Supplementary Material
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

A.1. Pseudocode

Our proposed method is detailed in Algorithm 1. We train our model for N_T epochs initially. During this warm-up period, we also record the values of the loss L_F for each sample i and epoch e. After this period, we calculate $Count$ for each sample i and determine their inclusion as a pseudo-trajectory if $Count(i) < D \times N_c$. Here, N_c denotes the number of epochs where the loss for each sample has been recorded. At the end of warm-up period $N_c = N_T$, while after the warm-up period $N_c = N_{Int}$, where N_{Int} is the epoch interval between pseudo-trajectory generations. To generate the final augmented samples we concatenate the reconstructed past timestep $\tilde{S}_p$ and the social forecaster output trajectory $SF(\tilde{S}_p)$. Prior to each augmentation, we erase the previously added trajectories from the training data.

A.2. Training and Architectural Details

Masking. To mask each scene, we calculate the number of total timesteps $T_{scene} = N \times t_p$. The number of masked timesteps can be calculated as $R \times T_{scene}$ for masking ratio R. We observed that masking a timestep solely by setting it to location zero was confusing to the model, as it would get interpreted as a non-masked zero location. For this reason, we concatenated a binary indicator with the masked input location $S_p^{masked}(t)$ for each timestep t, where 0 indicates no masking, and 1 indicates masking.

Hyperparameters The hyperparameters that we used to train our models with are depicted in Table A.1. We observed that the Univ dataset was sensitive to overfitting, due to a higher number of test samples compared to the train samples, as well as the difference between the fewer number of crowded scenes in the train partition compared to the larger number in the test partition. To effectively address this, the size (number of parameters) of the model was reduced for this dataset, by reducing the values of the hyperparameters d_m and d_{ff}, as shown in the first two rows of Table A.1. For the learning rate schedule, We used the Steplr scheduler, which has the two hyperparameters of gamma and stepsize as depicted in Table A.1. We provide the code for our paper at https://github.com/thisishale/SocRec.

Table A.1. Hyperparameters of our method for ETH/UCY and SDD.

Hyper-Params	Dataset						Description
	ETH	Hotel	Univ	Zara1	Zara2	SDD	
d_m	128	64	64	256	128	128	Model dimension
d_{ff}	512	256	128	512	512	256	Feedforw. layer dim.
d_z	32	32	32	32	32	32	Latent space dim.
n_{enc}^f	1	2	2	1	2	1	Encoder layers
n_{dec}^f	1	1	1	1	1	1	Decoder layers
n_{dec}^r	1	1	1	1	1	1	Recon. decoder layers
$n_{atthead}$	8	8	8	8	8	8	Attention heads
D	0.5	0.5	0.5	0.5	0.5	0.5	Difficulty threshold
ϵ	0.1	0.1	0.05	0.1	0.1	0.1	Epsilon for social loss
N_T	10	20	20	20	20	10	Threshold epoch
N_{Int}	10	10	10	10	10	10	interval epochs
R	30	10	10	30	20	10	Masking ratio
$gamma$	0.8	0.8	0.8	0.5	0.8	0.8	Steplr scheduler Gamma
lr	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	Learning rate
$stepsize$	10	20	20	10	40	10	Step size for scheduler
w_1	1	1	1	1	1	1	Forecaster loss weight
w_2	1	1	1	1	1	1	Recon. loss weight
w_3	1	1	1	1	1	1	Social loss weight

A.3. Additional Visualizations and Results

In this section, we provide visualizations of forecasted trajectories for four examples shown in Figure A.1 to illustrate the effect of social loss on the number of location overlaps. According to the standard protocol, trajectories are included within each scene only for those pedestrians whose trajectories (combining both ground truth past and future locations) comprise a length of 20 timesteps. For example, in Example 1, there are two such pedestrians, in Example 2, there are four, etc. The minimum distance be-

Algorithm 1 Training of our proposed method

N_{Tot}: Total number of epochs, N_T: Threshold epoch,
N_c: Loss observation duration, N_{Int}: Interval epoch,
N_m: Number of Training samples, N_a: Number of Augmented samples,
SF: Forecaster module, SR: Reconstructor module,
S_p: Past trajectory, S_p^{masked}: Masked past trajectory,
S_f: Future ground truth trajectory,
L_F: Forecaster CVAE loss function,
L_R: Reconstructor VAE loss function,
$\mathcal{L}_{Total}$: Total loss, L_{Soc}: Social loss function,
$l_{arr} \in \mathbb{R}^{N_m \times N_c}$:Array to save losses,
D: Difficulty Threshold,
$A_{arr} \in \mathbb{R}^{N_a}$: Array to save Augmented samples,

