Human Trajectory Prediction via Counterfactual Analysis

Guangyi Chen1,2,3, Junlong Li1,2,3, Jiwen Lu1,2,3,* Jie Zhou1,2,3
1Department of Automation, Tsinghua University, China
2State Key Lab of Intelligent Technologies and Systems, China
3Beijing National Research Center for Information Science and Technology, China
chen-gyl6@mails.tsinghua.edu.cn; ljlong.leo@gmail.com; {luwen, jzhou}@tsinghua.edu.cn

Abstract

Forecasting human trajectories in complex dynamic environments plays a critical role in autonomous vehicles and intelligent robots. Most existing methods learn to predict future trajectories by behavior clues from history trajectories and interaction clues from environments. However, the inherent bias between training and deployment environments is ignored. Hence, we propose a counterfactual analysis method for human trajectory prediction to investigate the causality between the predicted trajectories and input clues and alleviate the negative effects brought by the environment bias. We first build a causal graph for trajectory forecasting with history trajectory, future trajectory, and the environment interactions. Then, we cut off the inference from environment to trajectory by constructing the counterfactual intervention on the trajectory itself. Finally, we compare the factual and counterfactual trajectory clues to alleviate the effects of environment bias and highlight the trajectory clues. Our counterfactual analysis is a plug-and-play module that can be applied to any baseline prediction methods including RNN- and CNN-based ones. We show that our method achieves consistent improvement for different baselines and obtains the state-of-the-art results on public pedestrian trajectory forecasting benchmarks.1

1. Introduction

Human trajectory prediction aims at forecasting the future trajectory of pedestrians based on their past positions in complex and crowd environments. It is a critical and fundamental task for many applications, including the planning and controlling of the autonomous vehicles, the robot navigation, and the tracking and re-identification in the crowd surveillance [39, 7, 8, 3, 9]. Thanks to these significances, human trajectory prediction task has attracted much attention over the past few years [1, 53, 56, 15, 25].

Despite the recent progress, trajectory prediction is still a challenging problem due to complex social or physical environment interactions. In crowded environment, the pedestrians are always interacted with the social behavior from other pedestrians and the common scene context. For example, pedestrians may walk in parallel for social talking or stop for seconds to avoid collisions. Besides, the behaviors of pedestrians may be interacted by the traffic light, crosswalk, or just an obstacle like a tree.

Most existing methods concentrate on modeling the en-
vironment interactions and aggregate these interaction clues with history behavior clues for trajectory prediction. For example, Social LSTM [1] applies a social pooling module to extract clues from environment interaction. While Social-STGCNN [29] utilizes the spatial-temporal graph CNN to encode the environment interactions. However, these methods always ignore the inherent bias between training and deployment environments. As shown in part (a) of Figure 1, we statistically compare the interactions on training and testing environments of the UNIV scene. We observe an obvious gap between the interactions of different environments. We also visualize the environment difference in part (b) of Figure 1, where the training environment is a street scene while the testing environment is a more crowded public square. This environment bias causes heavy overfitting of complex environment interaction modules in most of trajectory prediction methods. When the training data contains many examples who turns left in the crossroads, the prediction of left trajectory will be wrongly attributed to crossroads, which misleads the prediction in the new environment where pedestrians turn right in the crossroads. Besides, deployment environments are always unpredictable in the applications of autonomous vehicles and intelligent robots. Thus, it is difficult to apply transfer learning methods to reduce environment bias.

