SocialCircle: Learning the Angle-based Social Interaction Representation for Pedestrian Trajectory Prediction

Conghao Wong[1*] Beihao Xia(✉)[1*] Ziqian Zou[1] Yulong Wang[2] Xinge You[1,3]
[1]Huazhong University of Science and Technology [2]Huazhong Agricultural University
[3]Research Institute of Huazhong University of Science and Technology in Shenzhen
{conghaowong, xbh_hust, ziqianzoulive}@icloud.com, ylwang@mail.hzau.edu.cn, youxg@mail.hust.edu.cn

Abstract

Analyzing and forecasting trajectories of agents like pedestrians and cars in complex scenes has become more and more significant in many intelligent systems and applications. The diversity and uncertainty in socially interactive behaviors among a rich variety of agents make this task more challenging than other deterministic computer vision tasks. Researchers have made a lot of efforts to quantify the effects of these interactions on future trajectories through different mathematical models and network structures, but this problem has not been well solved. Inspired by marine animals that localize the positions of their companions underwater through echoes, we build ***a new angle-based trainable social interaction representation****, named* ***SocialCircle****, for continuously reflecting the context of social interactions at different angular orientations relative to the target agent. We validate the effect of the proposed SocialCircle by training it along with several newly released trajectory prediction models, and experiments show that the SocialCircle not only quantitatively improves the prediction performance, but also qualitatively helps better simulate social interactions when forecasting pedestrian trajectories in a way that is consistent with human intuitions.*

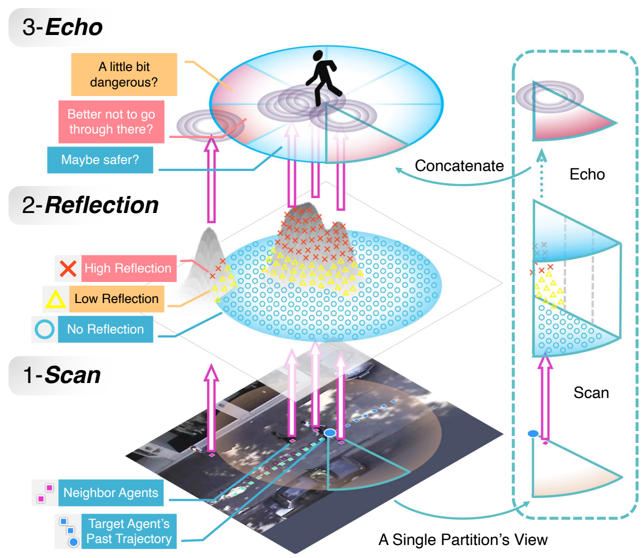

Figure 1. Motivation Illustration. Analogous to marine animals like dolphins and whales localizing other companions underwater through echolocation, we analyze agents' reactions to the potential socially interactive behaviors by assuming (1) they first ***Scan*** their interaction environment by sending signals from all angles, (2) then all neighbors feedback their ***Reflection*** signals to tell their directions, and (3) finally the target agent could make interactive decisions by the received ***echoes*** at different ***angular*** orientations.

1. Introduction

Analyzing, understanding, and forecasting behaviors of intelligent agents have been significantly required by more and more intelligent systems and applications. Due to the ease of access and analysis of trajectories, analyzing agents' behaviors through trajectories has also become a common approach. Trajectory prediction aims at forecasting agents' all possible future trajectories during a specific period by taking into account the positions of all agents that appeared in the scene [1]. It also considers the potential interactive behaviors [2, 8, 12, 33, 46] as well as the scene constraints [3, 5, 17, 25, 27, 30, 41, 47] when making predictions.

The **social interaction** [1, 28] (also known as the **agent-to-agent interaction**) considered in trajectory prediction takes into account not only all kinds of interactive behaviors among different agents but how they affect their trajectories. Current social-interaction-modeling methods in the trajectory prediction task can be classified roughly into *Model-based* and *Model-free* two classes[49]. *Model-based* methods may take some particular "rules" [49] as the primary foundation for the prediction. For example, the Social-

* equal contribution. Codes are available at https://github.com/cocoon2wong/SocialCircle.

Force-based methods [9, 28] model and simulate agents' behaviors mainly according to the rules in Newtonian mechanics. Some other methods like [3, 42, 49] also turn trajectory prediction into an optimization problem by introducing their different mathematical models. However, designing a generalized "rule" that fits most socially interactive cases is often difficult, making them challenging to apply to complex scenes. On the contrary, *model-free* methods are mostly driven by data, and few manual interventions are considered. For example, graph-based methods [4, 14, 36] may build a series of spatial or temporal graph structures, thus learning to simulate agents' social interactions. Most model-free methods could fully utilize the ability of neural networks to fit data, but they may heavily rely on different network structures and pose limited explainability.

