Appendix

A. Definitions of Metrics and Attention Scores

Metrics. We evaluate prediction accuracy using the Average/Final Displacement Error (known as ADE and FDE) [1, 4]. Models are validated by the best metrics computed from 20 randomly generated trajectories for each case (*best-of-20*, *i.e.*, minADE$_{20}$ and minFDE$_{20}$). For agent i, we have

$$\text{minADE}_{20}\left(\mathbf{Y}^i, \left\{\hat{\mathbf{Y}}^i_k\right\}\right) = \min_k \frac{1}{t_f} \sum_{t=t_h+1}^{t_h+t_f} \|\mathbf{p}^i_t - \hat{\mathbf{p}_k}^i_t\|_2, \tag{1}$$

$$\text{minFDE}_{20}\left(\mathbf{Y}^i, \left\{\hat{\mathbf{Y}}^i_k\right\}\right) = \min_k \|\mathbf{p}^i_{t_h+t_f} - \hat{\mathbf{p}_k}^i_{t_h+t_f}\|_2. \tag{2}$$

Here, vectors with $_k$ come from the k-th prediction.

Attention Scores. We introduce the *Attention Scores* to quantitatively analyze how each SocialCircle partition relatively contributes to the final predicted trajectories. For the target agent i and the n-th partition, it is defined as the normalized squared sum of each $\mathbf{f}^i\left(\theta_n\right) \in \mathbb{R}^{d_{\text{sc}}}$. Formally,

$$\text{AttentionScore}(i, n) = \frac{\mathbf{f}^i\left(\theta_n\right)^\top \mathbf{f}^i\left(\theta_n\right)}{\sum_{m=1}^{N_\theta} \mathbf{f}^i\left(\theta_m\right)^\top \mathbf{f}^i\left(\theta_m\right)}. \tag{3}$$

The attention score evaluates the contribution of different partitions to the subsequent prediction network at the **feature level**, meaning that a partition with more neighbors may not directly lead to a higher score. It is obtained through the combined effect of multiple layers together during the training process, including the embedding layers g_{embed}, the fuse layer $\{\mathbf{W}_{\text{fuse}}, \mathbf{b}_{\text{fuse}}\}$, as well as the backbone prediction model B_{pred}. Thus, we choose this item to analyze how the SocialCircle contributes to the whole prediction model only *qualitatively*.

B. Additional Experimental Analyses on NBA SportVU Dataset

Due to the page limitations, we only report SocialCircle models' performances on ETH-UCY and SDD with both quantitative and qualitative results. This section further validates their detailed performance in handling different social interaction cases in the **NBA SportVU Dataset** by providing more additional qualitative results.

B.1. Dataset Configurations

The **NBA SportVU Dataset** [9] (short for **NBA** dataset) is made up of a large number of real-world trajectories of ten players plus a ball captured by the SportVU tracking system during several NBA games. The complex interactions between different players will pose significant challenges for trajectory prediction. Positions of all players and balls are labeled in foot (1 foot = 0.3048 meter).

Models	ADE (4.0s)	FDE (@2.0s)	FDE (@4.0s)
Social-LSTM[1]	1.79	1.53	3.16
S-GAN[4]	1.62	1.36	2.51
Social-STGCNN[12]	1.59	0.99	2.37
STAR[20]	1.26	1.28	2.04
PECNet[10]	1.83	1.69	3.41
NMMP[5]	1.33	1.11	2.05
GroupNet+NMMP[18]	1.25	1.08	1.80
GroupNet+CVAE[18]	**1.13**	**0.95**	1.69
MemoNet[19]	1.25	N/A	**1.47**
V^2-Net*[16]	1.28	0.96	1.68
V^2-Net-SC	1.22	**0.92**	**1.51**
E-V^2-Net*[17]	1.26	**0.93**	1.64
E-V^2-Net-SC	**1.18**	**0.90**	**1.46**

Table 1. Comparisons on NBA under *best-of-20* in meters. Lower ADE and FDE indicate better prediction performance. Models with "*" are reproduced under the same training settings.

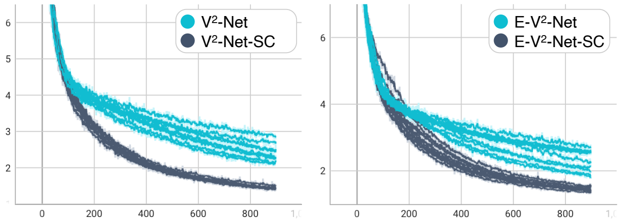

Figure 1. Loss curves (ℓ_2 loss at different training epochs) of different models at different training runs on NBA dataset. Curves are smoothed with the decay factor $= 0.8$.

Following the settings of [18, 19], we predict future $t_f = 10$ frames' trajectories based on the past $t_h = 5$ frames' observations. The sample interval between two frames is still set to $\Delta t = 0.4$s. Frames where the basketball is not on the court will be ignored. We randomly sample about 50K prediction cases (*i.e.*, 50K trajectories) from multiple games to validate models. Among these cases, 65% (about 32,500 samples) will be used for training, 25% (about 12,500 samples) for testing, and the remaining 10% for validation.

B.2. Baselines

We choose Social-LSTM[1], S-GAN[4], Social-STGCNN[12], STAR[20], PECNet[10], NMMP[5], GroupNet+NMMP[18], GroupNet+CVAE[18], MemoNet[19], V^2-Net*[16], and E-V^2-Net*[17] as our baselines on NBA dataset.