To address this problem, in this paper, we propose a counterfactual analysis method to alleviate the overdependence of environment bias and highlight the trajectory clues itself. Inspired by the causal inference methods [37, 36, 12], we propose to use the counterfactual intervention to investigate the causality between the observed clues and predicted trajectories. Different from conventional causal inference methods [34, 10], we furthermore extend these to the training process for the optimization of the prediction model. Specifically, we first construct a factual causal graph by the human prior knowledge, whose nodes include past trajectory, environment interaction, and future trajectory. As shown in the left part of Figure 1 (b), environment interaction may be a negative confounder due to the bias between training and deployment environments. Then, we conduct the counterfactual intervention on the history trajectory, which cuts off the dependence between the environment and trajectory. Motivated by [48, 45], we replace the history trajectory feature with the counterfactual one, such as uniform rectilinear motion, mean trajectory, or random trajectory. This counterfactual prediction indicates the effect of biased environment clues. Finally, we subtract the counterfactual prediction from original prediction as the causality-aware prediction since the negative effect of confounder is alleviated. We highlight that the proposed counterfactual analysis method is a plug-and-play module which can be applied to any baseline prediction method including RNN- and CNN-based ones. In the experiments, we apply our method for two baseline methods including RNN-based STGAT [18] and CNN-based Social-STGCNN [29]. We show that our counterfactual analysis method achieves consistent improvement for both two baseline methods and obtains the state-of-the-art results on public pedestrian trajectory prediction datasets. We summarize three advantages of our method as follows:

- We show that the environment bias may cause the heavy overfitting of the complex environment interaction modules.
- We propose a counterfactual analysis method to alleviate the overdependence of environment bias and highlight the trajectory clues itself.
- Our counterfactual analysis is a simply plug-and-play module which can be easily applied to any baseline predictor, and consistently improves the performance on many human trajectory prediction benchmarks.

2. Related Work

**Environment Interaction:** The environment interaction consists of social environment interaction and physical environment interaction. To model the complex social environment interaction, previous methods always model human crowd motion with handcrafted rules or energy parameters, e.g. Social Force Model [17], continuum dynamics [47], Discrete Choice framework [2], Gaussian processes [46, 51] and crowd analysis [40, 54, 23]. In deep learning methods, early studies [1, 15] employ the pooling module to capture social interactions. While some methods [49, 11, 41, 55, 6] apply the attention model to distinguish the importance of different neighbors. Many recent methods employ the graph model to extract the clues from interactions, e.g. the States Refinement module for message passing in SR-LSTM [56]; the Graph Attention model for modeling interactions in [18, 21]; STR-GGRNN [16] uses data-driven adaptive online neighborhood recommendation. In addition, some methods [22, 41, 53, 21, 4] also incorporate physical environment interactions of the scene context for more reliable prediction. However, most of these methods ignore the environment bias between training and deployment environments, which causes heavy overfitting of the environment interaction modules. To address this, we propose a counterfactual analysis method to focus more on trajectory itself, instead of the biased interaction.

**Predictor:** Deep learning based methods always regard the trajectory forecasting as the sequence prediction problem and apply the RNN model [1, 15, 21, 56, 41] or temporal CNN model [31, 29] as the predictor. For example, Social LSTM [1] employs the LSTM to encode human motion, then fuses the environment interaction clues extracted by a pooling module and feeds them into a LSTM
decoder to sequentially predict the trajectory, while Nikhil et al. [31] first proposed to use the temporal CNN model to replace the RNN-based predictors. Besides, [13] proposes to use Transformers for predicting the future trajectories. However, both RNN-based and CNN-based network are likelihood-based predictor whose causalities between observed clues and predicted results are still non-transparent. In our counterfactual analysis, we can investigate the effects of different clues including history trajectory and environment interactions by the intervention.

**Generative Modeling:** Another popular research direction of human trajectory prediction is exploring the indeterminacy of the human behavior. Many methods [15, 19, 21, 57, 18, 43] employ generative model to predict multiple plausible paths, instead of the deterministic one. These methods employ Generative Adversarial Network (GAN) [14] to implicitly incorporates a latent variable into the encoded embedding [15, 21, 41, 57], or Variational Auto-Encoder (VAE) [20] to explicitly model the multimodal distribution with a latent variable [22, 19, 42, 28]. While STGCNN [29] directly predicts a bi-variate Gaussian distribution to replace some deterministic trajectories with multi-modal predictions.