"Rules" and "data" play essential roles but put different limitations on these methods accordingly. A natural thought is to add several "lite-rules" to data-driven backbones to provide limited constraints as guidance to improve either the data-fit process or the explainability. In short, we want to constrain the learning process with relatively weak rules for social interactions rather than solid mathematical rules, thus benefiting from both rules and data-fit capabilities.

Analyzing agents' interactive behaviors through bionics and psychology is a natural choice. Animals would not analyze others' behaviors by solving complex equations but with relatively simple judgment rules when planning trajectories. Some researchers in the social psychology area also point out that each agent in a complex multiagent system tends to behave and interact with each other according to simple rules rather than extensive computations, which inspired a series of agent-based simulation models that have been widely applied in economics and political science [35]. It is fascinating that some marine animals can locate others in the deep sea through *echolocation* rather than visual factors due to the weak light. They may firstly **scan** the environment by sending some unique signals (like ultrasounds) at different angles, which could be **reflected** in contact with others and produce **echoes**. Then, they gather echoes from different directions, thus locating, interacting, or communicating with others, and finally modifying their behaviors.

As shown in Fig. 1, the echolocation is similar to how agents interact with others. Only a few "rules" are established, like the time from they send to receive the echo and the direction where it comes. This way, we bring a simple priori to model social behaviors that interactions are considered to be **angle-based**. In detail, all interactive behaviors are considered in a special angle space where the angle θ (*which direction the echo comes from*) plays as the independent variable. We assume that most social interactions can be "inferred" by several simple components corresponding to each angle θ, like the velocity of each participant (*in which way the participant's position changes during two echolocations*) and the distance between each participant and the target agent (*how long the echo arrives since scanning*). Thus, we can obtain an angle-based vector function $\mathbf{f}(\theta)\ \ (0 \leq \theta < 2\pi)$ to represent the current socially interactive context when forecasting trajectories. We call that angle-based social interaction representation **SocialCircle**.

SocialCircle can be classified as *Model-based*, but it is also inspired by model-free methods to fit data with relatively weak rules, *i.e.*, simple components at different angles. Then, the observed trajectory and the angle-based SocialCircle will be analyzed together in a data-driven way, as they could be both *treated as* sequences, to catch the temporal-attentive portions in trajectories and the angle-attentive portions in the current interactive context simultaneously, thus establishing connections among these weak rules and agents' real-world socially interactive behaviors as well as the forecasted trajectories.

In summary, we contribute (1) The angle-based SocialCircle representation for pedestrian trajectory prediction to model social interactive behaviors; (2) The serialized modeling strategy that treats and encodes the spatial social interactions in the temporal sequences' way along with trajectories; (3) Experiments on multiple backbone prediction models show quantitative and qualitative superiority.

2. Related Work

Model-based Social Interaction Methods. Model-based methods aim to use mathematical rules as the foundation to forecast trajectories. The classic Social Force Model [9] is proposed to model human dynamics with Newtonian mechanics. Pellegrini *et al.* [28] introduce Social Force factors to model social behaviors in the multi-agent tracking task. More Social-Force-based methods like [19, 24, 50] are also proposed to model crowds' interactions.

Other mathematical tools and models are also used to simulate socially interactive behaviors when forecasting trajectories. Xie *et al.* [42] propose the "Dark Matter" model to simulate and forecast social behaviors with fields and agent-based Lagrangian Mechanics. Xia *et al.* [41] propose a social transfer function to model human socially interactive behaviors via a uniform way across multiple prediction scenes. Yue *et al.* [49] propose a neural differential equation model, in which the explicit physics model serves as a strong inductive bias in modeling pedestrian behaviors.

However, these methods are often difficult to cover all possible socially interactive cases. Even though some methods like [41, 49] draw on the advantages of data-driven approaches to make some key parameters trainable, they may still be limited by the complex mathematical rules and equations in complex prediction scenes.

Model-free Social Interaction Methods. Model-free methods simulate interactive behaviors mostly through a data-driven form. Alahi *et al.* [1] propose the Social-

Pooling method to connect nearby sequences to share hidden states with each other, thus simulating the information-sharing process. Variations of social pooling methods like [8, 30] are proposed to pool features by considering different scales or locations simultaneously. Grid-based methods like [11] have been proposed to explore additional simple rules to enhance the capacity of pooling methods. With the quick development of graph neural networks, graph structures have been widely used to model social interactions. Graph Attention Networks (GATs) [18, 26], Graph Convolutional Networks [6, 32, 36] are employed to simulate interactions as the edges between different nodes.

