.62	1.58	1.14	0.70	0.82	0.52	1.05	0.97	0.80	0.48	0.97	1.08	0.75	0.51	1.22
T-GNN (Xu et al., 2022b)	2022	1.13	1.25	0.94	1.03	2.54	1.08	2.25	1.41	0.97	0.54	0.61	**0.23**	0.88	0.78	0.59	0.32	0.87	0.72	0.65	0.34	0.96
TTA-GNN (Ivanovic et al., 2022)	2022	**0.33**	0.56	0.50	0.38	0.80	0.60	0.43	**0.31**	1.03	0.41	0.41	0.38	0.93	0.32	0.48	0.35	0.91	0.31	0.49	0.44	0.52
Social-STGCNN*	2020	1.06	0.97	1.18	1.03	2.50	0.97	2.06	1.16	0.75	0.45	0.41	0.34	**0.66**	0.81	0.54	0.30	0.79	0.97	0.57	0.36	0.89
SGCN*	2021	1.01	0.77	0.70	0.55	1.77	0.91	1.63	0.97	**0.70**	0.37	0.46	0.34	0.84	0.55	0.50	**0.26**	**0.68**	0.47	0.53	0.30	0.71
CF-STGCNN* (Chen et al., 2021)	2021	1.14	0.73	0.78	0.63	1.64	0.77	1.44	0.83	0.97	0.58	0.43	0.33	0.90	0.91	0.59	0.31	0.84	1.02	0.63	0.43	0.79
Social-STGCNN + MP	-	0.65	0.69	0.69	0.66	**0.85**	**0.41**	0.43	**0.32**	0.72	0.29	0.32	0.26	0.78	0.23	**0.39**	0.27	0.82	0.32	0.40	**0.32**	0.49
Social-STGCNN + BMP	-	0.41	**0.45**	**0.49**	**0.43**	0.86	0.42	**0.41**	0.35	0.72	**0.23**	**0.32**	0.26	0.78	**0.24**	**0.39**	0.28	0.79	**0.22**	**0.38**	**0.32**	**0.44**

Method	Year	Performance (FDE_{20}) (Source2Target)																				Avg
		A2B	A2C	A2D	A2E	B2A	B2C	B2D	B2E	C2A	C2B	C2D	C2E	D2A	D2B	D2C	D2E	E2A	E2B	E2C	E2D	
Social-STGCNN (Mohamed et al., 2020)	2020	3.24	2.86	2.53	2.43	5.16	2.51	4.86	2.88	2.30	1.34	1.74	1.10	2.21	1.99	1.41	0.88	2.10	2.05	1.47	1.01	2.30
PECNet (Mangalam et al., 2020)	2020	3.33	2.83	2.53	2.45	5.23	2.48	4.90	2.86	2.22	1.32	1.68	1.12	2.20	2.05	1.52	0.88	2.10	1.84	1.45	0.98	2.29
RSBG (Sun et al., 2020)	2020	3.42	2.96	2.75	2.50	5.28	2.59	5.19	3.10	2.36	1.55	1.99	1.37	2.28	2.22	1.77	0.97	2.19	2.29	1.81	1.34	2.50
Tra2Tra (Xu et al., 2021)	2021	3.29	2.88	2.66	2.45	5.22	2.50	4.89	2.90	2.29	1.33	1.78	1.09	2.26	2.12	1.63	0.92	2.18	2.06	1.52	1.17	2.34
SGCN (Shi et al., 2021)	2021	3.22	2.81	2.52	2.40	5.18	2.47	4.83	2.85	2.24	1.32	1.71	1.03	2.23	1.90	1.48	0.97	2.10	1.95	1.52	0.99	2.29
T-GNN (Xu et al., 2022b)	2022	2.18	2.25	1.78	1.84	4.15	1.82	4.04	2.53	1.91	1.12	1.30	0.87	1.92	1.46	1.25	0.65	1.86	1.45	1.28	0.72	1.82
TTA-GNN (Ivanovic et al., 2022)	2022	0.65	1.18	1.06	0.81	**1.59**	1.19	0.89	0.64	1.98	0.81	0.93	0.84	1.81	0.57	1.01	0.70	1.71	0.54	1.02	0.96	1.04
Social-STGCNN*	2020	1.14	1.08	1.66	1.28	4.69	1.82	4.26	2.35	1.27	0.80	0.69	0.57	**1.20**	1.49	0.88	**0.42**	1.70	1.74	0.94	0.52	1.53
SGCN*	2021	1.77	1.36	1.27	0.97	2.93	1.68	3.10	1.80	**1.16**	0.65	0.88	0.64	1.80	1.06	0.89	0.46	**1.32**	0.84	0.94	0.52	1.30
CF-STGCNN* (Chen et al., 2021)	2021	1.88	1.09	1.28	0.94	2.48	1.29	2.38	1.37	1.75	0.96	0.76	0.58	1.49	1.54	1.08	0.51	1.56	1.75	1.11	0.71	1.32
Social-STGCNN + MP	-	0.76	0.89	0.85	0.78	1.66	0.68	0.67	0.54	1.49	0.48	**0.53**	0.45	1.66	0.35	0.67	0.49	1.75	0.53	**0.67**	**0.52**	0.82
Social-STGCNN + BMP	-	**0.41**	**0.52**	**0.56**	**0.50**	1.71	**0.60**	**0.53**	**0.46**	1.48	**0.32**	**0.53**	**0.44**	1.65	**0.34**	**0.66**	**0.47**	1.66	**0.32**	**0.67**	0.53	**0.72**

Table 1: ADE ↓ (top) results of our models with the motion prior in comparison with existing state-of-the-art baselines on 20 tasks. "2" represents from source domain to target domain. A, B, C, D, and E denote ETH, HOTEL, UNIV, ZARA1, and ZARA2, respectively. "*" denotes that these results are reproduced by us and methods in **bold** are our works: "-MP" denotes the method with the motion prior and "-BMP" denotes the model with motion prior and trained using Train-on-Best-Motion.

Train-on-Best-Motion. Pedestrians usually change their routes due to terrain constraints such as curved pathways and obstacles. Ideally, a trajectory prediction model such as (Sadeghian et al., 2019; Dendorfer et al., 2021; Xue et al., 2018; Manh & Alaghband, 2018) that can handle both social and scene interactions is expected in our framework since the scene interactions module can encourage the scene features to be domain invariant. However, the scene information is usually unknown in this work since our baselines only use 2D coordinates in the dataset and querying the scene features can make our framework less general. Besides, extracting domain invariant features from scene images is currently impossible in our work as all datasets we used are static scenes and provide only one image or even motion data only. As a result, these scene features become domain-specific features which are harmful to the model. To handle this problem, we propose a training trick called "Train-on-Best-Motion" to effectively incorporate scene features.

