

# Socially-Informed Reconstruction for Pedestrian Trajectory Forecasting

Haleh Damirchi    Ali Etemad    Michael Greenspan  
Dept. ECE & Ingenuity Labs Research Institute  
Queen’s University, Kingston, Canada

{haleh.damirchi, ali.etemad, michael.greenspan}@queensu.ca

## Abstract

*Pedestrian trajectory prediction remains a challenge for autonomous systems, particularly due to the intricate dynamics of social interactions. Accurate forecasting requires a comprehensive understanding not only of each pedestrian’s previous trajectory but also of their interaction with the surrounding environment, an important part of which are other pedestrians moving dynamically in the scene. To learn effective socially-informed representations, we propose a model that uses a reconstructor alongside a conditional variational autoencoder-based trajectory forecasting module. This module generates pseudo-trajectories, which we use as augmentations throughout the training process. To further guide the model towards social awareness, we propose a novel social loss that aids in forecasting of more stable trajectories. We validate our approach through extensive experiments, demonstrating strong performances in comparison to state-of-the-art methods on the ETH/UCY and SDD benchmarks.*

## 1. Introduction

Pedestrian trajectory prediction estimates a target pedestrian’s future location after examining the observed locations of the pedestrian in the scene [1]. This capability is essential for safe route planning of autonomous vehicles [31] as it requires accurate and reliable predictions to ensure the safety of pedestrians and drivers [3]. By accurately forecasting future paths of pedestrians in the scene, proactive measures can be taken to prevent potential accidents [22]. Similarly, trajectory forecasting helps to understand and model pedestrian behaviour, allowing vehicles to anticipate and react to pedestrians’ movements in real time [4, 18, 32]. Additionally, it can assist in the analysis of pedestrian movement patterns in crowded areas to optimize pedestrian infrastructure [26].

A key aspect of human trajectory forecasting that distinguishes it apart from other time-series problems [24, 29] is the inherent human element [25]. Humans are social beings, and interactions between individuals significantly influence

how we navigate through (especially crowded) spaces [1]. For instance, when walking in a crowded area, people naturally adjust their paths to avoid potential collisions with others. This social dynamic is critical in predicting human movements accurately. Prior works [6, 13, 14, 34] have shown that considering social interactions in trajectory forecasting models can lead to more precise predictions, as these interactions play a crucial role in determining individual movement patterns. Another challenge in human trajectory forecasting, which is shared by many deep learning applications, is the high cost of collecting labeled data. A standard approach to mitigate this issue is to apply standard augmentations to the data prior to training [2, 5, 30]. However, while standard augmentation techniques have been initially developed and are well-suited for static images, they do not necessarily translate well to trajectory data. As a result, developing methods to generate more effective augmented trajectories, particularly those that consider social interactions, can significantly enhance the learning process, leading to better performance in real-world applications.

In this paper, to address the problems described above, we propose a novel trajectory forecasting model. Our method uses several distinct elements to improve representations and enhance the stability of the predictions. First, it uses a trajectory reconstruction module to operate alongside the forecaster. The inclusion of this module improves the learned representations and simultaneously allows the reconstructed pseudo-trajectories to be fed back to the training pipeline as augmentations. Our model only selects challenging pseudo-trajectories to be used as augmentations to ensure that the model is exposed to difficult scenarios that improve its robustness and generalization capabilities. Second, our model includes a novel loss, called social loss, to enforce socially accurate predictions in the generated future time-stamps. We illustrate an overview of our method in Figure 1. To evaluate the performance of our solution, we experiment on 5 popular pedestrian trajectory forecasting benchmarks [10, 17], namely ETH, Hotel, Univ, Zara1, Zara2, and Stanford Drone Dataset (SDD) [20]. Our experiments show our method to outperform the state-of-the-art based on standard forecasting

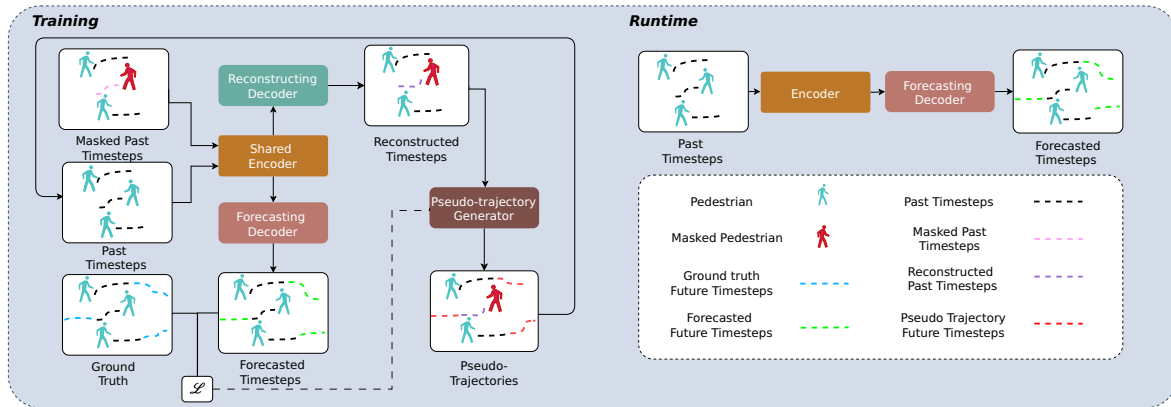


Figure 1. Overview of our proposed method. During training we continuously generate new socially-aware samples and add them to the training set. At runtime, we only use the encoder and forecaster to predict future timesteps.

metrics, while showing more stable performance across all predictions (as opposed to only the top prediction), especially as a result of our social loss. Various ablation and sensitivity studies demonstrate the impact of different components of our method.

In summary, we make the following contributions. (1) We propose a novel solution for human trajectory forecasting. Our method uses a reconstructor alongside the forecasting module to learn stronger trajectory representations as well as to generate pseudo-trajectories, the more difficult of which is used as augmentations throughout the training process. (2) We propose the use of a novel loss to enforce physically and socially viable trajectories, which aids in generating more stable predictions. (3) We perform experiments on the widely utilized benchmarks and demonstrate that our method outperforms state-of-the-art trajectory prediction methods. We make our code publicly available at <https://github.com/thisishale/SocRec>.