Most model-free methods prefer to focus more on the structure through which to fit the data so that the predicted trajectory could reflect the effects of social interactive cues. In this process, few direct mathematical rules are constraints, making them more dependent on different network structures and high-quality data.

The proposed SocialCircle tries to address these problems by introducing "lite-rules" to these trainable backbones, thus taking advantage of the data-driven approach combined with the explainability of the model-based approach to model interactive behaviors. It also avoids designing complex mathematical models or solving complex equations during the interaction modeling.

3. Method

Formulations. This work only concerns trajectories of 2D coordinates $\mathbf{p} = (x, y)^\top$. Denote the historical trajectory of agent (pedestrian) i during t_h observation steps as $\mathbf{X}^i = \left(\mathbf{p}^i_1, ..., \mathbf{p}^i_{t_h}\right)^\top$, trajectory prediction focused in this paper aims at forecasting one or more possible future trajectories $\hat{\mathbf{Y}}^i = \left(\hat{\mathbf{p}}^i_{t_h+1}, ..., \hat{\mathbf{p}}^i_{t_h+t_f}\right)^\top$ through its observed $\mathbf{X}^i$ and all its' neighbors' trajectories $\mathcal{X}^{/i} = \left\{\mathbf{X}^j | 1 \leq j \leq N_a, j \neq i\right\}$ along with the scene image $\mathbf{I}_{t_h}$.

Angle-based SocialCircle Representation. In this paper, all social-interaction-related operations are described and implemented in an "angle" space, where the angle θ is the independent variable that describes the location of interactive behaviors. We first define the angle $\theta^i(j) \in [0, 2\pi)$ to represent the relative position of a neighbor agent j to the target agent i. It is computed as the "direction" of the vector that begins from agent i and ends at agent j at the current observation step ($t = t_h$). Formally,

$$\theta^i(j) = \mathrm{atan2}\left(\mathbf{p}^j_{t_h} - \mathbf{p}^i_{t_h}\right). \tag{1}$$

Here, $\mathrm{atan2}$ is the "quadrant-sensitive" $\arctan$ function that computes angle of the input $\mathbf{p} = (x, y)^\top$ from 0 to 2π.

Agent i's **SocialCircle representation** (short for **SocialCircle**) can be treated as a head-to-tail cyclic vector function $\mathbf{f}^i(\theta)$ for all $\theta \in [0, 2\pi)$. To make the computation easier, the angle variables will be discretized into N_θ "partitions". This way, agent i's SocialCircle can be denoted as

$$\mathbf{f}^i = \left(\mathbf{f}^i(\theta_1), \mathbf{f}^i(\theta_2), ..., \mathbf{f}^i(\theta_{N_\theta})\right)^\top. \tag{2}$$

Here, $0 = \theta_0 < \theta_1 < ... < \theta_{N_\theta} = 2\pi$. As shown in Fig. 2, each $\mathbf{f}^i(\theta_n) \in \mathbb{R}^{d_{sc}}$ $(n = 1, 2, ..., N_\theta)$ is used to represent the overall interactive effort in the nth partition caused by all participants from the set $\mathbf{N}^i(\theta_n)$, which satisfies

$$\theta_{n-1} \leq \theta^i(j) < \theta_n, \quad \forall j \in \mathbf{N}^i(\theta_n). \tag{3}$$

We treat agent i as its self-neighbor in the 1st partition. Denote the number of agents in $\mathbf{N}^i(\theta_n)$ as $\left|\mathbf{N}^i(\theta_n)\right|$, we have

$$i \in \mathbf{N}^i(\theta_1), \quad \sum_n \left|\mathbf{N}^i(\theta_n)\right| = N_a. \tag{4}$$

SocialCircle Meta Components. Each SocialCircle partition is computed via three meta components:

$$\mathbf{f}^i_{\text{meta}}(\theta_n) = \left(\mathbf{f}^i_{\text{vel}}(\theta_n), \mathbf{f}^i_{\text{dis}}(\theta_n), \mathbf{f}^i_{\text{dir}}(\theta_n)\right)^\top. \tag{5}$$

(a) **Velocity $\mathbf{f}^i_{\text{vel}}$.** Agents with higher velocities may pose potentially more significant dangers to others around them. SocialCircle takes the "average velocity" (the movement length during the observation period) of all neighbors in one partition to simulate this interactive factor. Formally,

$$\mathbf{f}^i_{\text{vel}}(\theta_n) = \frac{1}{\left|\mathbf{N}^i(\theta_n)\right|} \sum_{j \in \mathbf{N}^i(\theta_n)} \left\|\mathbf{p}^j_{t_h} - \mathbf{p}^j_1\right\|_2, \tag{6}$$