**Causal Inference:** Different from conventional associated inference, the causal inference aims to investigate the effect of variables when some cause is changed [36, 35]. Recently, many methods take effort to apply causal inference for the effect analysis of deep neural network in different fields including reinforcement learning [30, 12], natural language processing [50, 32], and visual representation learning [5, 27, 52, 45]. The clear causalities can improve the transparency of the deep models as most of them are treated as black-box nowadays. In this paper, we propose to apply the counterfactual analysis to the trajectory prediction system to analyze the effects of different clues and enhance the reliability, which attempts to explain how neural networks make the prediction and optimize the prediction by removing the affects from biased dataset. It is helpful to understand the causal relation between the input clues and output trajectory predictions.

### 3. Approach

#### 3.1. Problem Definition

Pedestrian trajectory prediction task can be defined as a sequential prediction problem, whose inputs include the history trajectory and environment interaction. Many method designs complex model to learn the clues from environment interactions. However, the inherent bias between training and deployment environments may cause the overfitting of interaction model. In the conventional trajectory prediction framework, given N pedestrians in a scene, we can define the history trajectory of the ith pedestrian as $X_i = \{(x_{i,t}^t, y_{i,t}^t) \in R^2| t = 1, 2, ..., t_{obs}\}$, where the $(x_{i,t}^t, y_{i,t}^t)$ is the 2D location at time t. The ground-truth future trajectory of the ith pedestrian can be defined as $Y_i = \{(x_{i,t}^t, y_{i,t}^t) \in R^2| t = t_{obs} + 1, t_{obs} + 2, ..., t_{pred}\}$. For the normalization of scale, many methods employ the relative locations even the relative speeds to replace the absolute locations. The environment interactions can be extracted from the trajectories including other pedestrians and the physical scene context as $E_i = \mathcal{E}_\phi(\{X_i\}_{i=1}^N, C)$, where $C$ denotes the physical scene context. The sequential trajectory prediction process is modeled as

$$\hat{Y}_i = F_\theta(X_i, E_i), \quad (1)$$

where $\hat{Y}_i = \{(\hat{x}_{i,t}^t, \hat{y}_{i,t}^t) \in R^2| t = t_{obs} + 1, t_{obs} + 2, ..., t_{pred}\}$ denotes the predicted trajectory.
Given a set of training pedestrian trajectory data \( \{(X_i, Y_i)\} \in \Omega \), the predictor is optimized by the L2 loss:

\[
L_{tra}(\theta, \phi; \Omega) = L_{L2}(Y_i, \hat{Y}_i),
\]

where \( \theta \) and \( \phi \) denote the parameters of the pedestrian trajectory prediction system.

### 3.2. Causal Graph

In this subsection, we reformulate above trajectory prediction framework with the causal graph \( G = \{V, E\} \) with the prior knowledge. As shown in part (c) of Figure 1, the nodes \( V \) in the graph denote the variables including history trajectory \( X \), environment interactions \( E \) (including social and physical environment interactions), and future trajectory \( Y \). While the causal links \( E \) indicate the hidden causal relations and how these variables interact with each other. In a link \( E \rightarrow X \rightarrow Y \), we can call the node \( E \) as the parent node of \( X \), while \( Y \) is the child node of \( X \). The link from the parent node to the child node denotes a causality. \( E \rightarrow (X, Y) \) indicates all the human trajectories are influenced by the environment. In the causal inference theory [33], when a variable \( E \) simultaneously affects two variables, it becomes a confounder in the causal analysis of these two variables. For example, if turning left is often observed with the environment interaction of crossroads, the prediction of left trajectory will be wrongly attributed to crossroads. As a result, the real causality underlying the history trajectory may be ignored.