## 2. Related Work

### 2.1. Trajectory Forecasting

A variety of methods have focused on the social attribute of trajectories, in which multiple pedestrians are influenced by each other. Social GAN [6] employed adversarial training to improve the accuracy of their model in complicated social scenarios. By synthesizing realistic trajectories, it refines predictions by incorporating both individual and social context. A pooling module was used to aggregate features from other pedestrians in the scene, though this module could cause the model to struggle in crowded environments. Later, Trajectron++ [23] introduced a social model for trajectory forecasting, where interactions were modelled by summing the feature vectors of neighboring pedestrians. While computationally efficient and a relatively simple operation, this method could oversimplify the interactions in a scene. So-

cial Spatio-Temporal Graph Convolutional Neural Network (Social-STGCNN) [13] departs from previous aggregation methods and uses a graph network with an adjacency matrix based on distances between pedestrians to forecast future trajectories. Although this operation allows the model to account for neighbouring pedestrians, it may not adapt to changing social contexts such as group behaviors due to the non-learnable adjacency matrix. To improve the social connection between trajectories, Agentformer [34] proposed a new form of attention in their transformer architecture. Whereas methods prior to Agentformer generally modelled the social and temporal information separately, the new form of attention jointly addressed both the temporal and the social aspects of trajectory prediction. Later, Social Implicit [14] proposed to forecast each pedestrian’s future trajectory by using sub-modules trained for specific speed ranges. This approach may not fully capture scenarios where pedestrians adjust their speed in response to others.

A different set of methods explore the importance of goals, i.e. the final timestep, for trajectory prediction [12, 32, 35]. Unlike previous models that focus on a single, long term goal, SGNet [28] introduced a step-wise goal estimator which dynamically estimated and utilizes goals at multiple timesteps. YNet [11] cast goal estimation and future trajectory prediction into image form, utilizing convolutional neural networks in the form of a UNet architecture to predict both the goals and the future trajectory by sampling from the generated heatmap at the network output. However, both SGNet and YNet lack a structure for the social prediction of trajectories.

### 2.2. Social, Psychological, and Behavioral Factors

Incorporating environmental context, psychological factors and behaviour patterns of pedestrians is crucial for predicting trajectories and motions of humans. Psychological and behavior attributes such as personal space and goal-

oriented behaviour is explored and simulated in [7]. In this study a simulation is implemented where pedestrians adjust their trajectories based on attractive and repulsive forces in their environment, which reflects psychological tendencies. They conclude that pedestrians experience *motivations to act*, which drives them to accelerate or decelerate as a reaction to the perceived information from their environment. In [16], an extended Kalman filter is used to create a dynamic model to consider the social interactions between pedestrians for motion tracking. They also use contextual information such as pedestrian heading to estimate the destination of pedestrians. Additionally, pedestrian walking style is explored in [21], where pedestrian trajectories are categorized into clusters based on their *social sensitivity*. A low social sensitivity value indicates that a pedestrian’s navigation is not dependant on other targets in the scene. Each cluster is evaluated in a separate model that is specialized for its corresponding category. Pedestrian intention has also been used in previous works for pedestrian trajectory prediction from a vehicle’s view point [19]. In this study, cropped images of pedestrians and their bounding box locations on dashcam images are fed to a recurrent neural network, and the future locations of bounding boxes on the image are estimated.

## 3. Method

### 3.1. Problem Setup

Let  $X_t = \{x_t^1, x_t^2, \dots, x_t^N\}$  represent a set of 2D locations of  $N$  pedestrians at time  $t$ , where  $x_t^i \in \mathbb{R}^2$ . We denote  $S_p = \{X_1, X_2, \dots, X_{t_p}\}$  and  $S_f = \{X_{t_p+1}, X_{t_p+2}, \dots, X_{t_f}\}$  as sequences of locations of  $N$  pedestrians at the past and future timesteps, respectively, with  $t_p$  and  $t_f$  as the present and final timesteps. The goal of social trajectory forecasting is to accurately predict multiple plausible future trajectories given  $S_p$ , such that  $x_t^i - x_t^j > \epsilon$ ,  $\forall i \neq j$ , where  $\epsilon$  is the minimum acceptable distance between two pedestrians at a given timestep.

### 3.2. Proposed Approach

To address the above-mentioned problem, we propose a new solution which consists of three key components: a social forecaster, a social reconstructor, and a pseudo-trajectory generator. At a high level, the social forecaster predicts the future trajectory locations for each pedestrian in the scene given their locations in past timesteps. The social reconstructor reproduces the locations at past timesteps which have been partially masked for each pedestrian. Finally, the pseudo-trajectory generator creates new trajectory sequences by sampling from the reconstructed trajectories, which will then be used in future training iterations. The architecture for our proposed approach is illustrated in Figure 2. Following, we describe each element of our method in detail.

#### 3.2.1 Social Forecaster

The past trajectories  $S_p$  are first fed to the social forecaster module, which is inspired by the Conditional Variational Autoencoder-based forecaster proposed in [34] and consists of four components: Encoder, Conditional Prior Network (CPN), Predictor Posterior Distribution Network (PPDN), and Forecasting Decoder. The encoder module consists of a transformer encoder which uses agent-aware attention [34]. The input to this attention unit is query  $Q$ , key  $K$  and value  $V$ . The attention output is calculated as:

$$\text{SocialAtt}(Q, K, V) = \text{softmax}\left(\frac{A}{\sqrt{d_m}}\right)V, \quad (1)$$

$$A = M \odot (Q_{self} K_{self}^T) + (1 - M) \odot (Q_{other} K_{other}), \quad (2)$$

where  $A$  is the attention weight matrix;  $d_m$  is the model dimension;  $Q_{self} = QW_{self}^Q$  and  $Q_{other} = QW_{other}^Q$  are the query values projected by weights  $W_{self}^Q$  and  $W_{other}^Q$ , and;  $K_{self} = KW_{self}^K$  and  $K_{other} = KW_{other}^K$  are the key values projected by weights  $W_{self}^K$  and  $W_{other}^K$ . In Eq. 2, for matrix  $M$ , the attention mask between pedestrian  $i$  and  $j$  is defined as  $M_{ij} = \mathbb{I}(i \bmod N = j \bmod N)$ . When  $i = j$ , the attention weight matrix  $A$  is masked to include only those weights that correspond to the relationship between the pedestrian itself at different timesteps. However, when  $i \neq j$ , the attention weight matrix represents the relationship between pedestrian  $i$  and pedestrian  $j$  at different timesteps.