(b) **Distance $\mathbf{f}^i_{\text{dis}}$.** Agents present different interaction preferences as the distance to the interaction participant changes. SocialCircle takes the average Euclidean distance (at $t = t_h$ moment) between the target agent and all its neighbors in one partition to model this factor. Formally,

$$\mathbf{f}^i_{\text{dis}}(\theta_n) = \frac{1}{\left|\mathbf{N}^i(\theta_n)\right|} \sum_{j \in \mathbf{N}^i(\theta_n)} \left\|\mathbf{p}^i_{t_h} - \mathbf{p}^j_{t_h}\right\|_2. \tag{7}$$

(c) **Direction $\mathbf{f}^i_{\text{dir}}$.** Partitioning the continuous $\theta \in [0, 2\pi)$ may cause the loss of angle details. We use the average angles of all neighbors in one partition relative to the target agent as a compensation factor. It also acts as a positional coding term to distinguish different partitions. Formally,

$$\mathbf{f}^i_{\text{dir}}(\theta_n) = \frac{1}{\left|\mathbf{N}^i(\theta_n)\right|} \sum_{j \in \mathbf{N}^i(\theta_n)} \theta^i(j). \tag{8}$$

Serialized Modeling of Social Interaction. SocialCircle partitions $\left\{\mathbf{f}^i(\theta_n)\right\}_n$ are obtained by concatenating and embedding all meta components. Formally,

$$\mathbf{f}^i(\theta_n) = \begin{cases} g_{\text{embed}}(\mathbf{0}), & \left|\mathbf{N}^i(\theta_n)\right| = 0; \\ g_{\text{embed}}\left(\mathbf{f}^i_{\text{meta}}(\theta_n)\right), & \text{Otherwise.} \end{cases} \tag{9}$$

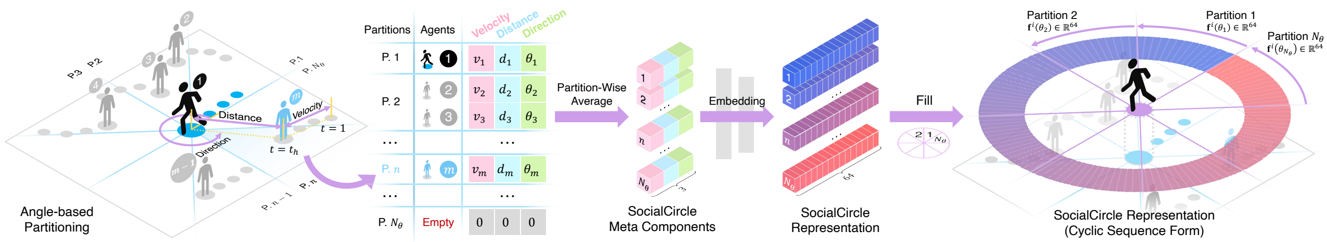

Figure 2. Computation pipeline of the proposed SocialCircle. Each agent's SocialCircle is different to others'. For a target agent, it first computes three meta components: velocity, distance, and direction. Then, these meta components will be averaged within each angle-based SocialCircle partition, and finally embedded into the set of high-dimensional head-to-tail cyclic representation $\mathbf{f}^i\left(\theta_n\right) \ (1 \leq n \leq N_\theta)$.

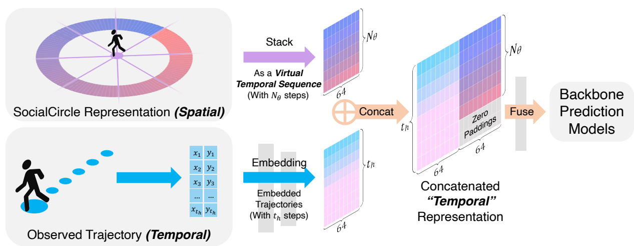

Figure 3. The serialized modeling of social interactions. SocialCircle is treated as a "virtual" temporal sequence that shares the same sequence shape as the embedded trajectory by adding zero paddings, so they could be fused to forecast trajectories together.

Here, g_{embed} is the embedding function that contains 2 fully connected layers with 64 output units. ReLU activation is used in the first layer while $\tanh$ is used in the second layer.