We consider that the influence of the scene will directly change the overall trajectory direction and hence we should select the "best" motion during training. Concretely, we build a motion set by generating several motions, each one rotated with a distinct degree according to the pedestrians' heading directions. We search the best one from the set with the minimum average displacement error to the ground truth as our motion prior during training. Therefore, when pedestrians change their directions due to the scene constraints, the motion prior can cover this information and let the model focus on the social interaction on that path and thus predict the offsets without considering the scene interaction. At inference time, we can either choose the motion with the heading direction as our prior or follow Chai et al. (2020) to dynamically select a certain number of motion priors as the anchors and predict residuals for each of them.

Model	**DA**	**MP**	ADE_{20}	FDE_{20}
Social-STGCNN			0.89	1.53
	✓		0.83	0.93
		✓	0.49	0.82
	✓	✓	**0.40**	**0.77**
SGCN			0.71	1.30
	✓		0.59	1.06
		✓	0.44	0.81
	✓	✓	**0.43**	**0.78**

Table 2: The average performance on 20 tasks of Social-STGCNN and SGCN w/w.o. motion prior (MP) and data augmentation (DA).

Model	Aug	Average Performance (ADE_{20}/FDE_{20})						Avg
		0 m/s	1 m/s	2 m/s	3 m/s	4 m/s	5 m/s	
Social-STGCNN		0.27/0.23	4.00/7.66	9.04/17.08	14.17/26.60	19.35/36.16	24.13/45.37	11.89/22.25
CF-STGCNN		0.86/1.11	2.25/3.98	5.38/10.06	8.60/16.03	11.41/20.62	14.45/25.40	7.16/12.87
SGCN		0.09/0.10	2.18/4.20	5.52/10.61	9.02/17.26	12.54/23.94	16.08/30.64	7.57/14.46
Social-STGCNN	✓	0.22/0.23	1.12/1.84	2.76/4.74	4.64/7.95	6.53/11.37	8.42/14.73	3.95/6.81
Social-STGCNN-MP	✓	0.09/0.13	0.28/0.40	0.58/0.93	1.01/1.56	1.51/2.31	2.08/3.24	0.92/1.43
Social-STGCNN-BMP	✓	**0.07/0.08**	**0.24/0.23**	**0.47/0.61**	**0.81/1.28**	**1.24/2.13**	**1.71/3.02**	**0.76/1.22**

Table 3: The average ADE/FDE scores against different training sets and different speed controls using our synthetic dataset. "Aug" means whether data augmentation is used during training. "-MP" and "-BMP" denote the model using the motion prior and trained on Train-on-Best-Motion respectively.

Method	Average ADE/FDE
Social-STGCNN	0.89/1.53
+ synthetic data only	0.47/0.87
+ pretrained on synthetic data	0.52/0.97
+ strong augmentation	0.83/0.93
+ pretrain	0.54/0.84
+ augmented on synthetic data	0.46/0.82
+ motion prior	**0.40/0.77**

Table 4: Average performance on pedestrian tasks using our synthetic data for pretraining and data augmentation.

4 Experiments

4.1 Experimental Settings

Datasets We use well-known pedestrian trajectory prediction datasets ETH (Lerner et al., 2007) & UCY (Pellegrini et al., 2010) as the main benchmark in our experiment, which contain 5 scenes in total: ETH, HOTEL, UNIV, ZARA1 and ZARA2, abbreviated as A, B, C, D and E respectively, with trajectories sampled every 0.4 seconds and recording the world coordinates of pedestrians. We also generated a synthetic dataset containing 6 sets of constant velocity trajectories with controlled velocity as $\{0, 1, 2, 3, 4, 5\}$ m/s based on the toy model in (Amirian et al., 2019), each containing motions with 30 different directions. This dataset is used to analyse the extraction of domain invariant features from the model. We can also use this dataset to augment or pretrain the model to enhance performance. Finally, we use two autonomous driving datasets named NuScenes (Caesar et al., 2020) and Lyft (Houston et al., 2020) collected in different cities with trajectories sampled at 2 Hz and 10 Hz respectively.

Evaluation Metrics. In this work, we use standard evaluation metrics including *Average Displacement Error* (ADE_k), the average L_2 distance between ground truth and predictions, *Final Displacement Error* (FDE_k), the distance between the last predicted positions with ground truth endpoints and, *Negative Log-Likelihood* (NLL), the probabilities of ground truth can be sampled from distributions.

Baselines We compare our model with state-of-the-art pedestrian trajectory prediction models including (1) **Social-STGCNN** (Mohamed et al., 2020), a well-known social interaction model in trajectory prediction and the baseline for (Shi et al., 2021; Xu et al., 2022b; Chen et al., 2021); (2) **SGCN** (Shi et al., 2021), a state-of-the-art social interaction model which improves Social-STGCNN; (3) **CF-STGCNN** (Chen et al., 2021), a multi-source domain generalisation model based on Social-STGCNN that uses counterfactual analysis to alleviate the domain shifts. We adapt the method for single-source domain adaptation and reproduce the CF-STGCNN by setting the input node features as zeros in Social-STGCNN as the counterfactual intervention and evaluate it under our evaluation protocol; (4) **T-GNN** (Xu et al., 2022b), a single source unsupervised domain adaptation framework built upon Social-STGCNN and using a transferable attention mechanism to align domains features; (5) **Trajectron++** (Salzmann et al., 2020), an CVAE-based autoregressive model and (6) **TTA-GNN** (Ivanovic et al., 2022), the most recent test-time domain adaptation model based on (Salzmann et al., 2020) and adaptive meta-learning (Harrison et al., 2018). We use its **k0** and **Adaptive** models for comparison, which are two variants of (Salzmann et al., 2020) before the adaptation. These models are not exposed to the ground truth of the target domains.