### 3.3. Counterfactual Analysis

For conventional likelihood-based predictors, the causalities between different observed clues and predicted results are non-transparent. The prediction model is easy to be “cheated” by the short cut between the biased environment interaction and final future trajectory. Inspired by the causal inference methods [33, 34] which attempt to analyze the causalities among different clues, we propose to apply the counterfactual analysis to trajectory prediction system to mitigate the negative effect of biased environment and encourage the model to focus more on trajectory analysis. Counterfactual intervention means to imagine a non-existent observation to replace the original factual clues in the trajectory prediction system. In the causal inference methods [37], the intervention is formulated as \( do(\cdot) \). Once one variable is intervened, its all in-coming links in the causal graph are cut off and its value is independently given, while other variables who are not affected still maintain the original value. In our method, as shown in part (c) of Figure 1, we replace the history trajectory features in the system with the imagined ones. We can obtain the prediction results under this intervention as:

\[
\hat{Y}_{X_i=x'} = F_\theta(do(X_i = x'), E_i),
\]

where \( X = x' \) indicates the counterfactual value. Specifically, we can apply the different counterfactual interventions such as the zero vector (which denotes uniform rectilinear motion when the relative speed is used as input), the mean vector of all history trajectories or the random trajectories as the counterfactual intervention.

The original factual prediction \( \hat{Y}_i \) is dependent on both trajectories and environment interactions, while the counterfactual predicted result \( \hat{Y}_{X_i=x'} \) is only dependent with environment interactions since the trajectory is replaced by counterfactual intervention. To investigate the real effect of the trajectory itself, we compute the difference between the counterfactual prediction and factual prediction and name it as the causal prediction:

\[
\hat{Y}_{causal} = \hat{Y}_i - \hat{Y}_{X_i=x'}.\]

Compared with the original likelihood prediction, the causal prediction is more reliable by avoiding the biased effects from environment confounders. In the training process, we optimize the network to predict the causal prediction:

\[
L_{Causal}(\theta, \phi; \Omega) = L_{L2}(Y_i, \hat{Y}_{causal}).
\]

### 3.4. Generative Model

It is worth noting that our counterfactual analysis is robust to the slight change of trajectory prediction systems, such as different predictors, whether the physical environment content is used, or whether the generative model is used. These changes cannot affect the structure of the causal graph. When using the generative methods, we will add the noise latent variable in the trajectory prediction system and predict the stochastic future trajectory. Taking the GAN as the example of generative model, the loss function in (2) is reformulated as

\[
L_{GAN}(\theta, \phi, \psi; \Omega) = L_{L2}(Y_i, \hat{Y}_i) + \log D_\psi(Y_i) + \log(1 - D_\psi(\hat{Y}_i)),
\]

where \( \hat{Y}_i = F_\theta(X_i, E_i, Z) \) indicates the trajectory is generated with a noise latent variable \( Z \) for multi-modal distributions. While \( D_\psi \) is the discriminator in the GAN to judge whether the trajectory is generated with the noise latent variable. Our counterfactual analysis can work together with generative methods, which only needs to add the noise latent variable \( Z \) in the side of predicted trajectory \( Y \) with \( Z \rightarrow Y \). The causal prediction with the latent variable is formulated as:

\[
\hat{Y}_{causal} = \hat{Y}_i(X_i, E_i, Z) - \hat{Y}_{X_i=x'}(do(X_i = x'), E_i, Z).
\]
Then, we only need to replace the original prediction with causal prediction as:
\[ L_{\text{CausalGAN}}(\theta, \phi, \psi | \Omega) = L_{L2}(Y_i, \hat{Y}_{\text{causal}}) + \log \mathcal{D}_\psi(Y_i) + \log(1 - \mathcal{D}_\psi(\hat{Y}_{\text{causal}})). \] (8)

3.5. Implementation Details

As shown in Figure 2, the framework of trajectory prediction system always contains 4 modules including history path encoder, scene perceptor, interactions analysis module, and trajectory predictor. Instead of the conventional likelihood-based trajectory predictor, in this paper, we conduct the counterfactual intervention for some clues and apply a causal predictor by computing the difference between the prediction results with factual and counterfactual clues. Our causal prediction is simple yet effective for the reliability of trajectory prediction system. To evaluate the generality of our counterfactual analysis method, we incorporate it as a plug-and-play module into two different baseline methods including RNN-based STGAT [18] and CNN-based Social-STGCNN [29]. We simply introduce these two implementations in the following. Note that though the whole counterfactual analysis method contains the features of environment scene, here we do not use the scene features in the implementations since our baseline models do not use scene images.