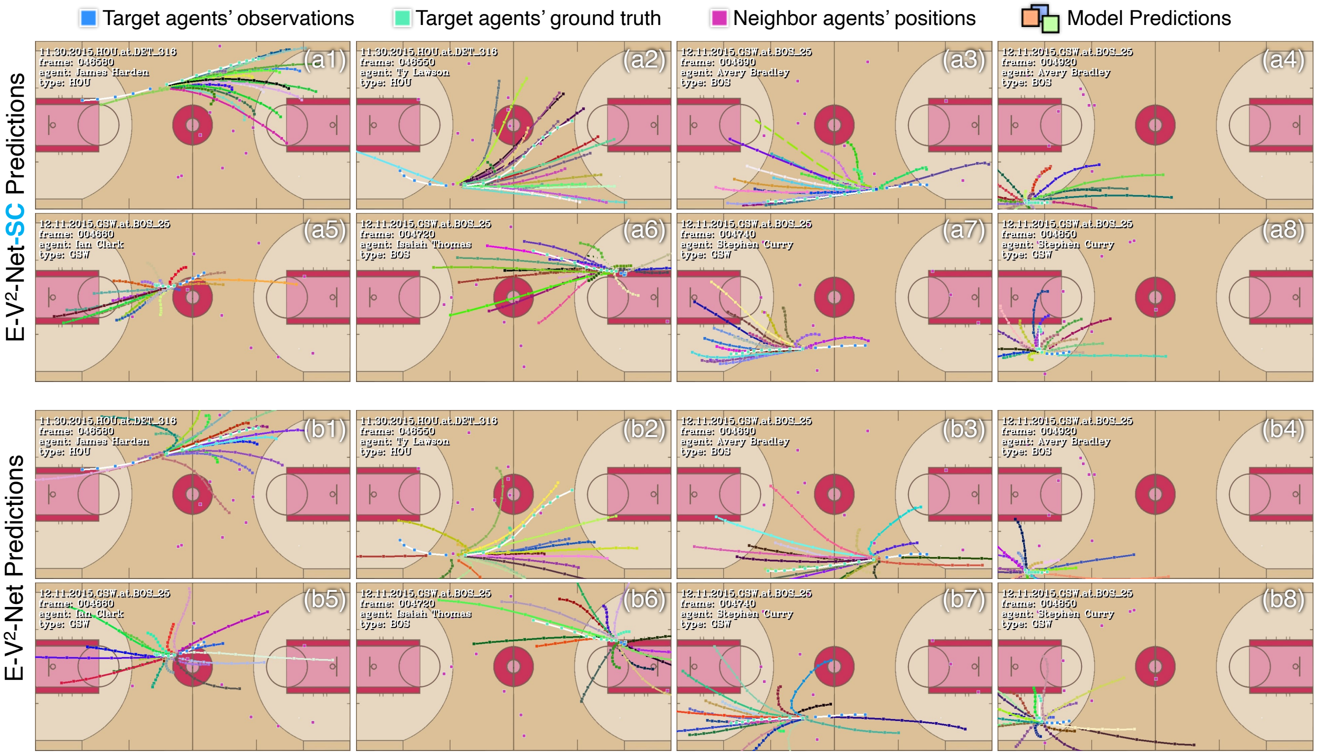

Figure 2. Visualized predicted trajectories provided by SocialCircle model E-V^2-Net-SC (subfigures (a1) to (a8)) and the original E-V^2-Net (subfigures (b1) to (b8)) on several NBA prediction scenes. Each sample includes 20 randomly generated trajectories.

B.3. Metrics

Except for ADE and FDE (minADE$_{20}$ and minFDE$_{20}$), following [18], we use the FDE-at-t-moment as a new metric to measure prediction performance. In detail, under the setting of $(t_h, t_f) = (5, 10)$ with sample interval $\Delta t = 0.4$s, the newly added metric FDE-at-5th-moment (minFDE$_{20}$@2.0s, short for FDE@2.0s) is defined as

$$\text{minFDE}_{20}(t) = \min_k \left\| \mathbf{p}_t^i - \hat{\mathbf{p}}_{kt}^i \right\|_2, \tag{4}$$

$$\text{FDE@2.0s} = \text{minFDE}_{20}(t = t_h + 5). \tag{5}$$

The original FDE can be treated as FDE@4.0s, *i.e.*,

$$\text{FDE@4.0s} = \text{minFDE}_{20}(t = t_h + 10). \tag{6}$$

B.4. Quantitative Analyses

Comparisons to State-of-the-Art Methods. As shown in Tab. 1, the SocialCircle model E-V^2-Net-SC has achieved competitive results. Compared with the GroupNet+CVAE that obtains the best ADE, E-V^2-Net-SC's ADE is not as well as that model (about 4.42% worse ADE), but its FDEs (both at 2.0s and 4.0s) are better than those for about 5.26% and 13.60%. In addition, even though the FDE@4.0s of MemoNet and E-V^2-Net-SC are at the same level (less than 1% differences), E-V^2-Net-SC outperforms the other for about 5.60% ADE. Although the original E-V^2-Net performs not as well as these newly published methods, the proposed SocialCircle makes it available to achieve competitive results.

Ablation Studies. We validate SocialCircle on two backbone models, V^2-Net and E-V^2-Net, and report their corresponding SocialCircle models' performance in Tab. 1. With the help of the proposed SocialCircle, both these models have achieved considerable quantitative performance gains. In detail, compared with the basic V^2-Net, V^2-Net-SC has achieved the 4.68% better ADE and the 10.11% better FDE (@4.0s). The E-V^2-Net-SC also outperforms E-V^2-Net for about 6.34% ADE and 10.97% FDE (@4.0s). These results indicate the quantitative effectiveness of the proposed SocialCircle for handling prediction cases with complex social interactions on NBA dataset.

B.5. Qualitative Analyses

Analyses of the Training Process. We visualize the loss (ℓ_2 loss) curves of V^2-Net, E-V^2-Net, and their SocialCircle models at multiple training runs on NBA dataset in Fig. 1. All these models are trained under the same settings. It shows that the loss values drop faster and finally become lower by introducing SocialCircle to baseline models. In addition, their loss values become more stable across different training runs compared to the original model. We can infer that the proposed SocialCircle may also play a normalization factor, thus reducing the influence of randomized training factors (such as the shuffle operation at each training epoch and the randomly sampled noise vectors to generate multiple predictions).

Visualizations of Social Behaviors. We visualize trajectories forecasted by the SocialCircle model E-V^2-Net-SC and the original E-V^2-Net in several NBA scenes in Fig. 2. These models do not take into account agents' categories (*i.e.*, players with different teams or basketball) when forecasting trajectories. For prediction scenes with different distributions of neighbor players, E-V^2-Net-SC's predictions present better interactive trends.

Comparing Fig. 2 (a1 to a4) and (b1 to b4), several trajectories predicted by the non-SocialCircle model (b1 to b4) have gone out of the court, while there are rarely these cases in the predictions of SocialCircle model (a1 to a4). It shows that SocialCircle models could learn players' different behavior patterns according to the SocialCircle, even though they do not know where the borders of the court are, thus making their predictions in line with the scene context.

In addition, the game-related interaction is a class of interactions specific to the NBA dataset, such as players carrying the ball on offense, switching from offense to defense, and many other interactive behaviors. Comparing Fig. 2 (a5 to a8) and (b5 to b8), we can see that SocialCircle could also better describe these interactive behaviors. For example, agent "Isaiah Thomas" moves from a complete standstill to start moving from the free throw lane during the observation period in case (a6). According to other players' status, the SocialCircle model finally provides predictions that seem like running to the frontcourt to start the offense. Unlike predictions shown in Fig. 2 (a6), trajectories predicted by the non-SocialCircle model appear very confusing, including both aggressive and defensive. Other game-interactive cases, like scoring in various ways in case (a7) and the flexible movements in case (a8), present similar trends, which indicates SocialCircle's capability to handle various social-interactive behaviors in different prediction scenes.

C. Additional Experimental Analyses on nuScenes Dataset

SocialCircle is proposed to handle interactions among pedestrians. In this section, we conduct a series of experiments on the nuScenes dataset [2, 3] to further validate how SocialCircles model interactions among vehicles as well as how they perform in traffic prediction scenes.

C.1. Dataset Configurations

The **nuScenes**[2, 3] is a large-scale real-world dataset of 1000 driving scenes collected in the urban cities of Boston and Singapore. Each scene has 20 seconds and is annotated at 2 fps. 850 scenes were manually annotated for 23 classes, such as pedestrians and vehicles, and included visibility, activity, and pose attributes. Note that only vehicles' 2D trajectories $\left\{\mathbf{p}_t^i\right\}_{i,t} = \left\{\left(x_t^i, y_t^i\right)\right\}_{i,t}$ are used in this paper. Following the settings of [6], we predict future $t_f = 12$ frames' trajectories according to vehicles' past $t_h = 4$ frames' observed trajectories. The sample interval between two adjacent frames is set to $\Delta t = 0.5s$. Since the annotations of the official 150 test sets are not available, following previous works like [14], we use 550 scenes to train, 150 scenes to validate, and the other 150 scenes to test.