4.2 Experiments on Pedestrian Data

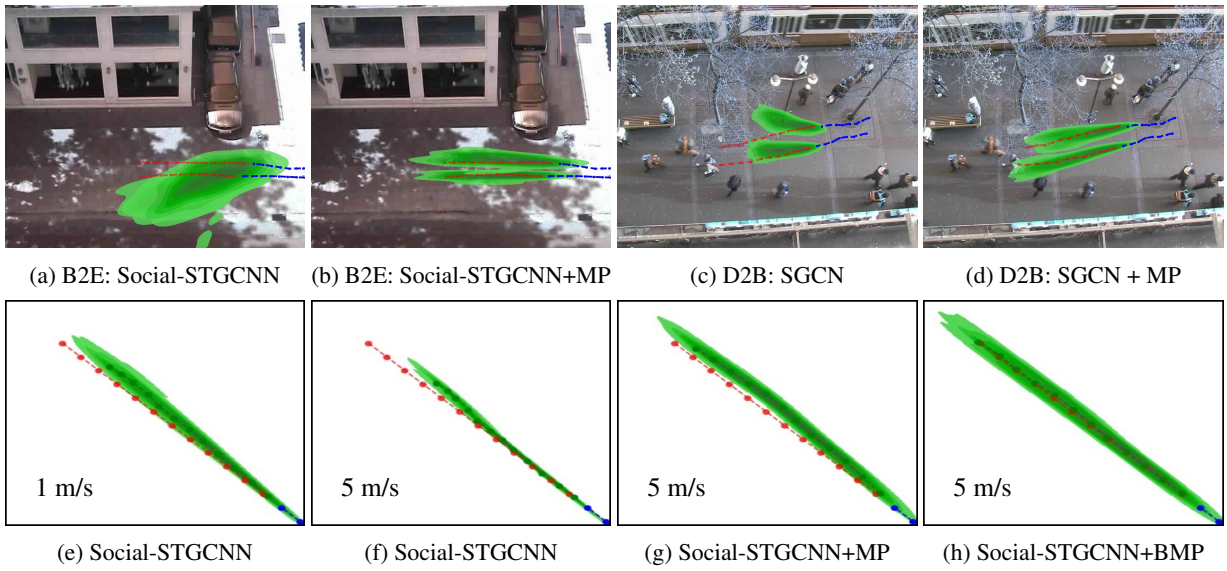
Evaluation Protocol and Experiment Configuration. In this section, we follow the evaluation protocol in (Xu et al., 2022b) to conduct 20 tasks in total on ETH & UCY datasets by training our model using one of these scenes as the source domain data and testing it using the remaining scenes as the target domain data. We observe 3.2 seconds (8 frames) for training and the next 4.8 seconds (12 frames) for testing. To ensure that the target domain is unseen during training, we use the source domain validation set to select the model during training. We also follow the multi-modal setting of our baselines (Gupta et al., 2018; Mohamed et al., 2020; Shi et al., 2021; Zhao & Wildes, 2021; Sun et al., 2020; Xu et al., 2022b) by using the best prediction among $k = 20$ samples for ADE and FDE metrics. However, to conduct a fair experiment, we build our environment using Social-STGCNN and further tune hyperparameters of Social-STGCNN, SGCN and CF-STGCNN to reach the best results in our experiment. All models are trained with 200 epochs and a batch size of 16 and learning rate of 0.001. We also set a gradient clip of 100 to avoid gradient explosion and coefficient λ is 0.5. Reproduced results can be seen in Table 1 with "*" near the model name. For "Train-on-Best-Motion" strategy, we select 5 constant velocity motions with rotated degrees of [-60, 30, 0, 30, 60] respectively from the heading direction as our motion set.

Benchmark Results vs Other Baselines. The performance of our methods against our basesline as shown Table 1. Our method that integrates a motion prior can outperform all baselines across different datasets, indicating the effectiveness of integrating motion priors alongside a deep learning model. A better performance of +10.2% can be achieved when using Train-on-Best-Motion which reaches state-of-the-art in this benchmark. Specifically, our model outperforms T-GNN by around 50% and the current state-of-the-art model TTA-GNN by around 6%, which validates that our method can be even better than current single source domain adaptation methods. Domains such as Hotel (B) contain different trajectories to others and hence the performance gaps between D2B and C2B using T-GNN and TTA-GNN are around 0.24/0.34 and 0.09/0.25 of ADE/FDE respectively. On the other hand, our method only produces a difference of only 0.06/0.13 in ADE/FDE after adding the motion and 0.01/0.02 after using the "Train-on-Best-Motion", which indicates our model can generalise better to unseen domains. We also note that our method is better than SF-STGCNN. A plausible reason is that the motion prior can be more direct and effective in reducing model overfitting than counterfactual learning. Finally, we show the visualisation in Figure 3a and 3c that the original Social-STGCNN and SGCN models can predict incorrect distributions and in Figure 3b and 3d that using a motion prior can consistently enhance the predictions with matched velocity.

Strong Data Augmentation vs Motion Prior. Strong data augmentation is essential in domain generalisation to handle out-of-domain data. As suggested by Schöller et al. (2020), we augment rotated trajectories with a degree of $\{0, 1/4, 1/2, 3/4, 1\}\pi$ and their flipped scenes and the reversed trajectories. These data augmentations are applied on the scene level and therefore, the adjacency matrix remains unchanged. A single augmentation is selected randomly at each training iteration. Table 2 shows the performance with and without adding the motion prior or data augmentation on Social-STGCNN and SGCN. Note that SGCN also predicts the bi-variate Gaussian distributions and we can follow the same way to integrate the motion prior as on Social-STGCNN. As shown in Table 2 adding strong data augmentation can largely boost the performance. This suggests that data augmentation is essential in trajectory prediction when environmental differences are large - which has not been explored by Mohamed et al. (2020); Xu et al. (2022b). On the other hand, we add a motion prior on SGCN and Social-STGCNN and both can enhance the performance and obtain even better results than using data augmentation only. This implies that using a motion prior can be a more effective way to boost performance than using the data augmentation on the training set. Finally, we combine the motion prior with the data augmentation resulting in improved performance of around 6%.

4.3 Experiments on Synthetic data

Performance on Different Velocity. We first evaluate Social-STGCNN on our synthetic dataset using models trained on the UNIV dataset with a strong augmentation mentioned in Section 4.2. Table 3 shows the quantitative results of this experiment. Firstly, models without data augmentation cannot predict well in a high speed environment which supports the finding in (Schöller et al., 2020) because pedestrians in each dataset follow either horizontal or vertical motions. Then, using the augmentation helps the model generalise in different directions, but error still accumulates through the future timesteps when the speed increases, resulting in a mismatched speed when pedestrians walk at 5 m/s, approximately the maximum speed in the ETH dataset, which is also indicated in Figure 3f. By adding the motion priors, the model only accumulates slight errors in a high speed environments, which demonstrates the importance of motion prior. As shown in Figure 3g, the motion matches the speed even in high speed environment but the prediction is also biased, which may be due to the influence of the environment. Finally, a model using the Train-on-Best-Motion strategy provides a better result due to the elimination of environment bias, which can be further proven by the prediction of the straight motion in Figure 3h. All these results strongly illustrate the effectiveness and generalisation of our model in different domains due to the motion prior and our training strategy.



(a) B2E: Social-STGCNN (b) B2E: Social-STGCNN+MP (c) D2B: SGCN (d) D2B: SGCN + MP

(e) Social-STGCNN (f) Social-STGCNN (g) Social-STGCNN+MP (h) Social-STGCNN+BMP

Figure 3: Visualisation of trajectory prediction on real world and synthetic dataset with and without using motion prior (MP) and Train-on-Best-Motion (BMP). Syntheic dataset is tested using the model trained on UNIV. Blue lines are observation and red lines are ground truth. Predicted distributions are plotted using KDEPlot. Each test trajectory contain 8 time steps for observation and 12 steps for prediction.