Causal-STGAT: The STGAT method employs two LSTM (M-LSTM motion encoding and G-LSTM for graph) and a graph attention network (GAT) to encode the history trajectory and social interaction clues. In Causal-STGAT, we replace the feature vector from M-LSTM with the counterfactual intervention (the mean vector of all trajectory features, or the zero vector). Then we respectively conduct predictions with original connected feature and counterfactual connected feature whose history trajectory part is changed. Finally, we adopt the difference as the prediction.

Causal-STGCNN: In Social-STGCNN, the trajectories in a scene are represented as a graph \( G = \{N, E\} \), where nodes \( N \) denote the past trajectories, while the adjacency matrix \( A \) in the GCN denotes the social interactions. Then a time-extrapolator (TE) CNN is applied as the trajectory predictor. In Causal-STGCNN, we replace the original nodes \( N \) by counterfactual intervention and maintain the values of interaction clues \( A \). Then we respectively feed factual and counterfactual outputs to the TE-CNN predictor and compute the difference to get the causal prediction.

4. Experiments

4.1. Datasets and Experimental Settings

Datasets: We conduct experiments on two publicly available human trajectory prediction datasets: ETH [38] and UCY [24]. The human trajectories in both two datasets are captured in the real world scenes with rich social interactions. These datasets contain five unique scenes: Zara1, Zara2, Univ, ETH, and Hotel with 1536 detected pedestrians. All trajectories in datasets are sampled every 0.4 seconds. The experimental settings follow the previous methods like Social-GAN [15], Social BiGAT [21], and STGAT [18]. We also use the leave-one-out approach to train and validate our model on 4 sets and test on the remaining one. During evaluation, the first 3.2 seconds (8 frames) are observed and the next 4.8 seconds (12 frames) are to be predicted.

Evaluation Metrics: We adopt the same evaluation metrics as prior methods [15, 21, 18], including Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE computes the mean square error (MSE) of predicted trajectory and ground-truth trajectory, while FDE computes the L2 distance between the final locations of predicted and ground-truth trajectories. Since both baselines Social-STGCNN [29] and STGAT [18] are generative methods, we also follow the evaluation method in Social-GAN [15]. We generate 20 samples based on the predicted distribution and select the closest sample to the ground truth for ADE and FDE metrics.

4.2. Quantitative Evaluation

Evaluation of Counterfactual Analysis: To verify the effectiveness of our method, we compare the performance of our Causal-STGAT, Causal-STGCNN with the RNN-based baseline STGAT [18] and CNN-based Social-STGCNN [29] methods. The performance comparison about our causal-based methods, baseline methods, and other SOTA methods is summarized as Table 1. Note that the * in Table 1 means our reproduced results trained with the official released code. 2 Due to different implementation environments, the performance of our reproduced baselines are slightly lower than the performance reported in the paper. For the fair comparison, all hyperparameters and environments of our Causal-STGAT and Causal-STGCNN methods are same with the reproduced baselines. As shown in Table 1, our causal prediction method can consistently improve the performance than the original likelihood-based prediction with different baselines on most of datasets. Specifically, Causal-STGAT obtains +0.07/+0.16 ADE/FDE improvement than the baseline STGAT method on the average of 5 scenes, while Causal-STGCNN outperforms the reproduced Social-STGCNN method by over +0.02/+0.09 ADE/FDE results.

Besides, we also compare the proposed causal based

\[^{2}\text{We reproduced the STGAT [18] methods with the released code from https://github.com/leung-xx/STGAT, while the code of Social-STGCNN [29] comes from https://github.com/abduallahmohamed/Social-STGCNN.}\]
Table 1. Comparison with several state-of-the-art models. We notice that our causal-based methods have the best average error for RNN-based and CNN-based methods. The lower is the better.