Models	ADE$_5$	FDE$_5$	ADE$_{10}$	FDE$_{10}$
Trajectron++[15]	3.14	7.45	2.46	5.65
Y-net[11]	2.46	5.15	1.88	3.47
Agentformer[21]	1.59	3.14	1.30	2.47
MUSE-VAE[6]	1.38	2.90	1.09	2.10
E-V^2-Net*[17]	1.46	3.18	1.15	2.37
E-V^2-Net-SC	1.44	3.10	1.13	2.30

Table 2. Comparisons on nuScenes under *best-of-5* and *best-of-10* in meters. Lower ADE and FDE indicate better prediction performance. Models with "*" are reproduced under the same settings.

C.2. Baselines

We choose Trajectron++[15], Y-net[11], Agentformer[21], MUSE[6], and E-V^2-Net* as our baselines on nuScenes.

C.3. Metrics

Following previous works like [6], we use both *best-of-5* and *best-of-10* validations to evaluate models' performance on nuScenes. Like the main paper, we denote these metrics as minADE$_5$/minFDE$_5$ and minADE$_{10}$/minFDE$_{10}$ (short for ADE$_5$/FDE$_5$ and ADE$_{10}$/FDE$_{10}$).

C.4. Quantitative Analyses

Tab. 2 reports the quantitative performance of several baseline models and the corresponding E-V^2-Net model. Although the base model (E-V^2-Net) is not specifically designed to predict trajectories in traffic scenes, SocialCircle still shows its capability to model interactions among vehicles. Compared to the vanilla E-V^2-Net, E-V^2-Net-SC has a 1.4% better ADE$_5$ and a 2.5% better FDE$_5$. The performance gain brought by the SocialCircle is more remarkable as the number of predicted trajectories rises from 5 to 10, including 1.7% on the ADE$_{10}$ and 3.0% on the FDE$_{10}$.

Although SocialCircle could help the base model E-V^2-Net to perform better, there are still noticeable differences in the performance between E-V^2-Net-SC and the MUSE-VAE that focus mainly on vehicle trajectory prediction, including 3.7% and 9.5% worse ADE$_{10}$ and FDE$_{10}$. It is

Variations	N_θ	ADE/FDE	Gain (%)
V^2-Net*	-	7.04/10.94	-4.92%/-2.63%
V^2-Net-SC-a4	1	6.96/11.05	-3.73%/-3.66%
V^2-Net-SC-a5	4	6.79/10.80	-1.19%/-1.31%
V^2-Net-SC	8	6.71/10.66	(base)
V^2-Net-SC-a6	12	6.65/10.60	+0.89%/+0.56%
V^2-Net-SC-a7	16	6.68/10.65	+0.45%/+0.09%
V^2-Net-SC-a8	36	6.64/10.64	+1.04%/+0.19%
E-V^2-Net*	-	6.73/10.75	-2.91%/-3.76%
E-V^2-Net-SC-a4	1	6.66/10.70	-1.83%/-3.28%
E-V^2-Net-SC-a5	4	6.61/10.55	-1.07%/-1.83%
E-V^2-Net-SC	8	6.54/10.36	(base)
E-V^2-Net-SC-a6	12	6.50/10.34	+0.61%/+0.19%
E-V^2-Net-SC-a7	16	6.46/10.22	+1.22%/+1.35%
E-V^2-Net-SC-a8	36	6.57/10.41	-0.46%/-0.48%

Table 3. Ablation studies on verifying the number of SocialCircle partitions N_θ with different backbone models on SDD. Values in the "Gain" column are the percentage ADE and FDE gain compared to the base 8-partition model (denoted with "(base)").

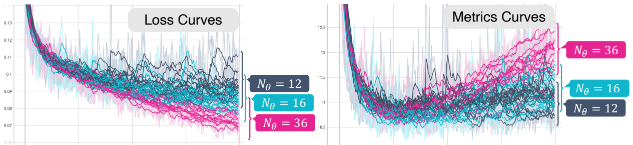

Figure 3. Loss curves (left, ℓ_2 loss) and metrics curves (right, ADE) of E-V^2-Net-SC variations a6 to a8 ($N_\theta \in \{12, 16, 36\}$).

worth noting that MUSE-VAE uses additional lane information to help predict better, whereas neither the base model E-V^2-Net nor the corresponding SocialCircle model E-V^2-Net-SC do not. This further inspires us to design meta components for the SocialCircle in traffic prediction scenarios.

D. Additional Experimental Analyses on the Number of SocialCircle Partitions

D.1. Quantitative Analyses

We run ablation experiments to validate how the number of SocialCircle partitions N_θ affects models' quantitative performance. In Tab. 3, 8-partition SocialCircle models perform the best, outperforming 4-partition variations for about 1.1% to 1.8% ADE and FDE. Especially, models with $N_\theta = 1$ work even worse, including up to 2.5% ADE drop compared to 4-partitions'. Comparing V^2-Net and V^2-Net-SC-a4, we find that the latter one even has about 0.1 pixels worse FDE. It aligns with our intuition that the more partitions the higher resolutions for describing social behaviors. While vice versa, too few partitions may lead to a coarse description of interactions, even mislead the model, thus significantly reducing prediction performance.

Note that due to the settings of predicting trajectories based on 8 historical observed frames on SDD, the maximum number of partitions is set to 8 to prevent unnecessary zero-paddings in trajectories' representations from pulling down the performance of the original backbone trajectory prediction network. To verify this thought, we expand the SocialCircle to make it available to handle $N_\theta > t_h$ cases by zero-padding trajectory representations (*i.e.*, the $\mathbf{f}^i_{\text{traj}}$ in Eq. (13)). Results of variations with postfixes {a6, a7, a8} reported in Tab. 3 are obtained under this new setting. In addition, we have attached the loss curves and metrics curves of these $N_\theta > t_h$ variations in Fig. 3. It shows that the loss may drop faster as the N_θ raises, but simultaneously exacerbates the risk of overfitting. We can further infer that even though a higher N_θ may provide better results, it also compresses the information in trajectories while reducing training stability. On balance, $N_\theta = 8$ may be a good compromise (ETH-UCY and SDD). As a result, we regard that N_θ should be no more than the t_h in the main paper.

D.2. Qualitative Analyses

Fig. 4 provides the visualized attention scores in different prediction cases on SDD-little0 with the $N_\theta = 4$ (subfigures (a1) to (a5)) and the $N_\theta = 8$ ((b1) to (b5)) E-V^2-Net-SC models. These two models are trained and validated under the same condition except for the N_θ.

Comparing Fig. 4 (a3) and (b3), the 8-partition model provides trajectories with different social behaviors for $\theta \in [1.5\pi, 2\pi)$, *i.e.*, partitions 7 and 8. In detail, predictions in partition-8 mostly try to avoid the right-coming neighbor, while predictions in partition-7 mostly walk as normal cases. For the 4-partition model's predictions in Fig. 4 (a3), predictions within the whole partition-4 all present the avoidance tendance, even though some predicted trajectories are far away from the existing neighbors. Similar cases also appear in cases (a2, partition-4) v.s. (b2, partitions 7 and 8) and cases (a5, partition-3) v.s. (b5, partitions 5 and 6). All these comparisons point out that a smaller number of SocialCircle partitions may lead to a coarser recognition and modeling of social behaviors, thus further causing misleading shifts in the predicted trajectories.