Method	ADE_1	FDE_1	NLL	ADE_5	ADE_{10}
Base (Salzmann et al., 2020)	85.19	179.69	7.99	48.2	38.24
K0 (Ivanovic et al., 2022)	48.29	89.62	10.90	35.56	25.11
Adaptive (Ivanovic et al., 2022)	107.47	204.99	8.274	41.24	21.03
Base (Salzmann et al., 2020) + MP	**25.72**	**59.31**	**7.12**	**22.45**	**20.61**

Table 5: Results of NuScenes → Lyft Task where K0 and Adaptive are two variants of Base (Trajectron++) in (Ivanovic et al., 2022).

Data Augmentation and Pretraining. We also use our proposed synthetic dataset to pretrain the model or augment the dataset and test the model on real-world datasets. The quantitative results are shown in Table 4. Using synthetic data to train the model can have better results than using ETH&UCY only. This suggests that speed diversity is essential for human trajectory prediction. Then we continue to train on ETH&UCY datasets with and without the data augmentation described in Section 4.2 and the performance improves. However, the model can forget the pretraining when it is overfitted to the real-world training set. Therefore, we further train the model with our synthetic data augmentation and performance largely improves. Finally, the results show that simply adding a motion prior can further improve the performance.A plausible explanation is that the model is not guaranteed to learn the necessary motion rules correctly with synthetic data, but the motion prior can.

4.4 Experiments on Autonomous Driving Data

In this experiment, we perform a transfer from NuScenes to Lyft dataset to illustrate that our method is also useful in dealing with domain shift in vehicle trajectory prediction datasets. These datasets have different sampling rates (2 Hz vs 10 Hz) and we observe 2 seconds for training and predict the next 6 seconds. Ideally, an auto-regressive model such as (Salzmann et al., 2020) can effectively be adapted to inputs and outputs with different frequencies. However, Ivanovic et al. (2022) show that the model overfits the displacement change between frames and results in significant performance degradation when applied on Lyft with extremely long predicted distances. We use the same experimental settings as in (Ivanovic et al., 2022) and let Trajectron++ predict residuals on our motion prior at each step. Table 5 shows that our model has better performance than our baselines for all metrics. This indicates that our motion prior approach can also be applied to autoregressive models for dynamic lengths, effectively alleviating domain shifts caused by different sampling rates without the data from Lyft.

5 Limitations and Future Work

For the residual prediction part, we expect the model to have the capability to model both social and scene interactions. However, the scene interaction is difficult to model using only one image. Our Train-on-Best-Motion strategy alleviates this problem so that models can concentrate on social interaction only but the target domain may also contain complex terrains. Therefore, our future work will explore more advanced approaches to model scene interactions using a single image during the training phase, or pretraining techniques to train the scene interaction in a zero-shot manner. Moreover, we will investigate the improvement of the social interaction module to further overcome the over-fitting problem such as replacing the graph convolutional mechanism with an inductive learning-based graph neural network. Finally, our currently used motion prior, the constant velocity motion, is effective especially when speed bias occurs in different datasets, which is the main problem in most domain shift cases. Future works will focus on more advanced motion priors to deal with all kinds of domain shifts.

6 Conclusions

In this work, we tackle single source domain generalisation in trajectory prediction. Specifically, we add the constant velocity motion priors into trajectory prediction models and then predict the residuals towards the final predictions. Our experiments illustrate that using this simple strategy can effectively improve generalisation and performance across unseen domains. It also outperforms current domain generalisation and adaptation approaches even when only a single domain is used for training without seeing the test domain, highlighting the importance of motion priors in robust trajectory prediction. We also propose Train-on-Best-Motion strategy that uses the best motion priors during the training and effectively alleviates the domain shift due to the scene interactions.

References

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016. 1, 3

Javad Amirian, Jean-Bernard Hayet, and Julien Pettre. Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories With GANs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1, 3, 7, 13

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 3

Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Non-probability sampling network for stochastic human trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6477–6487, 2022. 13

Mohammadhossein Bahari, Ismail Nejjar, and Alexandre Alahi. Injecting Knowledge in Data-driven Vehicle Trajectory Predictors. *Transportation research part C: emerging technologies*, pp. 103010, 2021. 4

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 3

Michel Bierlaire. Discrete Choice Models. In *Operations research and decision aid methodologies in traffic and transportation management*, pp. 203–227. 1998. 3

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020. 7, 13

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. In *Conference on Robot Learning*, pp. 86–99, 2020. 3, 5, 6

Guangyi Chen, Junlong Li, Jiwen Lu, and Jie Zhou. Human Trajectory Prediction via Counterfactual Analysis. In *IEEE/CVF International Conference on Computer Vision*, pp. 9824–9833, 2021. 1, 2, 3, 4, 6, 7, 13

Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. MG-GAN: A Multi-Generator Model Preventing Out-of-Distribution Samples in Pedestrian Trajectory Prediction. In *IEEE/CVF International Conference on Computer Vision*, pp. 13158–13167, 2021. 3, 6

Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *International conference on pattern recognition*, pp. 10335–10342, 2021. 13

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Neural Information Processing Systems*, pp. 2672–2680, 2014. 3

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018. 1, 3, 4, 8

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning Priors for Efficient Online Bayesian Regression. In *International Workshop on the Algorithmic Foundations of Robotics*, pp. 318–337, 2018. 2, 7

Helbing and Molnár. Social Force Model for Pedestrian Dynamics. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, pp. 4282–4286, 1995. 1, 3, 4

Johnny L. Houston, Guido C. A. Zuidhof, Luca Bergamini, Yawei Ye, Ashesh Jain, Sammy Omari, Vladimir I. Iglovikov, and Peter Ondruska. One Thousand and One Hours: Self-driving Motion Prediction Dataset. In *Conference on Robot Learning*, 2020. 7, 13

Pingxuan Huang, Yanyan Fang, Bo Hu, Shenghua Gao, and Jing Li. CTP-Net For Cross-Domain Trajectory Prediction. *arXiv preprint arXiv:2110.11645*, 2021. 3

Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. STGAT: Modeling spatial-temporal interactions for human trajectory prediction. In *IEEE/CVF International Conference on Computer Vision*, pp. 6272–6281, 2019. 3, 4

B. Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. *2019 IEEE/CVF International Conference on Computer Vision*, pp. 2375–2384, 2019. 15

Boris Ivanovic, James Harrison, and Marco Pavone. Expanding the Deployment Envelope of Behavior Prediction via Adaptive Meta-learning. *arXiv preprint arXiv:2209.11820*, 2022. 2, 3, 6, 7, 9, 13

Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian D. Reid, Seyed Hamid Rezatofighi, and Silvio Savarese. Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks. In *Advances in Neural Information Processing Systems*, 2019. 1, 3