To produce the latent code  $z \in \mathbb{R}^{N \times d_z}$  for all  $N$  target pedestrians, we pass the encoded features to the CPN module, which is a linear layer. This module generates the parameters  $\mu_i^{p_f}$  and  $\sigma_i^{p_f}$  for the prior Normal distribution  $p_{\phi_F}(z|S_p) = \mathcal{N}(\mu_i^{p_f}, \text{Diag}(\sigma_i^{p_f})^2)$ , where  $\phi_F$  is the set of parameters for the CPN module. PPDN consists of an agent-aware cross-attention mechanism (explained earlier), followed by a linear layer similar to that of the CPN. This module utilizes both ground truth (future) and past timesteps to output the parameters of the posterior distribution  $q(z|S_p, S_f) = \mathcal{N}(\mu_i^{q_f}, \text{Diag}(\sigma_i^{q_f})^2)$ . The latent code generated by PPDN and the ground truth future timesteps are then passed on to the transformer forecasting decoder at training time. This decoder consists of an agent-aware self-attention module followed by cross-attention. Masking is implemented in the agent-aware self-attention inside the decoder to prevent the previous timesteps from peeking into the future ground truth values.

To train the social forecaster, the negative evidence lower bound (ELBO) is employed as follows:

$$L_F = -\mathbb{E}_{q_{\phi_F}(z|S_p, S_f)}[\log(p_{\theta_F}(S_f|z, S_p))] + KL(q_{\phi_F}(z|S_p, S_f)||p_{\phi_F}(z|S_p)), \quad (3)$$

where  $p_{\theta_F}(S_f|z, S_p)$  is the conditional likelihood. The initial term in the formula denotes the prediction error, for

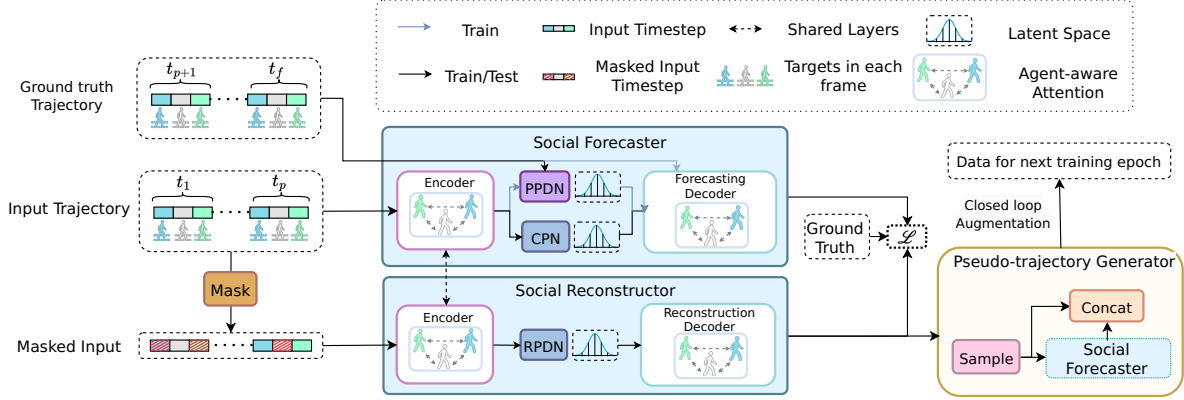


Figure 2. A detailed depiction of our method is presented. Our proposed model uses a social forecaster, a social reconstructor, and a pseudo-trajectory generator to augment the training data. The loss shown here consists of the forecaster loss, the reconstructor loss, and our novel social loss.

which the Mean Squared Error (MSE) between the predicted future timesteps and the ground truth is utilized, resulting in the following reformulation:

$$L_F = \text{MSE}(\tilde{S}_f, S_f) + KL(q_{\phi_F}(z|S_p, S_f)||p_{\phi_F}(z|S_p)), \quad (4)$$

where  $\tilde{S}_f$  is the forecasted trajectory. At training, the loss function minimizes the divergence between the posterior  $q(z|S_p, S_f)$  and conditional prior distributions  $p_{\phi_F}(z|S_p)$ , while ensuring the accuracy of the predictions. However, at test time, ground truth future sequences  $S_f$  are not available. Thus, PPDN is only used during training to condition the predictions on the distribution of future timesteps.

To guide the social forecaster module towards avoiding overlap between predicted pedestrian locations, we introduce a novel loss term that penalizes the model for forecasting future pedestrian locations that are closer to each other than a threshold  $\epsilon$ . The proposed loss function is given by

$$L_{\text{Social}} = \frac{1}{m} \sum_{t=t_p}^{t_f} \sum_{i=1}^N \sum_{j=i}^N \max(0, \epsilon - \text{dist}(\tilde{S}_f(i, t), \tilde{S}_f(j, t))), \quad (5)$$

where  $\text{dist}(\tilde{S}_f(i, t), \tilde{S}_f(j, t))$  is the squared euclidean distance between locations of pedestrians  $i$  and  $j$  at timestep  $t$  and  $m = N(N-1)/2$ .