We also add manual neighbors to real-world prediction cases on SDD-little0 to validate both $N_\theta = 4$ and $N_\theta = 8$ E-V^2-Net-SC models' responses. As shown in Fig. 5, $N_\theta = 8$ model presents better spatial resolutions for handling social interactions. For example, compared to the $N_\theta = 4$ case (c2, partition-1), the corresponding $N_\theta = 8$ partition (c4, partition-2) has been less affected due to the manual neighbor. As a result, predictions in 8-partitions cases {(c4, partition-3), (c4, partition-4)} show different interactive trends. These results indicate that 8-partition SocialCircle models have better angular resolution to model potential social interactions as well as quantify their roles

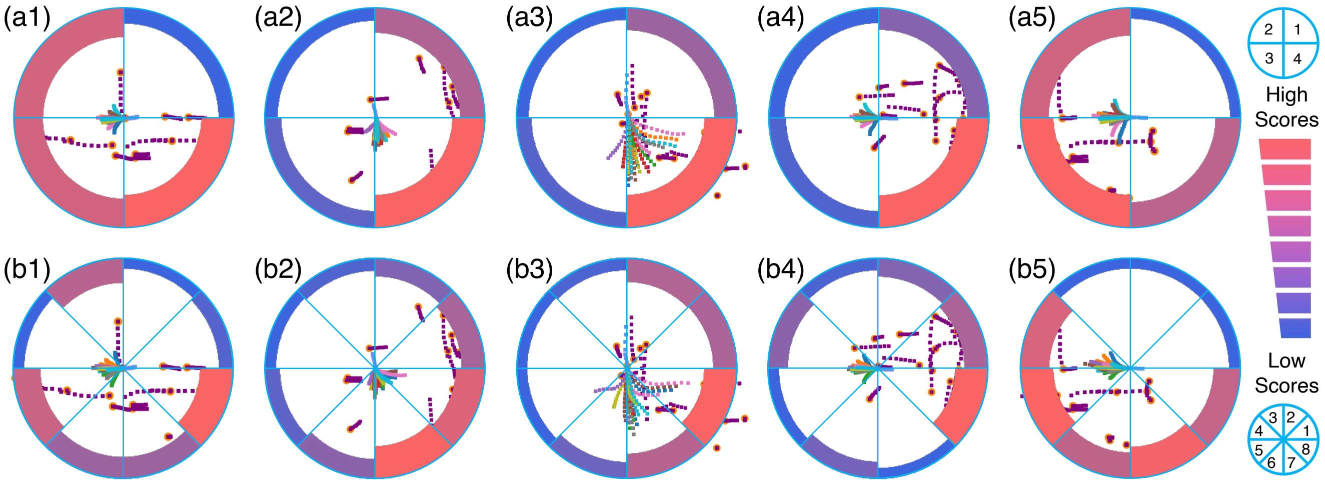

Figure 4. Visualized predicted trajectories and the corresponding attention scores of several real-world prediction cases on SDD-little0 provided by the **4-partition** E-V^2-Net-SC (a1) to (a5) and the **8-partition** E-V^2-Net-SC (b1) to (b5).

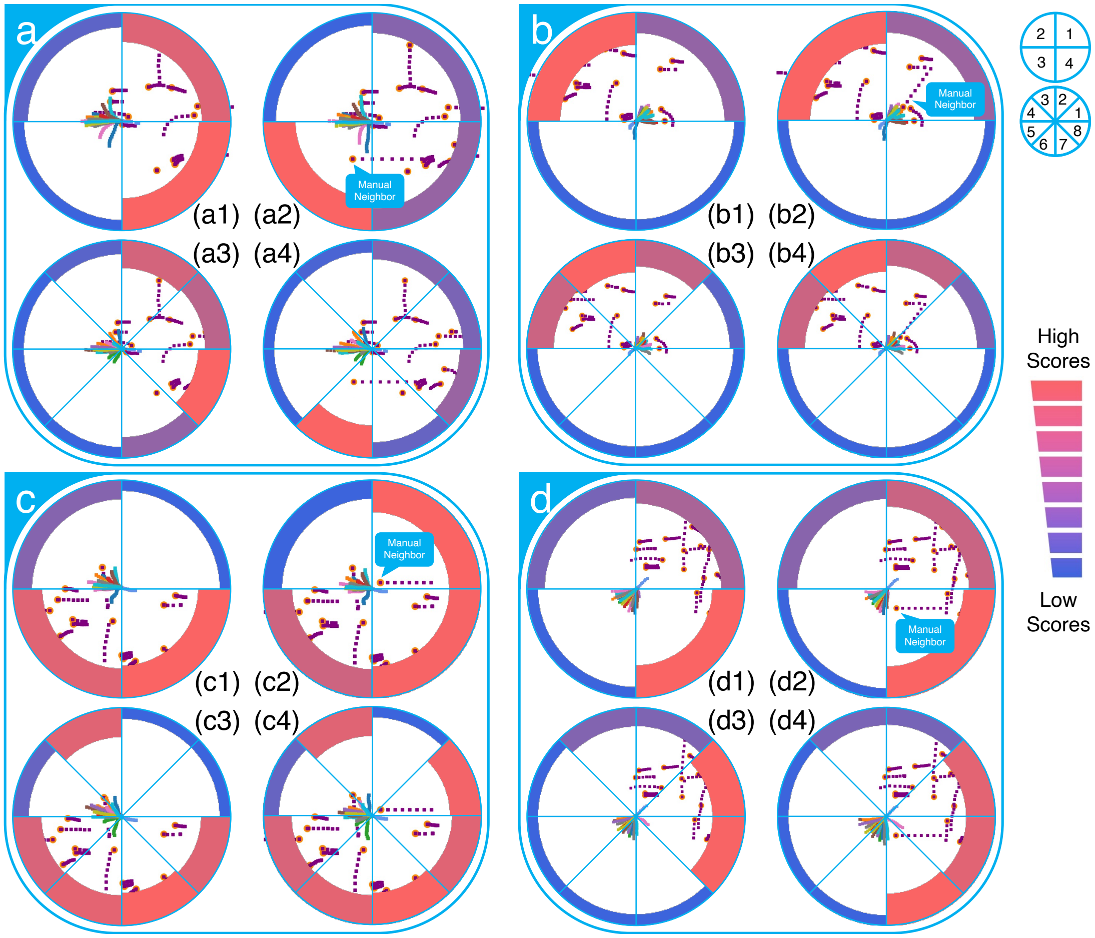

Figure 5. Visualized predicted trajectories and the corresponding attention scores of several real-world cases by adding additional manual neighbors. For each case $x \in \{\text{a, b, c, d}\}$, subfigure $(x1)$ is the **4-partition** $(N_\theta = 4)$ model's prediction, and $(x3)$ is **8-partition** $(N_\theta = 8)$ prediction. subfigures $(x2)$ and $(x4)$ are obtained by adding manual neighbors to cases $(x1)$ and $(x3)$, respectively.