<table>
<thead>
<tr>
<th>RNN-based Method</th>
<th>ETH</th>
<th>HOTEL</th>
<th>ZARA1</th>
<th>ZARA2</th>
<th>UNIV</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.10/2.41</td>
<td>0.86/1.91</td>
<td>0.41/0.88</td>
<td>0.52/1.11</td>
<td>0.61/1.31</td>
<td>0.70/1.52</td>
</tr>
<tr>
<td>S-LSTM [1]</td>
<td>0.10/2.35</td>
<td>0.79/1.76</td>
<td>0.47/1.00</td>
<td>0.56/1.17</td>
<td>0.67/1.40</td>
<td>0.72/1.54</td>
</tr>
<tr>
<td>SGAN [15]</td>
<td>0.81/1.52</td>
<td>0.72/1.61</td>
<td>0.34/0.69</td>
<td>0.42/0.84</td>
<td>0.60/1.26</td>
<td>0.58/1.18</td>
</tr>
<tr>
<td>Sophie [41]</td>
<td>0.70/1.43</td>
<td>0.76/1.67</td>
<td>0.30/0.63</td>
<td>0.38/0.78</td>
<td>0.54/1.24</td>
<td>0.54/1.15</td>
</tr>
<tr>
<td>SR-LSTM [56]</td>
<td>0.63/1.25</td>
<td>0.37/0.74</td>
<td>0.41/0.90</td>
<td>0.32/0.70</td>
<td>0.51/1.10</td>
<td>0.45/0.94</td>
</tr>
<tr>
<td>Social-BiGAT [21]</td>
<td>0.69/1.29</td>
<td>0.49/1.01</td>
<td>0.30/0.62</td>
<td>0.36/0.75</td>
<td>0.55/1.32</td>
<td>0.48/1.00</td>
</tr>
<tr>
<td>MATF [57]</td>
<td>1.33/2.49</td>
<td>0.51/0.95</td>
<td>0.44/0.93</td>
<td>0.34/0.73</td>
<td>0.56/1.19</td>
<td>0.64/1.26</td>
</tr>
<tr>
<td>MATF GAN [57]</td>
<td>1.01/1.75</td>
<td>0.43/0.80</td>
<td>0.26/0.45</td>
<td>0.26/0.57</td>
<td>0.44/0.91</td>
<td>0.48/0.90</td>
</tr>
<tr>
<td>IDL [54]</td>
<td>0.59/1.30</td>
<td>0.46/0.83</td>
<td>0.22/0.49</td>
<td>0.23/0.55</td>
<td>0.51/1.27</td>
<td>0.40/0.89</td>
</tr>
<tr>
<td>PIF [26]</td>
<td>0.73/1.65</td>
<td>0.30/0.59</td>
<td>0.60/1.27</td>
<td>0.38/0.81</td>
<td>0.31/0.68</td>
<td>0.46/1.00</td>
</tr>
<tr>
<td>STGAT [18]</td>
<td>0.65/1.12</td>
<td>0.35/0.66</td>
<td>0.34/0.69</td>
<td>0.29/0.60</td>
<td>0.52/1.10</td>
<td>0.43/0.83</td>
</tr>
<tr>
<td>STGAT*</td>
<td>0.73/1.39</td>
<td>0.38/0.72</td>
<td>0.35/0.69</td>
<td>0.32/0.64</td>
<td>0.57/1.22</td>
<td>0.47/0.93</td>
</tr>
<tr>
<td>Causal-STGAT</td>
<td>0.60/0.98</td>
<td>0.30/0.54</td>
<td>0.32/0.64</td>
<td>0.28/0.58</td>
<td>0.52/1.10</td>
<td>0.40/0.77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CNN-based Method</th>
<th>ETH</th>
<th>HOTEL</th>
<th>ZARA1</th>
<th>ZARA2</th>
<th>UNIV</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN [31]</td>
<td>1.04/2.07</td>
<td>0.59/1.17</td>
<td>0.43/0.90</td>
<td>0.34/0.75</td>
<td>0.57/2.32</td>
<td>0.59/1.22</td>
</tr>
<tr>
<td>Social-STGCNN [29]</td>
<td>0.64/1.11</td>
<td>0.49/0.85</td>
<td>0.34/0.53</td>
<td>0.30/0.48</td>
<td>0.44/0.79</td>
<td>0.44/0.75</td>
</tr>
<tr>
<td>Social-STGCNN*</td>
<td>0.66/1.12</td>
<td>0.40/0.65</td>
<td>0.35/0.56</td>
<td>0.31/0.47</td>
<td>0.51/0.93</td>
<td>0.45/0.75</td>
</tr>
<tr>
<td>Causal-STGCNN</td>
<td>0.64/1.00</td>
<td>0.38/0.45</td>
<td>0.34/0.53</td>
<td>0.32/0.49</td>
<td>0.49/0.81</td>
<td>0.43/0.66</td>
</tr>
</tbody>
</table>