SocialCircle represents the ***Spatial*** interactive context at the current step ($t = t_h$) through a serialized form. A natural thought is to handle this sequence $\mathbf{f}^i \in \mathbb{R}^{N_\theta \times d_{\text{sc}}}$ along with the observed trajectory $\mathbf{X}^i \in \mathbb{R}^{t_h \times 2}$ to learn the attentive portions inner these sequences simultaneously. In detail, we treat the spatial SocialCircle $\mathbf{f}^i$ as a **Virtual *Temporal* Sequence** that shares the same sequence length as the trajectory $\mathbf{X}^i$ or its representations. As shown in Fig. 3, $\mathbf{f}^i$ will be padded at first (we set $N_\theta \leq t_h$). Formally,

$$\mathbf{f}^i_{\text{pad}} = \Big(\underbrace{\mathbf{f}^i\left(\theta_1\right), ..., \mathbf{f}^i\left(\theta_{N_\theta}\right)}_{N_\theta}, \underbrace{\mathbf{0}, ..., \mathbf{0}}_{t_h - N_\theta} \Big)^\top \in \mathbb{R}^{t_h \times d_{\text{sc}}}. \tag{10}$$

In most previous works [1, 8], agent-i's observed trajectory $\mathbf{X}^i$ will be first embedded into the high-dimensional $\mathbf{f}^i_{\text{traj}} \in \mathbb{R}^{t_h \times d}$ by some embedding layer f_{embed}, *i.e.*, $\mathbf{f}^i_{\text{traj}} = f_{\text{embed}}\left(\mathbf{X}^i\right)$. Denote the computation of one trajectory prediction model (we call the *backbone prediction model*) as B_{pred}, future trajectories $\hat{\mathbf{Y}}^i$ are usually predicted by

$$\hat{\mathbf{Y}}^i = B_{\text{pred}}\left(\mathbf{f}^i_{\text{traj}}, \mathbf{f}^i_{\text{social}}, \mathbf{f}^i_{\text{others}}\right), \tag{11}$$

where $\mathbf{f}^i_{\text{social}}$ denotes the original social representations in the backbone prediction model, and $\mathbf{f}^i_{\text{others}}$ denotes other required features (like visual features from scene image $\mathbf{I}_{t_h}$).

SocialCircle Models (the *SocialCircle-lized* backbone prediction models) take the fused vector $\mathbf{f}^i_{\text{fuse}}$ containing both trajectory information and interactive context to instead the single $\mathbf{f}^i_{\text{traj}}$ to learn the temporal-attentive portions in the observed trajectories and the angle-attentive portions in the SocialCircle simultaneously. The $\mathbf{f}^i_{\text{fuse}}$ is fused by

$$\mathbf{f}^i_{\text{fuse}} = \tanh\left(\mathbf{W}_{\text{fuse}} \operatorname{Concat}\left(\mathbf{f}^i_{\text{traj}}, \mathbf{f}^i_{\text{pad}}\right) + \mathbf{b}_{\text{fuse}}\right). \tag{12}$$

Here, $\mathbf{W}_{\text{fuse}}$ and $\mathbf{b}_{\text{fuse}}$ are the trainable weights and bias. Meanwhile, the original $\mathbf{f}^i_{\text{social}}$ will be removed. The trajectory prediction pipeline of SocialCircle models has become

$$\hat{\mathbf{Y}}^i_{\text{SC}} = B_{\text{pred}}\left(\mathbf{f}^i_{\text{fuse}}, \mathbf{f}^i_{\text{others}}\right). \tag{13}$$

Training. SocialCircle does not introduce additional new loss functions. We take Transformer[37], MSN[40], V^2-Net[38], E-V^2-Net[39] as backbone trajectory prediction models to validate SocialCircle's performance in our experiments. These models will be trained with original loss functions and settings reported in their papers.

4. Experiments

Datasets.[1] (a) **ETH**[28]-**UCY**[13] contains several videos captured in pedestrian walking scenes. We use the *leave-one-out* strategy [1] to train with $\{t_h = 8, t_f = 12\}$ and sample interval $\Delta t = 0.4$s. (b) **Stanford Drone Dataset** [29] (SDD) has 60 drone videos. Different categories of agents are annotated in pixels. Following [15], we split 60% videos to train, 20% to validate, and 20% to test under $(t_h, t_f, \Delta t) = (8, 12, 0.4)$. (c) **NBA SportVU** [16] (NBA) includes trajectories captured by the SportVU tracking system in NBA games. Following [43, 44], we set $(t_h, t_f, \Delta t) = (5, 10, 0.4)$, and sample 50K trajectories, including 65% to train, 25% to test, and 10% to validate.

Metrics. We measure prediction accuracy with the best Average/Final Displacement Error over 20 generated trajectories (*i.e.*, minADE$_{20}$/minFDE$_{20}$ [1, 8], short for ADE/FDE). See their detailed definitions in the Appendix.

[1]**Ethics Note**: Datasets used in this work are publicly available and do not contain any sensitive personally identifiable information.