while $e < N_{Tot}$ **do**
  while $i < N_m$ **do**
    $\tilde{S}_f \leftarrow SF(S_p)$;
    $\tilde{S}_p \leftarrow SR(S_p^{masked})$;
    Calculate L_F, L_R, L_{Soc}, $\mathcal{L}_{Total}$ Compute gradients and backpropagate $\mathcal{L}_{Total}$
    $l_{arr}[i, e] \leftarrow L_F$
    if $e = N_{Thr}$ **then**
      $Count(i) \leftarrow \sum_{e=1}^{N_{thr}} \mathbb{I}(d_{i,e} > a_{i,e})$
      if $Count(i) < D \times N_c$ **then**
        $A_{arr} \leftarrow \tilde{S}_p \oplus SF(\tilde{S}_p)$
      end
      $N_{thr} \leftarrow N_{thr} + N_{Int}$
    end
    $i \leftarrow i + 1$
  end
  if $e = N_{Thr}$ **then**
    Erase previously added augmented smaples from training set
    Add A_{arr} samples to the training set
    Clear A_{arr}
    Clear l_{arr}
  end
  $e \leftarrow e + 1$
end

tween pairs of pedestrian trajectories in the scene is depicted above each example. Overlaps between pedestrians, where their separation within a timestep is smaller than $\epsilon \leqslant 0.1$, are highlighted with red circles. As shown in the figure, our proposed method provides improved socially-aware predictions, where the forecasted trajectories have a lower chance of overlapping with each other. We also investigate the effect of social loss on ADE_{20}^{mean} and FDE_{20}^{mean}, which is shown in Table A.2. Our method results in better performance regarding the mean error of the produced trajectories in four

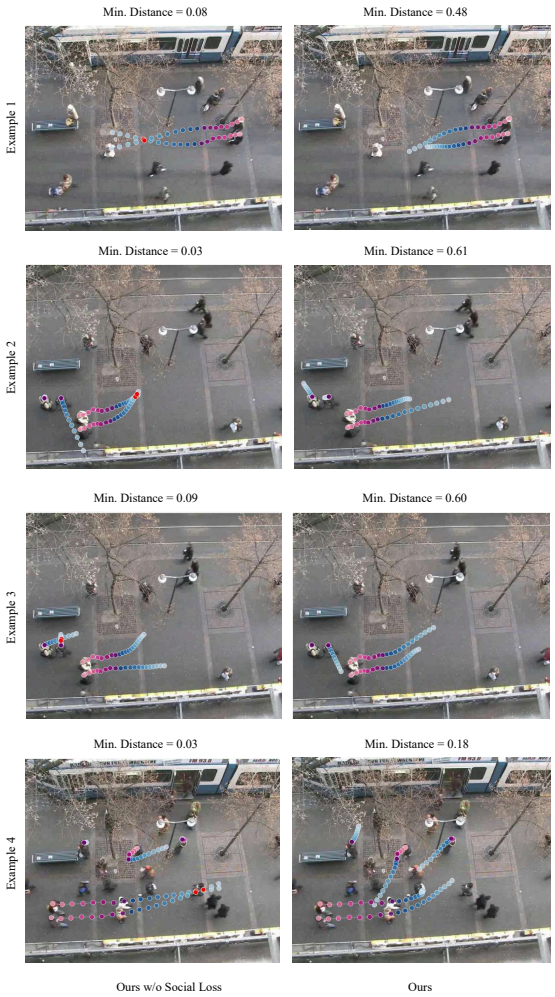

Figure A.1. Visualization of trajectories in a scene from Hotel subset. Red circles show location overlaps. Min. Distance denotes the minimum euclidean distance between pedestrians in meters. Past and future timesteps are denoted by red and blue, respectively.

Table A.2. Ablation studies for $\text{ADE}_{20}^{mean} \downarrow / \text{FDE}_{20}^{mean} \downarrow$.

Social Attention	Social Loss	Dataset ETH	Hotel	Univ	Zara1	Zara2
✓	✓	**1.27/2.61**	**0.57/1.33**	0.86/1.89	**0.70/1.52**	**0.59/1.34**
✓	✗	1.37/2.81	0.64/1.40	0.89/1.90	0.75/1.63	0.68/1.51
✗	✗	1.38/2.78	**0.57**/1.21	**0.82/1.80**	0.79/1.73	0.68/1.52

out of five subsets.

Two examples demonstrating the best, worst, and distribution of predicted trajectories by our proposed method, three ablated versions of our method, and Agentformer [1] are illustrated in Figure A.2. We observe that our method produces less dispersed distributions, as well as better 'best case' and more viable 'worst case' predictions. Additionally,