Models	ADE/FDE ↓ (ETH-UCY)	Time ↓	Paras. ↓
Social-LSTM[1]	0.72/1.54	1180 ms	264K
SR-LSTM[22]	0.45/0.94	1179 ms	64.9K
PECNet[10]	0.29/0.48	607 ms	2.10M
Next[8]	0.46/1.00	114 ms	360.3K
S-GAN[4]	0.58/1.18	97 ms	46.3K
DAG-Net[13]	N/A	46 ms	2.35M
Social-STGCNN[12]	0.44/0.75	2.0 ms	7.6K
STC-Net[7]	0.38/0.68	1.3 ms	**0.7K**
V^2-Net*[16]	0.18/0.28	19 ms	1.91M
E-V^2-Net*[17]	0.17/0.28	21 ms	1.92M
V^2-Net-SC	0.17/0.27	23 ms	1.92M
E-V^2-Net-SC	0.17/0.27	24 ms	1.98M

Table 4. Comparisons of inference time and model parameters. Results are obtained from [7] on one NVIDIA GeForce GTX 1080Ti card. Models with "*" are reproduced with PyTorch.

in modifying forecast results.

Model	Inference time @batchsize					Parameters
	1	50	100	500	1000	
V^2-Net	28	30	31	38	81	1,911,264
V^2-Net-SC	34	35	36	55	88	1,923,936
E-V^2-Net	28	33	37	67	112	1,976,864
E-V^2-Net-SC	34	39	43	73	119	1,989,536

Table 5. Inference times (in milliseconds) at different batch size settings (from 1 to 1000) and the number of trainable parameters of V^2-Net, E-V^2-Net, and their corresponding SocialCircle models. Results are obtained by running models (PyTorch) on one Apple Mac mini (M1, 2020) with 8GB memory.

E. Parameters and Inference Times

Comparisons with Other Baselines. We compare the inference speed and the number of parameters of different models in Tab. 4. All results are measured on one NVIDIA GeForce GTX 1080Ti GPU (short for "1080Ti"). Since the official codes of V^2-Net and E-V^2-Net are implemented with TensorFlow and run slowly in our Python environment on the server, we reproduce their codes with PyTorch and report their running time (batch size is set to 1, marked with "*") in Tab. 4. From these results we can see that the SocialCircle itself would not lead to a large number of computations and extra trainable variables. Compared to the original models, the inference times of their corresponding SocialCircle models are still considerable.

Further Discussions on the Inference Speed. Considering that the platform on which trajectory prediction models are running may not be equipped with high-performance computing devices, all results reported in Tab. 5 are obtained on one Apple Mac Mini with an Apple M1 chip (8GB memory), which performs similarly to current iPhones and iPads. Additionally, several researchers like [7] have defined the *low-latency trajectory prediction*, which indicates that the trajectory prediction method should predict trajectories within the sampling interval to achieve the real-time prediction goal. For example, when predicting trajectories on ETH-UCY with a sample rate of 2.5 fps, the implementing time of the model should be less than 400 ms. Results in Tab. 5 show that the proposed methods could meet the low-latency standard even when running on the Apple M1 chip, indicating their potential to be applied to complex application scenarios.

F. Additional Visualized Toy Examples

To demonstrate the effectiveness of the proposed SocialCircle in handling different social interaction cases, following the settings in Section 4.3 **Toy Examples I (Social Interactions)**, we provide more visualized toy examples in the real-world UCY-zara1 prediction scenes in this section. In these toy examples, we add one manual neighbor to each prediction case, thus visualizing how SocialCircle modifies the original predicted trajectories under different interaction contexts.

In the main paper, we use a simple linear interpolation method to simulate manual neighbors' trajectories. For agent i, given two points $\mathbf{p}_0^i$ and $\mathbf{p}_{t_h}^i$ $(1 \leq t \leq t_h)$, the linearly-interpolated coordinate $\mathbf{p}_t^i$ is computed via

$$\mathbf{p}_t^i = \mathbf{p}_0^i + \frac{\mathbf{p}_{t_h}^i - \mathbf{p}_0^i}{t_h} t. \tag{7}$$

Fig. 6 includes more visualized predictions under different linearly interpolated manual neighbor settings. We also designed a non-linear interpolation method to further validate SocialCircle's capability, which linearly interpolates the velocity from each adjacent two of the three given points to generate manual neighbors with curved trajectories via

$$\mathbf{v}_t^i = \mathbf{p}_t^i - \mathbf{p}_{t-1}^i, \tag{8}$$

$$\mathbf{v}_t^i = \mathbf{v}_0^i + t\Delta\mathbf{v}, \tag{9}$$

$$\sum_{t=1}^{t_h} \mathbf{v}_t^i = \mathbf{p}_{t_h}^i - \mathbf{p}_0^i. \tag{10}$$

Thus, $\Delta\mathbf{v}$ can be represented as

$$\Delta\mathbf{v} = \frac{2(\mathbf{p}_{t_h}^i - \mathbf{p}_0^i - \mathbf{v}_0^i t_h)}{t_h(t_h+1)}, \tag{11}$$