Models (ETH-UCY)	eth	hotel	univ	zara1	zara2	ETH-UCY	Models (SDD)	SDD
SHENet[25] (2022)	0.41/0.61	0.13/0.20	0.25/0.43	0.21/0.32	0.15/0.26	0.23/0.36	FlowChain[20] (2023)	9.93/17.17
MID[7] (2022)	0.39/0.66	0.13/0.22	**0.22**/0.45	**0.17**/0.30	**0.13**/0.27	0.21/0.38	SHENet[25] (2022)	9.01/13.24
EqMotion[45] (2023)	0.40/0.61	**0.12**/0.18	0.23/0.43	0.18/0.32	**0.13/0.23**	0.21/0.35	IMP[34] (2023)	8.98/15.54
MSN[40] (2023)	0.27/0.41	**0.11**/0.17	0.28/0.48	0.22/0.36	0.18/0.29	0.21/0.34	LED[23] (2023)	8.48/11.36
LED[23] (2023)	0.39/0.58	**0.11**/0.17	0.26/0.43	0.18/0.26	**0.13/0.22**	0.21/0.33	MID[7] (2022)	7.91/14.50
Trajectron++[31] (2020)	0.43/0.86	**0.12**/0.19	**0.22**/0.43	**0.17**/0.32	**0.12**/0.25	0.20/0.39	Y-net[22] (2021)	7.85/11.85
Agentformer[48] (2021)	0.26/0.39	**0.11/0.14**	0.26/0.46	**0.15/0.23**	**0.14/0.23**	**0.18/0.29**	MSN[40] (2023)	7.69/12.16
V^2-Net[38] (2022)	**0.23**/0.37	**0.10/0.16**	**0.21/0.35**	0.19/0.30	**0.14/0.24**	**0.18/0.28**	V^2-Net[38] (2022)	7.12/11.39
Y-net[22] (2021)	0.28/**0.33**	**0.10/0.14**	0.24/0.41	**0.17**/0.27	**0.13/0.22**	**0.18/0.27**	E-V^2-Net[39] (2023)	**6.57/10.49**
E-V^2-Net[39] (2023)	**0.25**/0.38	**0.11/0.16**	**0.20/0.34**	0.19/0.30	**0.13/0.24**	**0.17/0.28**	NSP-SFM[49] (2022)	**6.52/10.61**
MSN-SC (Ours)	0.27/0.39	0.13/0.18	**0.22**/0.45	0.18/0.34	0.15/0.27	**0.19**/0.33	MSN-SC (Ours)	7.49/12.12
V^2-Net-SC (Ours)	**0.25**/0.37	**0.12/0.15**	**0.21/0.35**	**0.17**/0.29	**0.13/0.22**	**0.17/0.27**	V^2-Net-SC (Ours)	**6.71/10.66**
E-V^2-Net-SC (Ours)	**0.25**/0.38	**0.12/0.14**	**0.20/0.34**	0.18/0.29	**0.13/0.22**	**0.17/0.27**	E-V^2-Net-SC (Ours)	**6.54/10.36**

Table 1. Comparisons on ETH-UCY (left) and SDD (right) under *best-of-20* and with $t_h = 8$ frames' (3.2s) observations to predict future $t_f = 12$ frames' (4.8s) trajectories. Metrics are reported as "ADE/FDE". Lower ADE and FDE indicate better prediction performance.

Models (NBA)	Metrics@2.0s	Metrics@4.0s
PECNet[21] (2020)	0.96/1.69	1.83/ 3.41
NMMP[10] (2020)	0.70/1.11	1.33/ 2.05
V^2-Net[38] (2022)	0.69/0.96	1.28/ 1.68
E-V^2-Net[39] (2023)	0.68/ **0.93**	1.26/ 1.64
MemoNet[44] (2022)	0.71/1.14	1.25/ **1.47**
GroupNet+NMMP[43] (2022)	0.69/1.08	1.25/ 1.80
GroupNet+CVAE[43] (2022)	**0.62/0.95**	**1.13**/1.69
V^2-Net-SC (Ours)	**0.67/0.92**	1.22/**1.51**
E-V^2-Net-SC (Ours)	**0.67/0.90**	**1.18/1.46**

Table 2. Comparisons on NBA under *best-of-20* in meters ($t_h = 5, t_f = 10$). Metrics are reported as "ADE/FDE" at different prediction lengths, including 2.0s (5 frames) and 4.0s (10 frames).

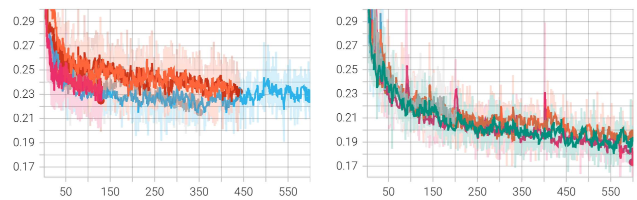

Figure 4. Loss curves (loss values after different training epochs) of the simple Transformer (left) and the corresponding Transformer-SC variation (right) during the training process on SDD with 600 epochs in total. Loss values are normalized, and each figure includes six training runs ("nan" loss values are not displayed). Curves are smoothed with the decay factor = 0.8.