### 3.2.2 Social Reconstructor

To further enforce the training process and provide stronger encoded representations of past trajectories  $S_p$ , our model includes a social reconstructor module in parallel with the social forecaster, where the encoder between the two modules are shared. To create the input for the social reconstructor, we mask the past trajectory  $S_p$  by a ratio of  $R$ . We zero out  $R\%$  of the total timesteps in the trajectory selected randomly, which we name  $S_p^{\text{masked}}$ . The social reconstructor

module is designed to fill in the missing points in  $S_p^{\text{masked}}$  utilizing a Variational Autoencoder (VAE). This module consists of an Encoder, a Reconstructor Posterior Distribution Network (RPDN), and a Reconstruction Decoder. The masked input  $S_p^{\text{masked}}$  is first passed through the encoder module, which is shared with the social forecaster module. Subsequently, the RPDN, which is a linear layer, estimates the parameters  $\mu_i^{qr}$  and  $\sigma_i^{qr}$  for the posterior distribution  $q(z|S_p) = \mathcal{N}(\mu_i^{qr}, \text{Diag}(\sigma_i^{qr})^2)$  of the encoded representations. The sampled latent code obtained from this module is then passed through the reconstruction decoder, which is a transformer decoder that uses the agent-aware attention described earlier.

The loss function to train the reconstructor module is:

$$L_R = -\mathbb{E}_{q_{\phi_r}(z|S_p)} \log p_{\theta_r}(S_p|z) + KL(q_{\phi_r}(z|S_p)||p_{\phi_r}(z)). \quad (6)$$

Here,  $\phi_r$  is the set of parameters for the RPDN and  $p_{\phi_r}(z)$  is a Gaussian distribution with mean of 0 and variance of 1. Similar to Eq. 3, the first term is replaced with the MSE function as

$$L_R = \text{MSE}(\tilde{S}_p, S_p) + KL(q_{\phi_r}(z|S_p)||p_{\phi_r}(z)), \quad (7)$$

where  $\text{MSE}(\tilde{S}_p, S_p)$  is the Mean Squared Error between the reconstructed sequence  $\tilde{S}_p$  and the observed sequences.

Similar to the forecaster module, we propose to use a social loss for the reconstruction module to reconstruct the masked sequences while avoiding overlaps with other pedestrians in the scene as follows:

$$L_{\text{Social}} = \frac{1}{m} \sum_{t=1}^{t_p} \sum_{i=1}^N \sum_{j=i}^N \max(0, \epsilon - \text{dist}(\tilde{S}_p(i, t), \tilde{S}_p(j, t))). \quad (8)$$



### 3.2.3 Total Loss

The final loss function is formulated as:

$$\mathcal{L}_{Total} = w_1 L_F + w_2 L_R + w_3 (L_{SoCF} + L_{SoCR}), \quad (9)$$

where  $w_1, w_2$  are corresponding weights for the CVAE, and VAE (reconstructor) modules, and  $w_3$  is the weight for the social loss functions for both the forecaster and reconstructor.

### 3.2.4 Pseudo-trajectory Generation

We propose a pseudo-trajectory generator for augmenting the training set with samples that are challenging for the forecaster module. We follow the strategy proposed in [33] and monitor the fluctuations in the loss value for each sample. Throughout training, each sample goes through phases of ‘descending’ where the loss decreases for that sample between two consecutive epochs, and ‘ascending’ where it increases. Accordingly, difficult samples tend to experience more ‘ascending’ states than ‘descending’ throughout the training. As the approach introduced in [33] is primarily designed for classification, we adapt the process for our specific regression task of trajectory forecasting:

$$d_{i,e} = \min(L_F(i, e) - L_F(i, e-1), 0) \ln\left(\frac{L_F(i, e)}{L_F(i, e-1)}\right), \quad (10)$$

$$a_{i,e} = \max(L_F(i, e) - L_F(i, e-1), 0) \ln\left(\frac{L_F(i, e)}{L_F(i, e-1)}\right), \quad (11)$$

where  $L_F(i, e)$  is the loss for instance  $i$  at epoch  $e$ . To label an instance as difficult after  $N_c$  epochs of observing  $a_{i,e}$  and  $d_{i,e}$ , we count the number of epochs in which  $d_{i,e} > a_{i,e}$ , meaning that the training loss has been descending for that instance. The cumulative sum of this indicator over  $N_c$  epochs is calculated as follows:

$$Count(i) = \sum_{e=1}^{N_c} \mathbb{I}(d_{i,e} > a_{i,e}). \quad (12)$$

An instance is flagged as difficult if  $Count(i) < D \times N_c$ , where  $D$  is a predefined threshold parameter.

During the training process, we initially focus on training the social forecaster and social reconstructor across the entire dataset for a fixed number of epochs,  $N_T$ . This step is important in establishing a robust foundation to enable reasonable reconstructions of difficult cases as part of a continuous training process.

The reconstructed version of the samples that are determined to be challenging, its reconstructed masked sequence,  $\tilde{S}_o$ , are then processed through the social forecaster module to predict future timesteps. We then concatenate the reconstructed sequences with their corresponding forecasted segments as follows:

$$S^{Aug} = \tilde{S}_p \oplus SF(\tilde{S}_p), \quad (13)$$

where  $S^{Aug}$  is the newly augmented scene, and  $SF(\cdot)$  is the social forecaster function. The integration of identifying and addressing difficult samples occurs in tandem with ongoing training, enhancing the model’s capacity to effectively handle such cases without interrupting the training process. This continuous loop ensures that our model dynamically adapts and improves its performance on challenging instances within the training set.

## 4. Experiments

**Datasets.** We evaluate our method using the ETH/UCY benchmark [10, 17] and SDD [20]. The ETH/UCY benchmark features real world pedestrian trajectories across five scenes including: ETH, Hotel, Univ, Zara1 and Zara2. Each scene contains both multiple and single trajectories, with the multiple trajectory scenarios demonstrating collision avoidance and group behaviour. We use the same coordinate format originally used by SGAN [6]. For SDD, following DAG-NET [15] and Social Implicit [14], we convert the pedestrian locations from pixels to metric coordinates.