Table 2. Evaluation of Causal-STGAT for conducting different counterfactual interventions.

<table>
<thead>
<tr>
<th>Method</th>
<th>ETH</th>
<th>HOTEL</th>
<th>ZARA1</th>
<th>ZARA2</th>
<th>UNIV</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>STGAT* (baseline)</td>
<td>0.73/1.39</td>
<td>0.38/0.72</td>
<td>0.35/0.69</td>
<td>0.32/0.64</td>
<td>0.57/1.22</td>
<td>0.47/0.93</td>
</tr>
<tr>
<td>Causal-STGAT-Zero</td>
<td>0.60/0.98</td>
<td>0.30/0.54</td>
<td>0.32/0.64</td>
<td>0.28/0.58</td>
<td>0.52/1.10</td>
<td>0.40/0.77</td>
</tr>
<tr>
<td>Causal-STGAT-Mean</td>
<td>0.65/1.17</td>
<td>0.33/0.60</td>
<td>0.35/0.71</td>
<td>0.29/0.60</td>
<td>0.54/1.15</td>
<td>0.44/0.84</td>
</tr>
<tr>
<td>Causal-STGAT-Random</td>
<td>0.59/1.05</td>
<td>0.31/0.55</td>
<td>0.34/0.70</td>
<td>0.28/0.59</td>
<td>0.53/1.13</td>
<td>0.42/0.80</td>
</tr>
</tbody>
</table>

Table 3. Parameters size and inference time of baseline models compared to ours. The lower is better.

<table>
<thead>
<tr>
<th>Method</th>
<th>Social-STGCNN</th>
<th>Causal-STGCNN</th>
<th>STGAT</th>
<th>Causal-STGAT*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inference Speed</td>
<td>0.0116</td>
<td>0.0124</td>
<td>0.3343</td>
<td>0.3418</td>
</tr>
<tr>
<td>Parameters Count</td>
<td>7.6k</td>
<td>7.6k</td>
<td>56k</td>
<td>56k</td>
</tr>
</tbody>
</table>

method with other SOTA methods, such as IDL [54], Social-BiGAT [21] and MATF [57]. As shown in Table 1, we achieve the SOTA performance with slight improvement over the RNN-based IDL [54] method and CNN-based Social-STGCNN [29] method. It also demonstrates the effectiveness of the proposed counterfactual analysis method. For both our methods in Table 1, we conducted the counterfactual intervention by replacing the features with zero vector. Other attempts are introduced below.