Implementation details. Models are trained on one NVIDIA Tesla T4 GPU. SocialCircle is computed on each agent's 50 nearest neighbors to save the computation resource. For SocialCircle models, we set $N_\theta = t_h$[2], and set $\theta_n = 2n\pi/N_\theta$ $(n = 1, 2, ..., N_\theta)$. Feature dimensions d and d_{sc} are set to 64. Following [51], trajectories are preprocessed by moving to $(0, 0)$. We set the learning rate to 1e-4, epochs to 600, and batch size to 1500.

4.1. Comparisons to State-of-the-Art Methods

ETH-UCY. Tab. 1 illustrates that SocialCircle models are competitive. In detail, the E-V^2-Net-SC has achieved a noteworthy prediction performance with 5.6% better ADE and 6.9% better FDE compared with Agentformer. Moreover, MSN-SC performs better than EqMotion with the improvement of 9.5% ADE and 5.7% FDE, even though MSN performs not as well as other new approaches.

SDD. In Tab. 1, V^2-Net-SC outperforms Y-net by 14.52% ADE and 10.04% FDE. It has also obtained 20.87% ADE and 6.16% FDE improvements compared to the newly published LED. In addition, E-V^2-Net-SC outperforms the state-of-the-art NSP-SFM by as much as 2.36% FDE.

NBA. In Tab. 2, compared with GroupNet+CVAE, E-V^2-Net-SC's ADE is not as well as that model (about 4.42% worse), 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, E-V^2-Net-SC outperforms the other for about 5.60% ADE (4.0s). Although E-V^2-Net performs not as well as these newly published methods, the proposed SocialCircle makes it available to achieve competitive results.

4.2. Ablation Studies and Quantitative Analyses

Overall Validation of SocialCircle. As shown in Tab. 3, SocialCircle could help even the simplest Transformer [37] (which considers nothing about agents' multimodality and interactions) improve 5.89% ADE and 4.00% FDE. SocialCircle also facilitates MSN, V^2-Net, and E-V^2-Net with up to 4.92% ADE gains and 8.00% FDE gains compared to their original models. In addition, we also plot loss curves (ℓ_2 loss) of 6 training runs of the simplest Transformer and the Transformer-SC in Fig. 4. It shows that SocialCir-

[2]See detailed analyses about N_θ in the Appendix.

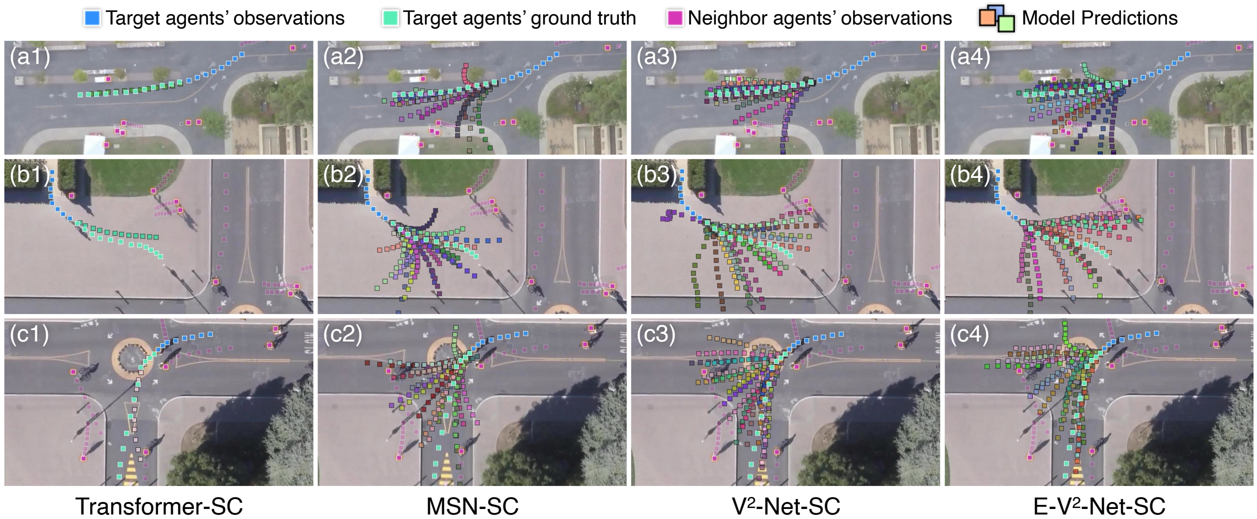

Figure 5. Visualized predictions of SocialCircle models with different backbone prediction models in several ETH-UCY and SDD scenes.