**Evaluation Metrics.** We use the following three metrics to evaluate our method.

*Average Distance Error (ADE<sub>K</sub>):* This metric calculates the average  $L_2$  distance between the ground truth and each of  $K$  trajectories predicted by the model. The formulas for minimum and mean ADE calculation are defined as follows:

$$ADE_K^{min} = \frac{1}{T} \min_{k=1}^K \sum_{t=1}^T \|\hat{S}_f^k(n, t) - S_f(n, t)\|^2, \quad (14)$$

$$ADE_K^{mean} = \frac{1}{KT} \sum_{k=1}^K \sum_{t=1}^T \|\hat{S}_f^k(n, t) - S_f(n, t)\|^2. \quad (15)$$

*Final Distance Error (FDE<sub>K</sub>):* This metric measures the distance between the predicted final destination and the ground truth final coordinates for all  $K$  predicted trajectories. The formulas for minimum and mean FDE are defined as:

$$FDE_K^{min} = \min_{k=1}^K \|\hat{S}_f^k(n, T) - S_f(n, T)\|^2, \quad (16)$$

$$FDE_K^{mean} = \frac{1}{K} \sum_{k=1}^K \|\hat{S}_f^k(n, T) - S_f(n, T)\|^2. \quad (17)$$

Since all benchmarks use metric measurements, the values of  $ADE_K$  and  $FDE_K$  are expressed in meters.

*Kernel Density Estimate-based Negative Log Likelihood (KDE-NLL):* This metric, which we refer to as KDE, calculates the mean negative log likelihood of the ground truth under the predicted distribution. The distribution is created by fitting a kernel density estimate to trajectory samples [8, 27]. As KDE measures the negative log likelihood, it is unitless.

**Implementation Details.** We follow a leave-one-out strategy used in previous works [6, 13, 14, 34], where we train our model on 4 sets of scenes, and evaluate on the remaining set.

Table 1.  $ADE_{20}^{min} \downarrow / FDE_{20}^{min} \downarrow$  (top) and  $KDE \downarrow$  (bottom) results for the proposed method, and state-of-the-art on the ETH/UCY benchmarks with  $K = 20$ .

Method	ETH	Hotel	Univ	Zara1	Zara2	Avg
SGAN [6]	0.63/1.00 9.388	0.42/0.80 6.328	0.50/1.01 10.557	0.29/0.58 3.484	0.27/0.46 2.639	0.42/0.77 6.479
S-STGCNN [13]	0.64/0.93 2.857	0.33/0.55 1.215	0.43/0.75 3.395	0.30/0.50 1.590	0.26/0.44 0.955	0.39/0.63 2.002
S-Implicit [14]	0.66/1.34 7.067	0.17/0.31 0.366	0.29/0.56 1.258	0.24/0.47 0.855	0.21/0.42 0.486	0.31/0.62 2.006
YNet [11]	<b>0.41/0.53</b> 4.613	<b>0.12/0.16</b> 1.864	<b>0.27/0.49</b> 2.283	<b>0.19/0.31</b> 2.250	<b>0.15/0.27</b> 0.904	<b>0.23/0.35</b> 2.383
Agentformer [34]	<b>0.45/0.75</b> 3.563	<b>0.14/0.22</b> 0.789	<b>0.25/0.45</b> 1.439	<b>0.18/0.30</b> 0.793	<b>0.14/0.24</b> -0.380	<b>0.23/0.39</b> 1.241
Ours	<b>0.46/0.75</b> <b>2.789</b>	<b>0.12/0.20</b> <b>-0.531</b>	0.28/0.51 <b>1.031</b>	0.21/0.39 <b>0.344</b>	0.16/0.29 <b>-0.704</b>	<b>0.25/0.43</b> <b>0.586</b>

We observe the trajectories for 3.2 seconds (8 timesteps) and predict the future trajectories over the next 4.8 seconds (12 timesteps). Following [34], we center crop each scene, and apply a random rotation in the range of 0 to 360 degrees.

Our model contains 8 attention heads in every encoder and decoder module. We train the model on an A100 Nvidia GPU using Adam [9] with a dropout of 0.1 and a learning rate of  $1e-4$ . All models are trained for 100 epochs. We used grid search to tune our hyperparameters. For more training and architectural details, see Appendix A2.

## 5. Results

**Performance.** We compare our method against several models, comprising Social Implicit [14], Social GAN [6], Social STGCNN [13], YNet [11], and Agentformer [34]. The  $ADE_{20}^{min} / FDE_{20}^{min}$  and KDE results for  $K=20$  model predictions on ETH/UCY are presented in Table 1. Looking at the  $ADE_{20}^{min} / FDE_{20}^{min}$  results, our method shows competitive performance compared to previous methods. However, relying solely on Eqs. 14 and 16, which are ‘best of K’ metrics, for evaluations may in some cases be misleading. While these metrics evaluate how well one of the produced set of  $K$  samples falls close to the ground truth, it could also improve merely by a method producing a set of trajectories with a large variance that spans across numerous possible future scenarios, and then simply choosing that trajectory that best matches the ground truth. In practice, during inference, ground truth is unknown, and so such a metric could lead us to a low-confidence prediction.

Following previous works such as [23], to address this issue, we use KDE which penalizes the distributions for both inaccuracy and spread. Unlike the other metrics, KDE considers both the closeness to the true value, as well as the degree of dispersion of a set of hypothesized solutions. In this way, KDE simultaneously rewards accurate and less dispersed distributions, which together we denote as *stability*. The KDE results for our proposed method in Table 1 show

Table 2.  $ADE_1 \downarrow / FDE_1 \downarrow$  results on the ETH/UCY benchmarks.

Method	ETH	Hotel	Univ	Zara1	Zara2	Avg
SGAN [6]	1.06/2.14	0.61/1.35	0.80/1.61	0.50/1.07	0.46/1.00	0.69/1.43
S-STGCNN [13]	1.23/2.12	0.61/1.21	0.72/1.38	0.54/1.08	0.46/0.90	0.71/1.34
S-Implicit [14]	1.06/2.21	0.29/0.55	<b>0.58/1.22</b>	<b>0.46/0.99</b>	0.41/0.83	0.56/1.16
Ynet [11]	<b>1.00/1.89</b>	0.35/0.72	0.81/1.79	0.52/1.10	0.45/1.02	0.63/1.30
Agentformer [34]	1.06/2.11	0.60/1.31	0.77/1.62	0.79/1.67	0.54/1.15	0.75/1.57
Ours	<b>1.00/2.00</b>	<b>0.28/0.54</b>	<b>0.63/1.30</b>	<b>0.46/0.98</b>	<b>0.35/0.76</b>	<b>0.54/1.12</b>

Table 3.  $ADE_{20}^{min} \downarrow$ ,  $FDE_{20}^{min} \downarrow$  (top) and  $KDE \downarrow$  (bottom) results for the proposed method, and state-of-the-art on the SDD dataset.