Parth Kothari, Brian Sifringer, and Alexandre Alahi. Interpretable Social Anchors for Human Trajectory Forecasting in Crowds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15556–15566, 2021. 3

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by Example. *Comput. Graph. Forum*, pp. 655–664, 2007. 3, 7, 13

Junwei Liang, Lu Jiang, and Alexander Hauptmann. SimAug: Learning Robust Representations from Simulation for Trajectory Prediction. In *European Conference on Computer Vision*, pp. 275–292, 2020a. 3

Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10508–10518, 2020b. 3

Yuejiang Liu, Riccardo Cadei, Jonas Schweizer, Sherwin Bahmani, and Alexandre Alahi. Towards Robust and Adaptive Motion Forecasting: A Causal Representation Perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17081–17092, 2022. 1, 3

Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction. In *European Conference on Computer Vision*, pp. 759–776, 2020. 3, 6

Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From Goals, Waypoints & Paths to Long Term Human Trajectory Forecasting. In *IEEE/CVF International Conference on Computer Vision*, pp. 15233–15242, 2021. 3

Huynh Trung Manh and Gita Alaghband. Scene-LSTM: A Model for Human Trajectory Prediction. *ArXiv*, abs/1808.04018, 2018. 6

Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14424–14432, 2020. 1, 2, 3, 4, 5, 6, 7, 8, 13

Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many Observations are Enough? Knowledge Distillation for Trajectory Forecasting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6553–6562, 2022. 4

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 *European Conference on Computer Vision*, pp. 282–298, 2020. 3

Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings. In *European Conference on Computer Vision*, pp. 452–465, 2010. 7, 13

Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. CoverNet: Multimodal Behavior Prediction Using Trajectory Sets. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14062–14071, 2020. 3

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019. 1, 3, 4, 6

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, pp. 683–700, 2020. 7, 9

Christoph Schöller, Vincent Aravantinos, Florian Samuel Lay, and Alois Knoll. What the Constant Velocity Model Can Teach Us About Pedestrian Motion Prediction. *IEEE Robotics and Automation Letters*, pp. 1696–1703, 2020. 2, 5, 8

Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sparse Graph Convolution Network for Pedestrian Trajectory Prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8994–9003, 2021. 1, 3, 4, 5, 6, 7, 8, 13

Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive Social Behavior Graph for Trajectory Prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 660–669, 2020. 3, 6, 8

Tessa van der Heiden, Naveen Shankar Nagaraja, Christian Weiss, and Efstratios Gavves. SafeCritic: Collision-Aware Trajectory Prediction. *ArXiv*, 2019. 3

Anirudh Vemula, Katharina Muelling, and Jean Oh. Social Attention: Modeling Attention in Human Crowds. In *IEEE International Conference on Robotics and Automation)*, pp. 1–7, 2018. 1, 3

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. GroupNet: Multiscale Hypergraph Neural Networks for Trajectory Prediction With Relational Reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6498–6507, 2022a. 3

Yi Xu, Dongchun Ren, Mingxia Li, Yuehai Chen, Mingyu Fan, and Huaxia Xia. Tra2Tra: Trajectory-to-Trajectory Prediction With a Global Social Spatial-Temporal Attentive Neural Network. *IEEE Robotics and Automation Letters*, pp. 1574–1581, 2021. 6

Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive Trajectory Prediction via Transferable GNN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6520–6531, 2022b. 1, 2, 3, 5, 6, 7, 8, 13

Hao Xue, Du Q. Huynh, and Mark Reynolds. SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194, 2018. 6

Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who Are You With and Where Are You Going? *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1345–1352, 2011. 3

Fangkai Yang and Christopher Peters. Social-aware navigation in crowds with static and dynamic groups. In *International Conference on Virtual Worlds and Games for Serious Applications*, pp. 1–4, 2019. 3

He Zhao and Richard P. Wildes. Where Are You Heading? Dynamic Trajectory Prediction With Expert Goal Examples. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7629–7638, 2021. 3, 8

	ETH	HOTEL	UNIV	ZARA1	ZARA2
No. of frame sequences	70	301	947	602	921
No. of Pedestrians	181	1053	24334	2253	5833
No. of pedestrians per sequence	2.586	3.498	25.696	3.743	6.333
Average Velocity. (m/s)	2.37	1.15	0.73	1.15	1.12
Min Velocity. (m/s)	0.00	0.00	0.05	0.08	0.07
Max Velocity. (m/s)	5.93	2.22	1.98	1.95	2.14

Table 6: Statistics of five different scenes, ETH, HOTEL, UNIV, ZARA1, and ZARA2. The velocity is calculated using the instance velocity during each 0.4 seconds.

A Appendix: Dataset Description

ETH&UCY We use two real-world datasets: the ETH (Lerner et al., 2007) & UCY (Pellegrini et al., 2010) to evaluate our methods. These datasets contain 5 different scenes: ETH, HOTEL, UNIV, ZARA1, ZARA2 and each provides a bird's-eye view of video and a set of 2D coordinates in a real-world domain to indicate the trajectories. Table 6 shows the statistics of these five datasets. It is worth noting that the speed differences among these datasets are large. For example, ETH contains higher speed motions while Hotel contains lower speed motions. Note that in the ETH dataset, the trajectories are accelerated as mentioned by Giuliari et al. (2021) and we consider it as an extreme case in our baselines. Moreover, ETH and HOTEL only contain 70 and 301 frame sequences respectively, which can result in strong overfitting when training only on these two datasets. More detailed information for these datasets can be seen in (Xu et al., 2022b).

Synthetic Dataset We create our synthetic dataset by modifying the code in SocialWays[1] (Amirian et al., 2019). Therefore, we are able to generate K bi-directional trajectories corresponding to $2K$ trajectories in total for each speed requirement, where K is a customisable hyper-parameter. In our experiment, we set $K = 30$ and the overview of our synthetic data is shown in Fig. 4. This dataset may not be very realistic in the real world, but can provide a basic indication of how models work in a trivial environment. We further analyse the model behaviour using this dataset. Since our baselines (Mohamed et al., 2020; Shi et al., 2021) require at least two pedestrians in a scene for each iteration during the training or testing phase, we combine any two trajectories as a batch with 100 meters away from each other, assuming that they do not affect each other at such far distances.

NuScenes $\rightarrow$ Lyft NuScenes (Caesar et al., 2020) and Lyft (Houston et al., 2020) are two large well-known autonomous driving datasets. NuSecenes contains 1000 scenes collected in Boston and Singapore with data sampled at 2 Hz. Lyft (Houston et al., 2020) contains 170K scenes collected in Palo Alto with data sampled at 10 Hz. To fairly compare with models in (Ivanovic et al., 2022), we use the official *trainval* set in NuScenes to train the model and evaluate on the *sample* set of Lyft in our experiment. As mentioned by Ivanovic et al. (2022), domain shift occurs on the moving ranges between two frames in these two datasets due to the varying sampling frequencies. More detailed information for this experiment can be seen in (Ivanovic et al., 2022).