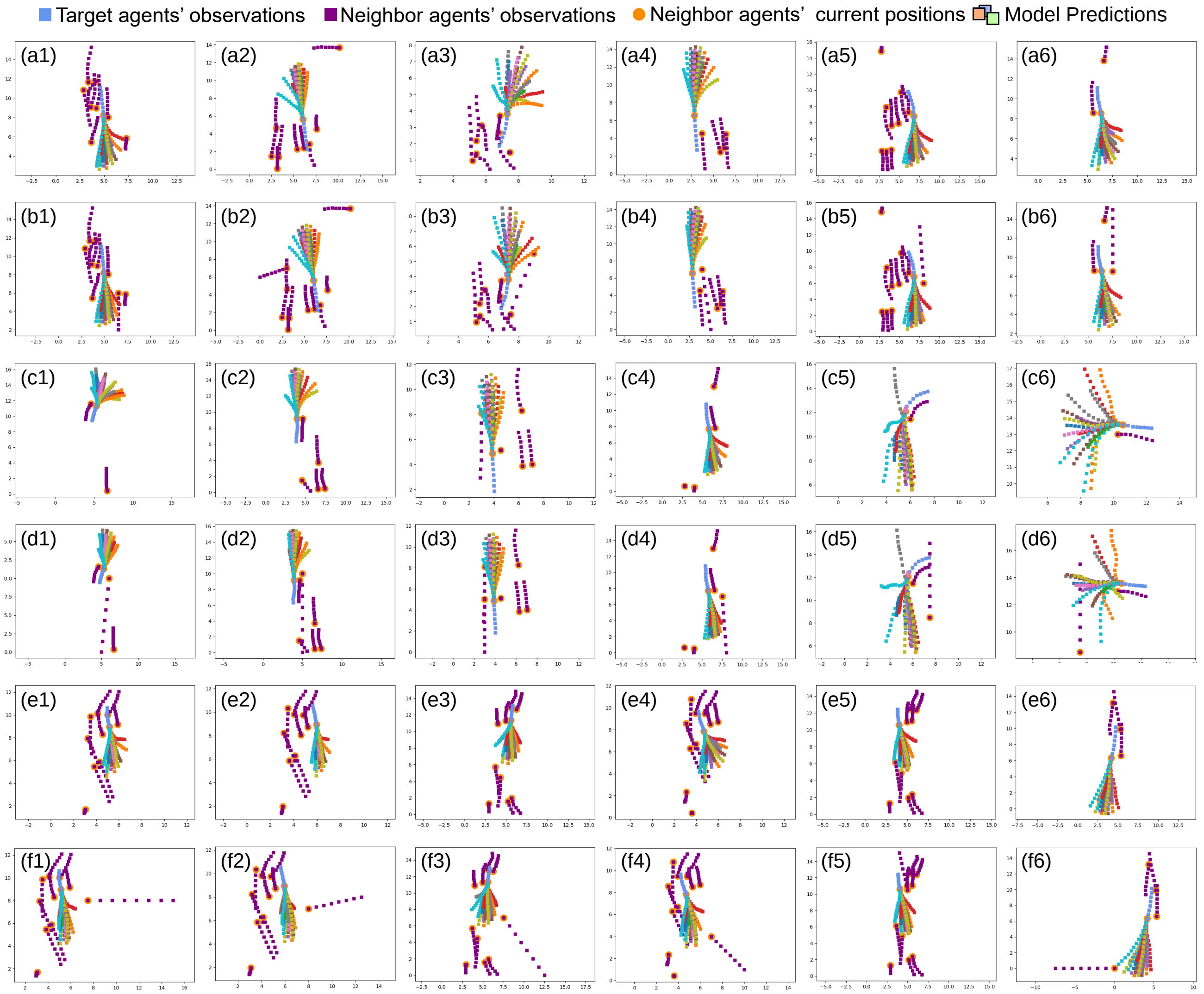

Figure 6. Toy examples (linear interpolation) on validating the effectiveness of the overall modification of social interactions. We add manual neighbors to the original ETH-UCY prediction scenes and visualize how they change the predicted trajectories. Prediction case in subfigure (xn), where $x \in \{\text{a, c, e}\}, n \in \{1, 2, 3, 4, 5, 6\}$, represents the original prediction scene in UCY-zara1, and the corresponding $(yn, y \in \{\text{b, d, f}\})$ case represents prediction considering the manual neighbor.

and we can finally determine the coordinate $\mathbf{p}_t^i$ at any moment t. Formally,

$$\mathbf{p}_t^i = \mathbf{p}_0^i + \sum_{n=1}^{t} n\Delta\mathbf{v}. \qquad (12)$$

These trajectories and the corresponding SocialCircle predictions are shown in Fig. 7. In both figures, we observe that after adding manual neighbors with a certain velocity around the target agent, its new predicted trajectories tend to keep a certain *social distance* to the manual neighbor in most cases. For example, in Fig. 6 (b2, b3, b4) and Fig. 7 (b1, b3, b4), the target agents are predicted to move away from the manual neighbors dramatically. In some cases, like Fig. 6 (d6, f1) and Fig. 7 (b5, b7), the originally predicted trajectories of the target agent before adding the manual neighbor have already demonstrated a strong trend of movement toward certain destinations. Among these cases, if we add a manual neighbor that also moves toward such a destination with a relatively fast velocity, the newly predicted trajectories of the target agent may change heavily to avoid possible collisions or keep certain social distances with the manual neighbor.

Unlike these situations, Fig. 6 (f6) and Fig. 7 (b2), represent a different way to handle interactions in which the predicted trajectories have shifted to the left to avoid the fast-moving manual neighbor coming from the left side, rather

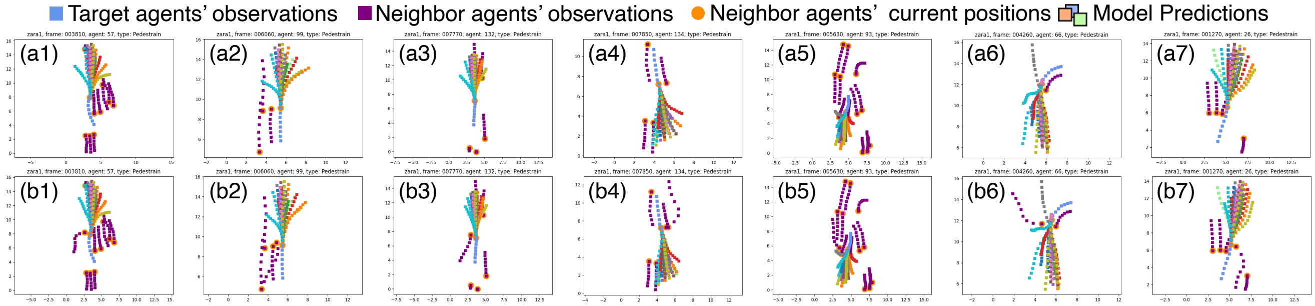

Figure 7. Toy example (linear-velocity interpolation) on validating social interactions. Compared to the linearly-interpolated trajectories, we add several non-linear patterns to the trajectories of manual neighbors to further reflect their fine-level motions. The prediction case in subfigure (an), where $n \in \{1, 2, 3, 4, 5, 6, 7\}$, represents the original prediction scene in UCY-zara1, and the corresponding (bn) case represents prediction considering the curved-moving manual neighbor.

Variations	V D R mR	ADE/FDE	Drop (%)
E-V^2-Net*	× × × ×	6.73/10.75	-2.91%/-3.76%
E-V^2-Net-SC	✓ ✓ ✓ ×	6.54/10.36	(base)
E-V^2-Net-SC-4f	✓ ✓ ✓ ✓	6.84/10.94	-4.59%/-5.60%

Table 6. Ablation studies on validating the movement direction ("mR") factor on SDD. "V", "D", and "R" represent current velocity, distance, and direction factors. Values in "Drop" are the percentage matrices drop compared to the base model.

than shifted to the right side. These phenomena demonstrate that the SocialCircle models could dynamically handle different interactive contexts in different prediction scenes, thus providing trajectories in line with social rules. In short, the three meta components (velocity, distance and direction) used in SocialCircle have the potential to reflect different interactive contexts and further promote the prediction networks to learn to generate divergent trajectories.

However, we also observe that there exist some cases in which predictions do not comply with interactive contexts. In Fig. 6 (d3), SocialCircle model still remains the way it forecasts trajectories for the target agent even after adding a near enough manual neighbor with a relatively fast velocity. In Fig. 6 (d2), after adding the fast-moving manual neighbor on the right side, the left part of the predicted trajectories are pruned off. Although the quantitative prediction performance has not been influenced, it actually constrains the diversity of the predicted trajectories. Therefore, the three meta components (velocity, distance and direction) used in SocialCircle are still worthy of further studies to simulate and forecast in more complex interactive cases.

G. Further Discussions on Limitations

As mentioned in the "Limitations" section, neighbor agents' movement directions have not been considered in the proposed SocialCircle. This section further discusses whether the movement direction factor should be considered as one of the SocialCircle meta components.

G.1. Limitation Analysis

As shown in Fig. 8, we conducted another toy experiment to show models' responses to the manual agent with different movement directions. In all 3-factor cases (a2) to (a5), the SocialCircle model forecasts almost the same trajectories (except for the noise factor for random generation). It is worth noting that the predictions in case (a3) are relatively "dangerous", for there might be potential collisions or too-close social distances with the manual neighbor.

From the point of view of network training, we can simply understand that the whole prediction network forecasts an "average" trajectory to satisfy all these training samples with the same SocialCircle but move in different directions. As a result, it may predict trajectories with avoidances for the neighbors that may not collide with the target agent (like Fig. 8 (a5)), or may still collide with others (like Fig. 8 (a3)).

It should be noted that these extreme cases in the toy experiments are rarely seen in real-world prediction scenarios. In most ETH-UCY and SDD scenes, SocialCircle models still work as expected. Nevertheless, these few uncovered social interaction cases still indicate their limitations, although they have achieved better quantitative performance.

G.2. The Movement Direction Factor.