**Evaluation of Counterfactual Implementations:** We have attempted different implementations of counterfactual intervention, including using the zero vector, the mean of all feature vector or a random vector sampled from a uniform distribution with $[-0.1, 0.1]$. These intervention implementations (“Zero”, “Mean”, “Random”) with limited computing cost are usually adopted [44] for causal model. For the zero vector and mean vector, the counterfactual intervention in the training and testing stage is invariant. While for random vector, we sampled the vector from a uniform distribution in the training stage and using the expectation of uniform distribution (zero vector) in the testing stage to avoid introducing the bias for testing data. As shown in bottom part of Table 2, all implementations obtain obvious improvements for the baselines, which demonstrates the generality of the proposed counterfactual analysis method. Furthermore, 1) The “Zero” obtains slightly better performance than the others. It might be because the “Zero” is a harder counterfactual intervention, which better emphasizes...
Figure 3. Qualitative analysis for our causal prediction methods. We compare our causal prediction methods (Causal-STGCNN and Causal-STGAT) with corresponding baselines in four scenarios: (a) a person walking alone; (b) two pedestrians gathering; (c) two persons meeting from the same direction; (d) two persons meeting from different directions.

Figure 4. Visualization examples of our Causal-STGCNN method in different scenes. We compare the trajectories generated by our Causal-STGCNN method and baseline Social-STGCNN method in the ETH and UCY datasets. Our causal-based method can solve complex scenes and social interactions with multiple humans by the counterfactual analysis. (Best viewed in color.)

the causal effect from history behaviors to future trajectories. 2) The performance of these implementations is close, which demonstrates the robustness of the implementation of counterfactual intervention.

Evaluation of Inference Speed and Model Size: Model size and inference speed are also critical for the deployment of the method in the real-world environment. We evaluated the speed of all models with all data in five scenes and compute the average inference time for one trajectory on one GTX 2080Ti GPU. For fairness, we fixed the batch-size as 1 for all methods and repeat the evaluation 3 times for average. The parameters cost and inference speed of our methods and baselines are summarized in Table 3. First, our counterfactual analysis method does not need any extra parameters since all parameters are shared with factual and counterfactual parts. Second, there is no such thing as a free lunch. The speed of our methods are lower than the baselines because of the extra computations for counterfactual analysis. However the extra speed cost is not heavy since the counterfactual analysis only uses part of network.
4.3. Qualitative Evaluation

We qualitatively analyze how our causal predictions are more reliable than conventional likelihood-based predictions. As shown in Figure 3, we analyze our methods in 4 different scenarios. Taking the (b) as an example, given two trajectories gathering, the conventional likelihood-based methods always predict they will meet each other since the negative effect brought by the training bias of the environment interaction, (like “most trajectories getting closer finally meet”). These training biases mislead the predictor to learn the spurious correlations of environments and ignore the real clues from history trajectory, (like “these persons tend to gather and move forward together”). By counterfactual analysis, we remove the negative effect contained in the counterfactual prediction from original predictions. It is effective to overcome the training bias and encourage the model to highlight the real causation itself.

Besides, we also provide the visualization examples with the environment scenes. As shown in Figure 4 and Figure 5, we respectively compare our Causal-STGCNN and Causal-STGAT methods with corresponding baselines in different scenes. We observe that our counterfactual analysis method can effectively captures the real causal relations instead of the biased environment interactions. Taking the second scene in ETH as an example, both CNN-based and RNN-based counterfactual analysis methods significantly outperform the baseline methods and generate more reliable predicted trajectories.

5. Conclusion

In this paper, we have presented a counterfactual analysis method to investigate the causalities between the observed clues and predicted trajectories. We apply the counterfactual intervention by replacing the features of trajectory clues with the counterfactual ones, and subtract this counterfactual prediction from original prediction. By this comparison, we encourage the model to learn the real causations of trajectory and alleviate the negative effects brought by the bias between training and deployment environments. Our counterfactual analysis is a plug-and-play module, which can be employed for off-the-shelf trajectory prediction models. In the experiments, we demonstrate the effectiveness of our counterfactual analysis method for different scenes, analyze the effects of different counterfactual implementations, and evaluate the generalization ability for different baseline methods.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102803, in part by the National Natural Science Foundation of China under Grant 61822603, Grant U1813218, and Grant U1713214, in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI), and in part by a grant from the Institute for Guo Qiang, Tsinghua University.
References