Method	$ADE_{20}^{min}$	$FDE_{20}^{min}$	KDE
DAG-NET [15]	0.53	1.04	1.76
S-Implicit [14]	0.47	0.89	3.89
Ours	<b>0.33</b>	<b>0.57</b>	<b>0.743</b>

an improvement (i.e. increased stability) over the state-of-the-art and previous works. Additionally, to further address the issue of ‘best of K’ predictions, we evaluate the methods based on  $K = 1$  and rely on  $ADE_1 / FDE_1$ . We present these results in Table 2, where we observe that our method generates stronger predictions in the majority of cases when compared to prior works. From the results presented in Table 1, we notice that Agentformer [34] uses all of the scenes in the benchmark regardless of the number of pedestrians in the scene. In contrast, Social Implicit [14], Social GAN [6], Social STGCNN [13], and YNet [11] train and test their methods on the same benchmark but exclude scenes with only a single pedestrian. To conduct a fair comparison, we retrained all methods excluding Agentformer [34] to include both the single pedestrian and multi-pedestrian scenes. This improved over the original reported results in Social STGCNN [13], Social GAN [6] and Social Implicit [14]. We also evaluate our method on SDD and depict the results in Table 3. Our results outperform the state-of-the-art on this dataset for all three metrics.

**Ablation Studies.** To investigate the contribution of each component in our proposed method, we perform extensive ablation studies on the ETH/UCY benchmark in Table 4. Specifically, we investigate the effect of five different scenarios: social reconstruction with three types of augmentation (difficulty-based sampling (ours), random sampling, and inverse sampling), social reconstruction without augmentation, and no social reconstruction. For random sampling augmentation, the sequences from the training set are chosen randomly and added to the dataset. For inverse sampling augmentation, we modified Eq. 12 to  $Count(i) = \sum_{e=1}^{N_e} \mathbb{I}(d_{i,e} < a_{i,e})$ . This modification implies selecting the easy sequences in the training data and further augmenting the dataset with these samples. We also

Table 4. Ablation studies of  $ADE_{20}^{min} \downarrow / FDE_{20}^{min} \downarrow$  (top) and  $KDE \downarrow$  (bottom) on the ETH/UCY benchmarks.

Social Recon.	Pseudo-trajectory			ETH	Hotel	Univ	Zara1	Zara2	Avg
	Difficulty based	Random	Inverse						
✓	✓	✗	✗	<b>0.46/0.75</b> 2.789	<b>0.12/0.20</b> -0.531	<b>0.28/0.51</b> 1.031	<b>0.21/0.39</b> 0.344	<b>0.16/0.29</b> -0.704	<b>0.25/0.43</b> 0.586
✓	✗	✓	✗	0.47/0.79 2.933	0.13/0.21 -0.438	<b>0.28/0.51</b> 1.103	0.23/0.41 0.445	0.17/0.30 -0.405	0.26/0.44 0.728
✓	✗	✗	✓	0.48/0.80 2.968	0.13/0.23 -0.216	<b>0.28/0.53</b> 1.140	0.22/0.41 0.396	0.17/0.30 -0.526	0.26/0.45 1.091
✓	✗	✗	✗	0.49/0.80 2.978	0.13/0.21 -0.467	0.28/0.52 1.357	<b>0.21/0.39</b> 0.358	0.16/0.29 -0.486	<b>0.25/0.44</b> 0.748
✗	✗	✗	✗	0.50/0.83 3.178	0.13/0.21 -0.351	0.30/0.53 1.431	0.22/0.41 0.465	0.17/0.31 -0.469	0.26/0.46 0.851

Table 5. Ablation studies of  $ADE_{20}^{min} \downarrow / FDE_{20}^{min} \downarrow$  (top) and  $KDE \downarrow$  (bottom) on the ETH/UCY benchmark. SL and SA denote Social Loss and Social Attention, respectively.

Method	ETH	Hotel	Univ	Zara1	Zara2	Avg
Ours	<b>0.46/0.75</b> 2.789	<b>0.12/0.2</b> -0.531	0.28/0.51 1.031	0.21/0.39 0.344	<b>0.16/0.29</b> -0.704	<b>0.25/0.43</b> 0.586
Ours w/o SL	0.47/0.76 <b>2.721</b>	0.13/0.21 -0.111	0.28/0.51 1.188	<b>0.20/0.38</b> <b>0.310</b>	0.17/0.31 -0.264	<b>0.25/0.43</b> 0.769
Ours w/o SL & w/o SA	0.47/0.75 3.086	0.14/0.21 -0.51	<b>0.26/0.52</b> <b>0.753</b>	0.22/0.42 0.399	<b>0.16/0.31</b> -0.591	<b>0.25/0.44</b> 3.137

Table 6. Count and percentage of overlap between pedestrians. SL and SA denotes Social Loss and Social Attention, respectively.

Method	ETH		Hotel		Univ		Zara1		Zara2	
	Count	%	Count	%	Count	%	Count	%	Count	%
Ours	<b>38</b>	<b>0.098</b>	<b>297</b>	<b>0.078</b>	6667	0.007	<b>802</b>	<b>0.074</b>	<b>3548</b>	<b>0.077</b>
Ours w/o SL	48	0.124	344	0.086	<b>6469</b>	<b>0.007</b>	850	0.075	4092	0.089
Ours w/o SL & w/o SA	66	0.170	757	0.200	15172	0.018	1415	0.132	12071	0.261

ablate the augmentation and reconstruction steps in the respective second last and last row of Table 4. All the ablated models in the last four rows result in comparable or worse  $ADE_{20}^{min}/FDE_{20}^{min}$  metrics compared to the proposed method. The results in Table 4 show our method to outperform all other variations for all datasets.