Following the "lite-rules" assumption, we attempt to add the movement direction factor to provide detailed interactive information. It is defined as the average of each neighbor's moving direction located in some partition. Formally,

$$\mathbf{f}^i_{\text{mdir}}\left(\theta_n\right) = \frac{1}{|\mathbf{N}^i(\theta_n)|} \sum_{j \in \mathbf{N}^i(\theta_n)} \operatorname{atan2}\left(f_{\text{2D}}\left(\mathbf{p}^j_{t_h} - \mathbf{p}^j_1\right)\right). \tag{13}$$

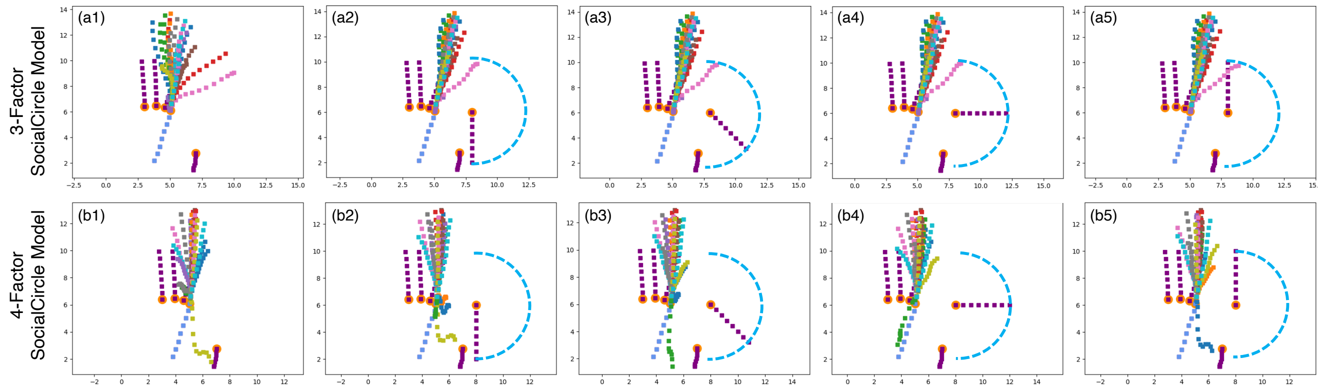

Figure 8. Visualized E-V^2-Net-SC predictions with manual neighbors with different movement directions. In this toy experiment, we set $d_m = 2.97$ and $v_m = 4.00$. (a1) to (a5) are predictions provided by the **3-factor** SocialCircle model, and (b1) to (b5) are predictions by **4-factor** model. Cases (a1) and (a5) are their original predictions without any given manual neighbors.

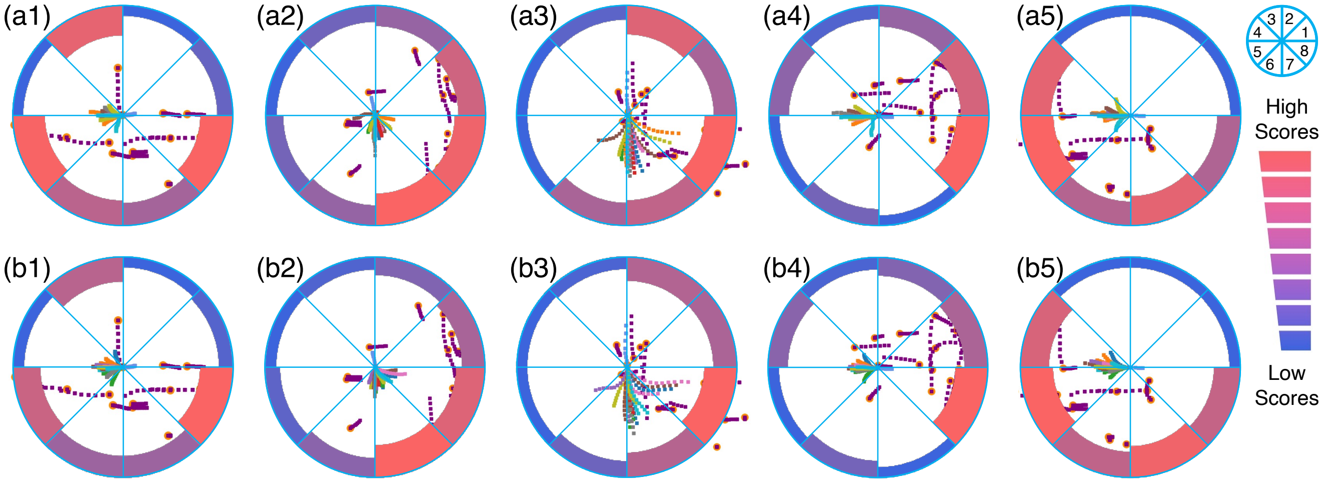

Figure 9. Visualized predicted trajectories and their corresponding attention scores in several real-world prediction cases (SDD-little0) provided by the **4-factor** E-V^2-Net-SC (a1) to (a5) and the **3-factor** E-V^2-Net-SC (b1) to (b5).

The corresponding 4-factor SocialCircle meta vector is

$$\mathbf{f}^i_{\text{meta}}(\theta_n) = \left(\mathbf{f}^i_{\text{vel}}(\theta_n), \mathbf{f}^i_{\text{dis}}(\theta_n), \mathbf{f}^i_{\text{dir}}(\theta_n), \mathbf{f}^i_{\text{mdir}}(\theta_n)\right)^\top. \tag{14}$$

G.3. Ablation Studies and Visualized Analyses of the Movement Direction Factor

Quantitative Analyses. We run experiments to quantitatively validate the usefulness of this movement direction factor on SDD, and their results are reported in Tab. 6. By adding this additional factor, the E-V^2-Net-SC-4f's performance drops significantly. Compared to the 3-factor E-V^2-Net-SC, it has 4.59% worse ADE and 5.60% worse FDE. Especially, its performance is even worse than the non-SocialCircle-model E-V^2-Net, which means that just adding such a simple new factor prevents other factors from expressing their contributions.

We infer that the movement direction factor brings more complex constraints to each prediction case, thus making the training process more difficult while reducing the model's generalization capability. In detail, the current three factors (velocity, distance, direction) are relatively "weak" rules to describe social interactions. Thus, the obtained SocialCircles could be similar even in different prediction cases. On the contrary, the movement direction factor varies from 0 to 2π for each neighbor in each partition, which brings extra "complexity" for each interactive case, thus further increasing the difficulty of model training in the case of the same network structure and training data.