B Appendix: Experimental Configurations

We use the experimental environment in Social-STGCNN[2] and implement the SGCN baseline using the code in NPSN[3] (Bae et al., 2022). For CF-STGCNN, we follow the implementation in (Chen et al., 2021) and replace all input nodes as zeros. We use one layer of spatial-temporal graph convolutional neural network (STGCNN) and 5 layers of time-extrapolator convolutional neural network (TXP-CNN). In addition, during inference, we select the mean values at all future time-steps as one of the sampled trajectories and randomly sample other trajectories in all experiments for models including Social-STGCNN, SGCN and CF-STGCNN.

[1] https://github.com/crowdbotp/socialways/blob/master/create_toy.py
[2] https://github.com/abduallahmohamed/Social-STGCNN
[3] https://github.com/InhwanBae/NPSN/blob/main/baselines/sgcn/

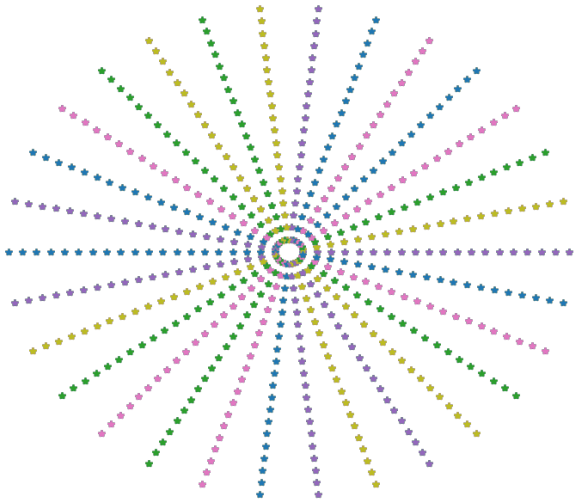


Figure 4: An overview of our synthetic data containing constant velocity motions heading 30 directions, with two trajectories towards and away from the center. Therefore, 60 different trajectories are used to evaluate the model.

Method	Performance (ADE_{20}) (Source2Target)																				Avg
	A2B	A2C	A2D	A2E	B2A	B2C	B2D	B2E	C2A	C2B	C2D	C2E	D2A	D2B	D2C	D2E	E2A	E2B	E2C	E2D	
Social-STGCNN (original)	1.29	1.21	1.74	1.34	3.46	1.20	2.32	1.47	**0.47**	0.59	0.59	0.98	0.73	0.87	0.54	0.40	0.82	1.01	0.60	0.48	1.11
Social-STGCNN (ours)	1.06	0.97	1.18	1.03	2.50	0.97	2.06	1.16	0.75	0.45	0.41	0.34	**0.66**	0.81	0.54	0.30	0.79	0.97	0.57	0.36	0.89
Social-STGCNN + MP	0.65	0.69	0.69	0.66	0.85	0.41	0.43	0.32	0.72	0.29	0.32	0.26	0.78	**0.23**	0.39	0.27	0.82	0.32	0.40	0.32	0.49
Social-STGCNN + MP+Aug	**0.28**	**0.37**	**0.40**	**0.31**	0.78	0.38	**0.31**	**0.25**	0.72	0.24	**0.30**	**0.24**	0.87	0.25	**0.38**	**0.27**	0.81	0.24	**0.37**	**0.29**	**0.40**
Social-STGCNN + BMP	0.41	0.45	0.49	0.43	0.86	0.42	0.41	0.35	0.72	**0.23**	0.32	0.26	0.78	0.24	0.39	0.28	0.79	0.22	0.38	0.32	0.44
Social-STGCNN + BMP + Aug	0.41	0.45	0.49	0.43	**0.72**	**0.23**	0.33	0.26	0.72	**0.23**	0.33	0.26	0.77	0.24	0.39	0.28	**0.77**	**0.22**	0.38	0.32	0.41

Method	Performance (FDE_{20}) (Source2Target)																				Avg
	A2B	A2C	A2D	A2E	B2A	B2C	B2D	B2E	C2A	C2B	C2D	C2E	D2A	D2B	D2C	D2E	E2A	E2B	E2C	E2D	
Social-STGCNN (original)	1.49	1.31	2.30	1.50	6.94	2.23	4.66	2.77	**0.73**	1.03	0.90	1.72	1.24	1.52	0.79	0.54	**1.43**	1.74	0.97	0.76	1.83
Social-STGCNN (ours)	1.14	1.08	1.66	1.28	4.69	1.82	4.26	2.35	1.27	0.80	0.69	0.57	**1.20**	1.49	0.88	**0.42**	1.70	1.74	0.94	**0.52**	1.53
Social-STGCNN + MP	0.76	0.89	0.85	0.78	1.66	0.68	0.67	0.54	1.49	0.48	**0.53**	0.45	1.66	0.35	0.67	0.49	1.75	0.53	**0.67**	**0.52**	0.82
Social-STGCNN + MP + Aug	**0.41**	0.58	0.60	0.53	1.76	0.77	0.57	0.48	1.45	0.37	0.54	**0.44**	1.87	0.39	0.74	0.49	1.74	0.39	0.74	**0.52**	0.77
Social-STGCNN + BMP	**0.41**	**0.52**	**0.56**	**0.50**	1.71	0.60	**0.53**	0.46	1.48	0.32	**0.53**	**0.44**	1.65	**0.34**	0.66	0.47	1.66	0.32	**0.67**	0.53	0.72
Social-STGCNN+ BMP + Aug	**0.41**	**0.52**	**0.56**	**0.50**	**1.49**	**0.31**	0.54	**0.44**	1.49	**0.31**	0.54	**0.44**	1.65	**0.34**	**0.66**	0.47	1.63	**0.31**	**0.67**	0.53	0.69

Table 7: Full ADE ↓ (top) and FDE ↓ (bottom) results of our ablation study across different training and testing sets. The first two rows compare the Social-STGCNN with original and our TXP-CNN decoders while the second and third two rows are results of our models with the motion prior before and after adding data augmentation. "2" represents from source domain to target domain. A, B, C, D, and E denote ETH, HOTEL, UNIV, ZARA1, and ZARA2, respectively. "-Aug" denotes the model trained with data augmentation, "-MP" denotes the method with the motion prior and "-BMP" denotes the model with motion prior and trained using Train-on-Best-Motion.

C Appendix: Results on Pedestrian Dataset

Effectiveness of Our TXP-CNN. Our TXP-CNN is different from the original TXP-CNN in Social-STGCNN. The results in Table 7 shows that our new TXP-CNN can lead to a higher performance on most scenes, which suggests that integrating neighbours' graph embeddings during the decoding is not necessary.