We examine the effect of social loss and social attention in our proposed method in Table 5. Our proposed method resulted in either the best or second best performance in all three evaluation metrics for all five datasets. To investigate the effect of social loss on the number of overlaps between forecasted pedestrian locations, we perform ablation studies on the elimination of both social attention and social loss in Table 6. Eliminating social loss resulted in an increase of overlaps between pedestrians in all subsets of ETH/UCY benchmark except for Univ. We observed that test data for the Univ subset consists of overlaps between locations, possibly due to noise and/or non-birdseye viewpoint recordings of scenes. The viewpoint of non-birdseye videos can result in overlapping pedestrians which do not occupy the same physical space, which does not adhere to the definition of

Table 7.  $ADE_{20}^{min} \downarrow / FDE_{20}^{min} \downarrow$  (top) and  $KDE \downarrow$  (bottom) on ETH/UCY for different augmentation and training strategies.

Method	ETH	Hotel	Univ	Zara1	Zara2	Avg
w/ pretrained recon.	0.50/0.83 2.845	0.13/0.22 -0.140	0.29/0.52 1.203	0.22/0.42 0.474	0.17/0.32 -0.412	0.26/0.46 0.794
w/ initial aug.	0.5/0.84 3.684	<b>0.13/0.20</b> -0.440	0.29/0.52 1.198	<b>0.21/0.39</b> 0.370	0.17/0.30 -0.269	0.32/0.45 0.909
w/ linear aug. 1	0.50/0.84 3.468	0.13/0.21 -0.412	0.30/0.53 1.207	0.22/0.40 0.520	0.17/0.30 -0.530	0.26/0.46 0.851
w/ linear aug. 2	0.48/0.8 3.150	0.13/0.21 -0.464	0.29/0.53 1.185	0.21/0.40 0.398	0.17/0.30 -0.230	<b>0.25/0.45</b> 0.808
w/ social force aug.	0.50/0.80 2.948	0.13/0.20 -0.478	0.3/0.55 1.299	0.21/0.39 0.421	<b>0.16/0.30</b> -0.317	0.30/0.45 0.775
Ours	<b>0.46/0.75</b> 2.789	<b>0.12/0.20</b> -0.531	<b>0.28/0.51</b> 1.031	<b>0.21/0.38</b> 0.344	<b>0.16/0.29</b> -0.704	<b>0.25/0.43</b> 0.586

social used here, as discussed in Sec. 3.1.

As depicted earlier in Figure 2 the generated pseudo-trajectories evolve alongside the rest of the framework and are fed back to the model during subsequent training cycles. Here, we aim to investigate the impact of this strategy based on several variants of our model. First instead of allowing the social reconstructor to continue to train alongside the rest of the model, we pre-train and freeze it in our framework. We refer to this variant as ‘w/ pretrained recon’. Next, in the second variant, we train and then place the social reconstructor outside of our model as a serial data augmentation module. This variant is referred to as ‘w/ initial aug’ in the table. We then explore three additional variants where we augment the data and train the social forecaster using the augmented data. We extrapolate the trajectories linearly based on the first two or the last two timesteps in the past horizon. We call these two augmentation methods ‘linear aug 1’ and ‘linear aug 2’, respectively. As the final augmentation, we generate samples inspired by the social force model [7]. We add an equal number of samples for all augmentations as our original model (ETH: 1,616, Hotel: 1,493, Univ: 1,277, Zara1: 1,365, Zara2: 370). This investigation shows that the  $ADE_{20}^{min}/FDE_{20}^{min}$  and KDE values deteriorate as a result of the mentioned modifications further demonstrating the benefits of our strategy in co-training the social reconstructor and

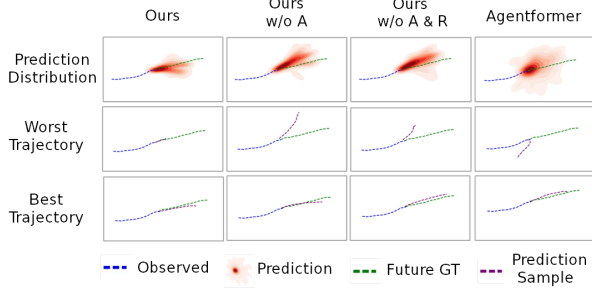


Figure 3. Prediction results for our method compared to two ablated models as well as Agentformer on two examples of the ETH scene. Past and future ground truth trajectories are shown in blue and green dashed lines, while the prediction samples are illustrated with purple dashed lines. ‘Ours w/o A’ indicates our method without Augmentations. ‘Ours w/o A and R’ is our method without the augmentations and social reconstructor.

forecaster, and re-using the augmented trajectories during training.

**Discussion.** We provide a qualitative comparison of our method with two ablated versions as well as Agentformer on two example scenes from the ETH test set in Figure 3. The second column shows our proposed method without augmented data, while the third presents an ablation without both augmentation and the social reconstructor. Looking at the prediction distribution, our method performs more accurately and with less variance among the predicted trajectories compared to other ablated varieties. The accuracy of the distribution also shows itself in the comparison among worst trajectories. The worst trajectories in our method are the best among all the ablations, while the best trajectories are comparable. This also indicates a limitation of the  $ADE_{20}^{min}/FDE_{20}^{min}$  evaluation metric, which only favors the best results. Our method also outperforms Agentformer qualitatively when comparing the prediction distributions and the worst trajectories, for which Agentformer produces more disperse and less directionally focused trajectories. More examples and comparison with an ablation of our method without the social loss are included in Appendix A.3.