Validation of Moving Directions. In Fig. 8 (b1) to (b5), we visualize the predicted trajectories provided by the 4-factor E-V^2-Net-SC corresponding to cases (a1) to (a5). We can easily see that predictions in cases (b2) to (b5) are different due to the various moving directions of the given

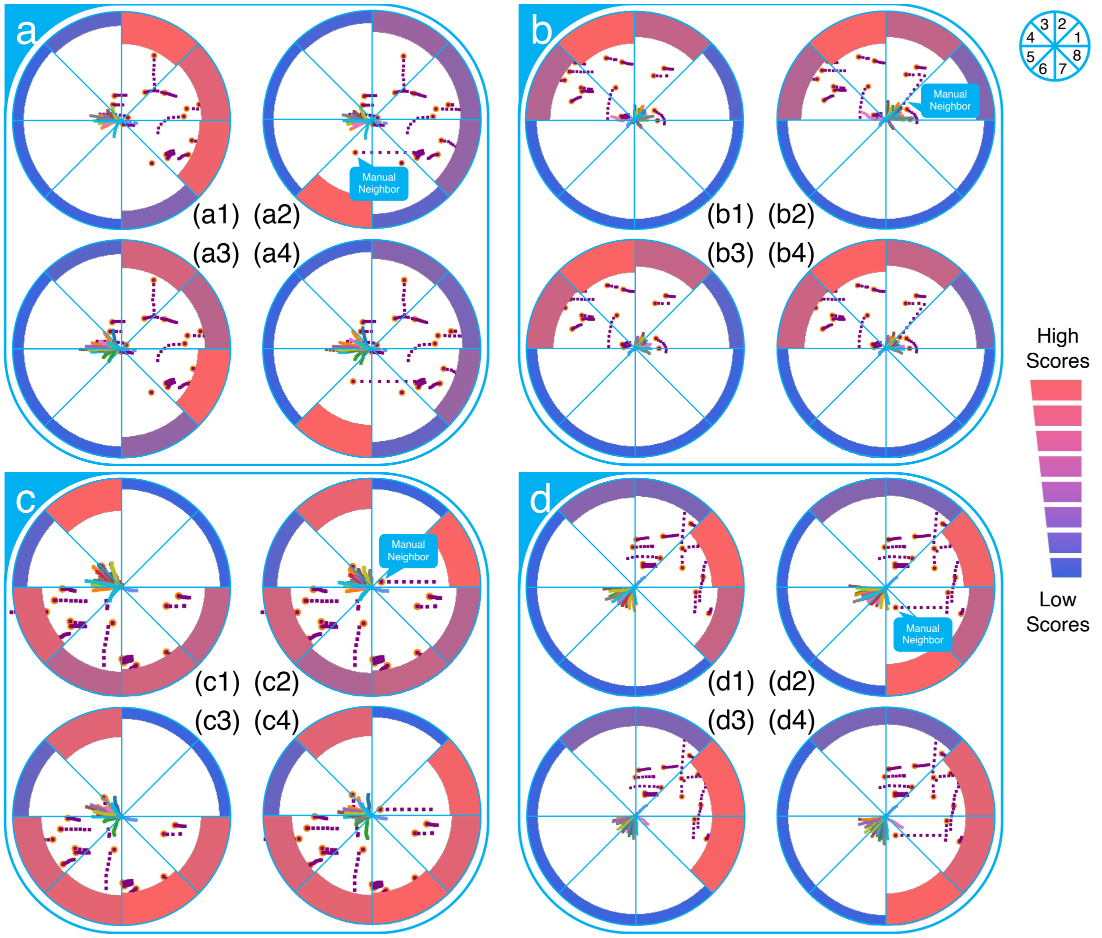

Figure 10. Visualized predicted trajectories and the corresponding attention scores of several real-world cases by adding additional manual neighbors. For each case $x \in \{\text{a, b, c, d}\}$, subfigure ($x$1) is the **4-factor** model's prediction, and (x3) is the **3-factor** model's prediction. subfigures (x2) and (x4) are obtained by adding manual neighbors to cases (x1) and (x3), respectively.

manual neighbor. However, trajectories forecasted by the 4-factor model are far worse than those predicted by the 3-factor model. In detail, several randomly generated trajectories are distributed "messily" around the target agent, which could be caused by the "misleading" of 4-factor SocialCircle on predicted trajectories at different spatial positions. In other words, the newly added movement direction factor may prevent the backbone prediction model from exhibiting its original prediction performance.

Moving Directions and Attention Scores. We visualize predictions of both 3-factor and 4-factor SocialCircle models on more real-world scenes in Fig. 9 and toy prediction cases with manual neighbors in Fig. 10. Comparing Fig. 9 (a1) and (b1), it shows that more SocialCircle partitions have been paid attention to (red colored partitions) in the 4-factor model in (a1) than (b1). Cases {(a2), (b2)} and {(a3), (b3)} also show similar trends. It means that more partitions or neighbors (*i.e.*, more "rules") are considered simultaneously to make final predictions for the 4-factor SocialCircle model. In addition, predictions provided by the 4-factor SocialCircle could hardly handle interactive behaviors in complex social interaction cases. For example, predictions in partitions 7 and 8 in Fig. 9 (b3) show strong avoidance trends to the coming neighbor. In contrast, predictions in the same partitions in (a3) have almost no responses. More visualized toy results with manual neighbors on real-world scenes are available in Fig. 9.

G.4. Summary of the Movement Direction Factor

The 3-factor SocialCircle (velocity, distance, direction) could not reflect neighbor agents' moving directions when modeling social interactions and forecasting trajectories. It takes an "average" way to handle neighbors with different movement directions, which means that its forecasted trajectories may not fit the interaction context well in some "extreme" interaction cases (like Fig. 8 (a3)).

We try to address this limitation by adding the new movement direction factor to the SocialCircle meta components. However, the newly added factor may lead to a performance drop. As we can see from the visualized predictions and attention scores, it is most likely due to adding too many constraints to the interaction cases, which reduces the model's ability to generalize across different complex prediction scenarios. Although the new factor could help to represent better interactive behaviors in some specific cases, degrading the original performance of the prediction model is something we do not expect. Therefore, the movement direction factor is deprecated in the SocialCircle. The currently proposed SocialCircle is a compromise that devotes itself to describing interactive behaviors through as few rules as possible while maximizing its usability in different trajectory prediction scenes. We will further investigate this limitation in our subsequent work.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 1, 6

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 3

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3

[4] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 1, 6

[5] Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6319–6328, 2020. 1

[6] Mihee Lee, Samuel S Sohn, Seonghyeon Moon, Sejong Yoon, Mubbasir Kapadia, and Vladimir Pavlovic. Musevae: Multi-scale vae for environment-aware long term trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2221–2230, 2022. 3

[7] Shijie Li, Yanying Zhou, Jinhui Yi, and Juergen Gall. Spatial-temporal consistency network for low-latency trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1940–1949, 2021. 6

[8] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019. 6

[9] Kostya Linou, Dzmitryi Linou, and Martijn de Boer. Nba player movements. https://github.com/linouk23/NBA-Player-Movements, 2016. 1

[10] 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. 1, 6

[11] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. pages 15233–15242, 2021. 3

[12] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020. 1, 6

[13] Alessio Monti, Alessia Bertugli, Simone Calderara, and Rita Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2551–2558. IEEE, 2021. 6

[14] Saeed Saadatnejad, Yi Zhou Ju, and Alexandre Alahi. Pedestrian 3d bounding box prediction. *arXiv preprint arXiv:2206.14195*, 2022. 3

[15] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020. 3

[16] Conghao Wong, Beihao Xia, Ziming Hong, Qinmu Peng, Wei Yuan, Qiong Cao, Yibo Yang, and Xinge You. View vertically: A hierarchical network for trajectory prediction via fourier spectrums. In *European Conference on Computer Vision*, pages 682–700. Springer, 2022. 1, 6

[17] Conghao Wong, Beihao Xia, Qinmu Peng, and Xinge You. Another vertical view: A hierarchical network for heterogeneous trajectory prediction via spectrums. *arXiv preprint arXiv:2304.05106*, 2023. 1, 3, 6

[18] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6507, 2022. 1, 2

[19] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6488–6497, 2022. 1

[20] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020. 1

[21] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M. Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9813–9823, 2021. 3

[22] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12085–12094, 2019. 6