Motion Prior with Data Augmentation. We extend our experiment by adding the data augmentation on our models with motion priors. As shown in Table 7, adding the data augmentation can consistently improve the performance, especially on the Hotel dataset. This suggests that data augmentation is essential in single source domain adaptation/generalisation tasks, especially when the dataset is not sufficient or representative enough to train the model.

	ADE (std)	FDE (std)	KDE-NLL
Social-STGCNN	0.89 (7.12e-4)	1.53 (2.32e-3)	5.28
Social-STGCNN-MP	0.49 (7.06e-4)	0.82 (1.84e-3)	3.16
Social-STGCNN-MP-MA	**0.43** (4.68e-4)	**0.71** (2.18e-3)	**2.68**

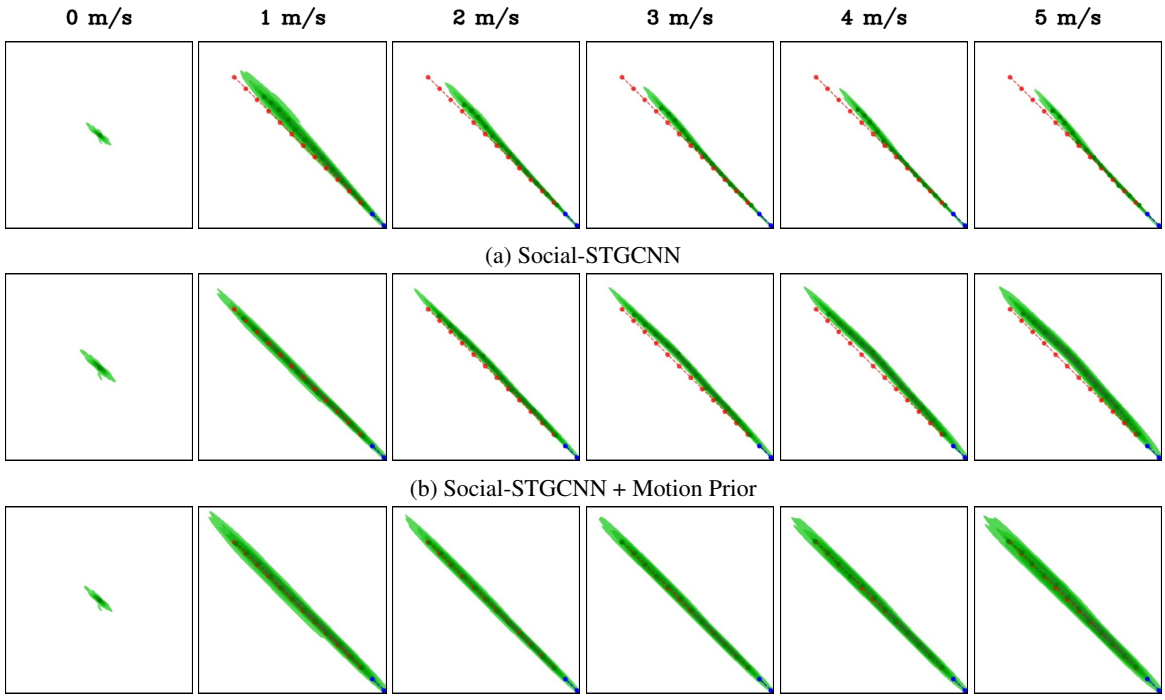
Table 8: Average ADE and FED results (including standard deviation) and KDE-NLL results on 20 tasks on ETH&UCY datasets for Social-STGCNN with our motion prior. Note that KDE-NLL does not require standard deviation due to large number of sampling.

Standard Deviation and KDE-NLL Results We also show the average ADE and FDE results with standard deviation from 10 runs and KDE-NLL (Ivanovic & Pavone, 2019) results on 1000 samples of our models in Table 8. Apparently, using motion prior and our Train-on-Best-Motion strategy can stably improve performance across different environments.

D Appendix: Results on Synthetic Dataset

Quantitative Results. We provide full quantitative results in Table 9 using our synthetic data. It is clear to see that the model using our motion prior can consistently predict trajectories with matched speed, which strongly illustrates the effectiveness of using a motion prior across datasets with environments of different speeds. Using the Train-on-Best-Model strategy can obtain better results and we will further illustrate it in the next section. We also notice that the model trained on UNIV shows a relatively larger error than HOTEL, ZARA1 and ZARA2 in this experiment, and we believe this is due to the speed constraint in such high density environment and many turning scenarios when entering the gate. Then, models trained on ZARA1 and ZARA2 have a similar performance to each other because these two datasets are quite similar to each other. Finally, we find that training on the ETH dataset does not provide better results under high speed scenes even when we augment the dataset, which may be due to the non-linear walking paths due to terrain constraints.

Qualitative Results. We show the full qualitative results in Figure 5 using the Social-STGCNN trained on the UNIV dataset. We can firstly see that the errors still occur for all models when pedestrians are static, which indicates that the biases in the model cause a level of uncertainty. Then, at the speed of 1 m/s, which is approximately the average speed in UNIV dataset, the predicted distribution indicates that the model can predict a matched speed. However, the speed errors are progressively increased when Social-STGCNN is applied in a higher speed spaces, which illustrates why these models cannot do well on the ETH dataset, where speeds of most trajectories are around 2-5 m/s or even higher than 5 m/s. Using the motion prior can predict trajectories with matched speed as shown in Figure 5b, but the visualisations also show that the model can generate non-straight predictions at the speed from 2 m/s to 5 m/s and the offset can be the incorrect prior implicit in the datasets, which may be due to the environmental issue. Figure 5c shows that using Train-on-Best-Motion strategy can alleviate this effect and can predict a straight future path following the observed trajectory, which largely boosts the performance.



(a) Social-STGCNN

(b) Social-STGCNN + Motion Prior

(c) Social-STGCNN + Motion Prior + "Training-on-Best-Motion"

Figure 5: The full visualisations of predicted trajectories on our synthetic data using our models. All models are on trained on UNIV dataset with strong data augmentations. From left to right are predictions with speed of $\{0, 1, 2, 3, 4, 5\}$ m/s. Blue dots are observed positions, red dots are ground truth future trajectories and green regions are predicted distributions using KDE plot. Each test trajectory contains 8 time steps of observation and 12 steps for prediction. Using a motion prior can eliminate the accumulated errors during the training time across the future timesteps and Train-on-Best-Motion can better eliminate the environmental bias.