To investigate the effect of the minimal accepted distance between pedestrians  $\epsilon$  (Sec. 3.1), as well as the number of overlaps between predicted pedestrian locations, we executed a sensitivity analysis on ETH/UCY dataset, where we increased  $\epsilon$  from 0 to 2 meters by increments of 0.5. As illustrated in Figure 4, an increase from 0 to 0.5 does not result in a significant change in  $ADE_{20}^{min}/FDE_{20}^{min}$  and  $ADE_{20}^{mean}/FDE_{20}^{mean}$ . However, further increasing it results in a significant deterioration of the mentioned evaluation metrics. KDE also degrades and the overlap between forecasted pedestrians further decreases with an increase in  $\epsilon$ . Our social loss ensures that pedestrians maintain a minimum distance of  $\epsilon$ . As per Eqs. 5 and 8, using higher values for

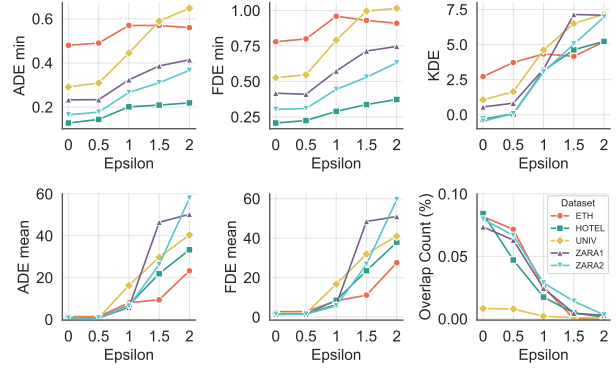


Figure 4. Sensitivity analysis on the value of  $\epsilon$  and its effect on six evaluation metrics:  $ADE_{20}^{min}$ ,  $FDE_{20}^{min}$ ,  $ADE_{20}^{mean}$ ,  $FDE_{20}^{mean}$ , KDE, and the overlap count percentage between pedestrians.



Figure 5. Examples of trajectory prediction with targets in close proximity of each other. Past and future timesteps are denoted by red and blue colors, respectively.

this hyperparameter will result in enforced distances between pedestrians in the scene, which can artificially increase the minimum separation between predictions, thus negatively affecting performance. Finally, to further study the impact of our social loss on the distance between pedestrians walking side-by-side, we explore a number of examples of such cases to identify any possible adverse effects. Figure 5 presents two examples, where we observe consistent predictions for the two target trajectories. Additional samples are provided in Appendix A.3.

## 6. Conclusion

In this paper, we introduced a pedestrian trajectory forecaster that uses a social loss to generate socially-informed pseudo-trajectories. Our proposed method uses a reconstructor to generate pseudo-trajectories which are used to augment the learning process. We have shown that injecting these pseudo-trajectories from the partially trained network improves results when compared to augmenting from a similar statically generated offline dataset. Our novel loss function decreases the number of overlaps between predicted pedestrian locations at future timesteps, while keeping the predictions accurate and increasing stability of predictions. Our experiments on standard benchmark datasets and metrics show our method to outperform existing state-of-the-art trajectory prediction methods.



## References

- [1] 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*, pages 961–971, 2016. **1**
- [2] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39, 2022. **1**
- [3] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018. **1**
- [4] Haleh Damirchi, Michael Greenspan, and Ali Etemad. Context-aware pedestrian trajectory prediction with multi-modal transformer. In *IEEE International Conference on Image Processing*, pages 2535–2539, 2023. **1**
- [5] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *International Conference on Image Processing*, pages 3688–3692, 2016. **1**
- [6] 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*, pages 2255–2264, 2018. **1, 2, 5, 6**
- [7] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. **3, 7**
- [8] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *IEEE/CVF International Conference on Computer Vision*, pages 2375–2384, 2019. **5**
- [9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. **6**
- [10] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664, 2007. **1, 5**
- [11] 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*, pages 15233–15242, 2021. **2, 6**
- [12] 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*, pages 759–776, 2020. **2**
- [13] Abdullallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020. **1, 2, 5, 6**
- [14] Abdullallah Mohamed, Deyao Zhu, Warren Vu, Mohamed Elhoseiny, and Christian Claudel. Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation. In *European Conference on Computer Vision*, pages 463–479, 2022. **1, 2, 5, 6**
- [15] Alessio Monti, Alessia Bertugli, Simone Calderara, and Rita Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *International Conference on Pattern Recognition*, pages 2551–2558, 2021. **5, 6**
- [16] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You’ll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE/CVF International Conference on Computer Vision*, pages 261–268, 2009. **3**
- [17] 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*, pages 452–465, 2010. **1, 5**
- [18] Amir Rasouli and Iuliia Kotseruba. Pedformer: Pedestrian behavior prediction via cross-modal attention modulation and gated multitask learning. In *IEEE International Conference on Robotics and Automation*, pages 9844–9851, 2023. **1**
- [19] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019. **3**
- [20] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, pages 549–565, 2016. **1, 5**
- [21] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, pages 549–565, 2016. **3**
- [22] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. **1**
- [23] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700, 2020. **2, 6**
- [24] Debaditya Shome, Pritam Sarkar, and Ali Etemad. Region-disentangled diffusion model for high-fidelity ppg-to-ecg translation. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 15009–15019, 2024. **1**
- [25] Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 660–669, 2020. **1**
- [26] Qiyue Sun and Yang Yang. Unsupervised video anomaly detection based on multi-timescale trajectory prediction. *Computer Vision and Image Understanding*, 227:103615, 2023. **1**
- [27] Luca Anthony Thiede and Pratik Prabhanjan Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. In *IEEE/CVF International Conference on Computer Vision*, pages 9954–9963, 2019. **5**

- [28] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022. [2](#)
- [29] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021. [1](#)
- [30] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137:109347, 2023. [1](#)
- [31] Bo Yang, Song Yan, Zheng Wang, and Kimihiko Nakano. Prediction based trajectory planning for safe interactions between autonomous vehicles and moving pedestrians in shared spaces. *IEEE Transactions on Intelligent Transportation Systems*, 2023. [1](#)
- [32] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *IEEE Robotics and Automation Letters*, 6(2):1463–1470, 2021. [1](#), [2](#)
- [33] Sihao Yu, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Zizhen Wang, and Xueqi Cheng. A re-balancing strategy for class-imbalanced classification based on instance difficulty. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 70–79, 2022. [5](#)
- [34] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021. [1](#), [2](#), [3](#), [5](#), [6](#)
- [35] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904, 2021. [2](#)