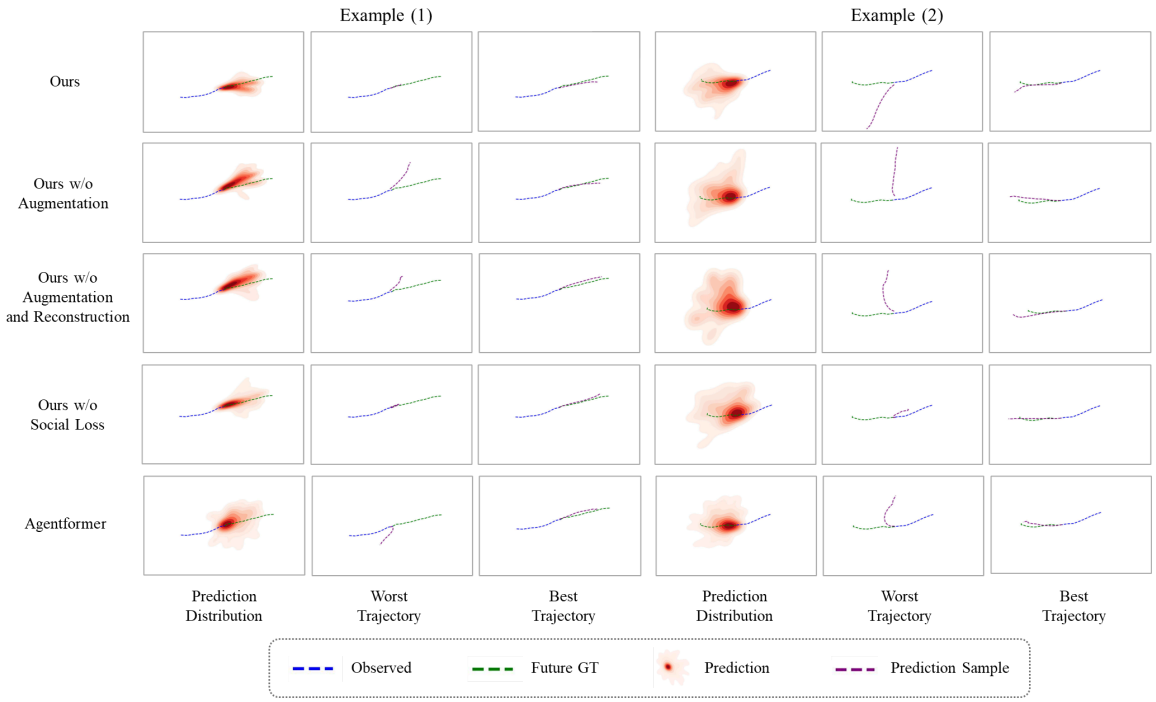

Figure A.2. Prediction results for our method compared to three ablated models as well as Agentformer on two examples of the ETH scene. Past and future ground truth trajectories are shown in blue and green dashed lines, while the prediction samples are illustrated with purple dashed lines. We observe that our proposed method produces a less dispersed distribution compared to all the ablated versions as well as Agentformer. Our proposed method, compared to the others, also produces the closest 'worst trajectories' to the ground truth, while predicting comparable 'best trajectories' to others.

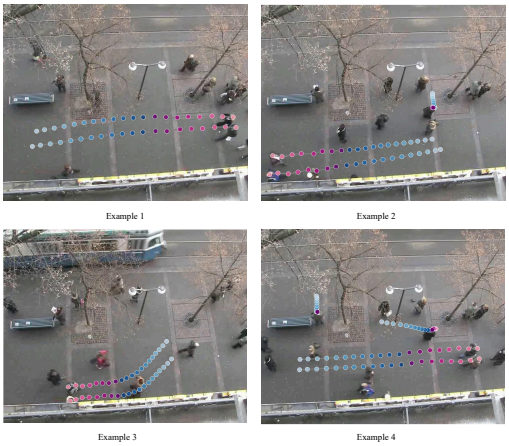

Figure A.3. Examples of trajectory prediction with targets in close proximity of each other. Past and future timesteps are denoted by red and blue, respectively.

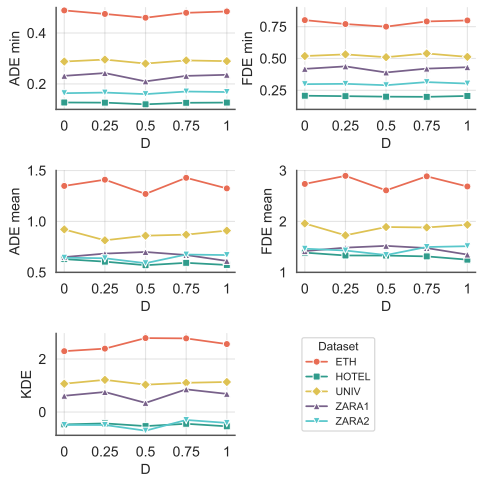

Figure A.4. Sensitivity analysis for D.

More examples illustrating our method's predictions in close proximity cases are shown in Figure A.3.

To analyze the effect of threshold D on the evaluation metrics, we perform a sensitivity analysis on this hyperparameter for different values between 0 and 1, where increasing D results in the inclusion of augmentations of easier samples in the training data. The results are depicted in Figure A.4, where we observe that $D = 0.5$ achieves the best overall

results by effectively balancing the inclusion and exclusion of augmented samples during training.

References

[1] 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. 2