Model	Src	Full Performance (ADE_{20}/FDE_{20})						Avg
		0 m/s	1 m/s	2 m/s	3 m/s	4 m/s	5 m/s	
Social-STGCNN	A	0.60/0.46	4.71/8.90	10.89/20.43	17.25/32.21	23.65/44.02	30.10/56.08	11.89/22.25
	B	0.19/0.23	4.04/7.82	9.20/17.41	14.52/27.26	19.86/37.10	25.24/47.03	
	C	0.19/0.16	3.25/6.31	7.54/14.40	11.84/22.42	16.22/30.56	20.55/38.63	
	D	0.17/0.16	3.84/7.30	8.46/15.88	13.17/24.52	17.87/33.13	22.57/41.73	
	E	0.19/0.16	4.17/7.98	9.07/17.27	14.08/26.59	19.12/35.99	24.13/45.37	
Causal-STGCNN	A	0.68/0.68	2.58/4.61	5.67/10.91	8.90/16.73	11.53/20.74	14.98/25.43	7.16/12.87
	B	0.10/0.14	2.16/3.94	5.50/10.38	8.71/16.22	11.35/20.37	14.15/24.56	
	C	1.72/2.30	1.77/2.98	4.60/8.57	7.82/14.70	10.74/19.63	13.68/25.04	
	D	0.94/1.05	2.32/4.06	5.63/10.26	9.03/16.41	12.07/21.33	15.08/25.85	
	E	0.87/1.39	2.44/4.33	5.47/10.14	8.55/16.12	11.38/21.03	14.36/26.10	
SGCN	A	0.19/0.21	3.44/6.46	7.69/14.37	12.07/22.54	16.49/30.80	20.99/39.18	7.57/14.46
	B	0.07/0.11	3.81/7.13	8.35/15.55	12.90/24.02	17.47/32.46	22.05/40.91	
	C	0.05/0.05	0.65/1.46	3.10/6.44	5.83/11.73	8.59/17.03	11.34/22.32	
	D	0.12/0.10	1.45/2.83	3.85/7.51	6.41/12.43	8.98/17.37	11.57/22.32	
	E	**0.02/0.03**	1.58/3.12	4.59/9.18	7.87/15.60	11.17/22.03	14.45/28.44	
Social-STGCNN-Aug	A	0.54/0.49	2.90/4.97	7.61/13.73	12.60/22.65	17.41/31.28	22.27/39.92	3.95/6.81
	B	**0.10/0.09**	1.06/1.57	2.38/2.97	4.51/5.63	6.85/9.61	9.17/13.51	
	C	0.11/0.11	0.62/1.23	1.60/3.41	2.60/5.57	3.63/7.72	4.62/9.78	
	D	0.31/0.41	0.55/0.79	1.12/1.86	1.75/3.05	2.37/4.25	3.00/5.46	
	E	0.04/0.04	0.45/0.66	1.07/1.72	1.73/2.87	2.40/3.98	3.05/5.00	
Social-STGCNN-MP-Aug	A	0.22/0.30	0.44/0.45	0.82/1.40	1.50/2.97	2.36/4.68	3.28/6.38	0.92/1.43
	B	0.12/0.12	**0.08/0.10**	0.18/**0.11**	0.28/**0.13**	0.39/**0.16**	0.50/**0.20**	
	C	**0.01**/0.04	0.31/0.52	0.75/1.29	1.28/1.92	1.95/2.84	2.73/4.24	
	D	0.06/0.11	0.22/0.34	0.47/0.67	0.72/0.93	0.97/1.05	1.44/1.49	
	E	0.02/0.06	0.33/0.60	0.71/1.18	1.25/1.84	1.89/2.81	2.44/3.89	
Social-STGCNN-BMP-Aug	A	0.13/0.13	0.49/0.47	0.87/1.36	1.49/2.73	2.28/4.28	3.16/5.85	0.75/1.22
	B	0.15/0.14	0.31/0.25	0.50/0.49	0.78/0.96	1.19/1.56	1.67/2.21	
	C	0.02/**0.02**	**0.16/0.19**	**0.42/0.46**	0.83/**1.04**	1.36/**2.26**	1.88/**3.57**	
	D	**0.05/0.04**	0.12/0.12	0.24/0.26	0.37/0.51	0.51/0.79	0.65/1.13	
	E	0.03/0.06	**0.11/0.12**	0.34/0.49	0.60/1.15	0.87/1.74	1.17/2.34	
SGCN-Aug	A	**0.05/0.09**	3.20/6.08	7.55/14.20	11.95/22.44	16.37/30.67	20.92/39.11	5.02/ 9.55
	B	0.11/0.10	0.46/0.77	1.73/3.32	3.47/6.66	5.36/10.23	7.30/13.88	
	C	0.02/0.04	1.03/2.09	3.97/7.76	7.19/13.84	10.45/19.95	13.69/26.05	
	D	0.11/0.10	0.16/0.27	1.41/2.65	2.95/5.42	4.52/8.21	6.10/10.99	
	E	0.03/0.05	0.31/0.61	2.10/4.33	4.08/8.28	6.08/12.19	8.04/16.09	
SGCN-MP-Aug	A	0.34/0.45	**0.37/0.44**	**0.44/0.44**	**0.53/0.49**	0.63/0.61	0.68/0.73	0.48/1.20
	B	0.08/0.09	0.10/0.13	0.22/0.57	0.45/ 1.53	0.82/ 2.66	1.28/ 3.89	
	C	0.04/0.06	0.23/0.44	0.46/1.40	**0.79**/ 2.57	**1.15**/ 3.74	**1.53**/ 4.83	
	D	0.03/0.06	0.10/0.12	0.14/0.26	**0.20/0.48**	**0.28/0.71**	**0.36/0.94**	
	E	0.01/0.03	0.22/0.48	0.42/1.06	0.62/ 1.68	0.83/ 2.30	1.04/ 2.91	
SGCN-BMP-Aug	A	0.74/0.64	0.72/0.63	0.68/0.61	0.63/0.59	**0.59/0.54**	**0.56/0.51**	**0.42/0.72**
	B	**0.03/0.05**	0.10/0.11	**0.15**/0.19	**0.19**/0.33	**0.26**/0.50	**0.33**/0.68	
	C	0.02/0.03	0.38/0.88	0.73/ 1.74	1.08/ 2.53	1.38/ 3.24	1.68/ 3.91	
	D	0.10/0.09	**0.08/0.09**	**0.12/0.24**	0.21/0.49	0.32/0.77	0.44/ 1.03	
	E	**0.02/0.03**	0.13/0.12	**0.18/0.18**	**0.22/0.21**	**0.25/0.28**	**0.28/0.36**	

Table 9: Full ADE/FDE scores against different training sets and different speed controls using our synthetic dataset. A, B, C, D, and E denote the model trained on ETH, HOTEL, UNIV, ZARA1, and ZARA2 respectively. "Src" indicates which training set is used. "-Aug" means data augmentation is used during training. "-MP" and "-BMP" denote the model using the motion prior and trained on Train-on-Best-Motion respectively.