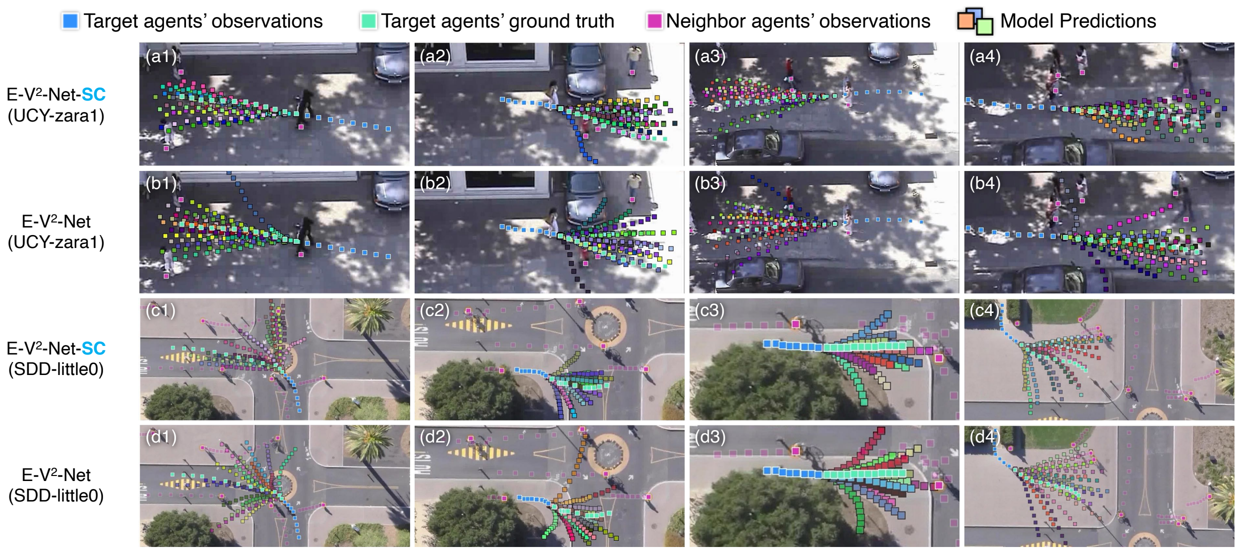

Figure 6. Prediction comparisons of SocialCircle models and their original models in several interactive ETH-UCY and SDD scenes.

cle could help the loss drop faster for Transformer-SC with an average of 13.04% lower than that in the Transformer after 600 training epochs. Moreover, 5 out of 6 Transformer runs' loss values fall into "nan" due to the inappropriate gradients before 450 training epochs. As for Transformer-SC, only two runs are terminated, demonstrating that SocialCircle may also somehow act as a normalization factor, thus improving the training stability.

Validation of SocialCircle Meta Components. As shown in Tab. 3, both V^2-Net and E-V^2-Net benefit further from the velocity and distance factors, each of which could provide at least 1.50% ADE and FDE gains (SC-a1 and SC-a2 variations). However, MSN-SC variations are not so sensitive to these two factors (less than 1.07% ADE differences among MSN-SC, MSN-SC-a1, and MSN-SC-a2) but rely more on the direction factor (up to 3.30% FDE gain has been brought by this factor). Although these factors contribute differently to different backbone models, the combination of all these factors promotes the most, for removing each one could lead to a serious performance drop.

Parameters & Efficiency. Please refer to the Appendix.

4.3. Qualitative Analyses

Visualized Predictions. Fig. 5 visualizes trajectories predicted by several SocialCircle models. We can see that all SocialCircle models' predictions present similar ways

Variations	V D R	ADE/FDE	ADE/FDE Gain (%)
Transformer*	×××	17.44/33.36	-5.89%/-4.00%
Transformer-SC	✓✓✓	16.47/32.08	(base)
MSN*	×××	7.79/13.09	-4.01%/-8.00%
MSN-SC-a1	×✓✓	7.53/12.30	-0.53%/-1.49%
MSN-SC-a2	✓×✓	7.57/12.40	-1.07%/-2.31%
MSN-SC-a3	✓✓×	7.60/12.52	-1.47%/-3.30%
MSN-SC	✓✓✓	7.49/12.12	(base)
V^2-Net*	×××	7.04/10.94	-4.92%/-2.63%
V^2-Net-SC-a1	×✓✓	6.86/10.82	-2.24%/-1.50%
V^2-Net-SC-a2	✓×✓	6.87/10.87	-2.38%/-1.97%
V^2-Net-SC-a3	✓✓×	6.78/10.71	-1.04%/-0.47%
V^2-Net-SC	✓✓✓	6.71/10.66	(base)
E-V^2-Net*	×××	6.73/10.75	-2.91%/-3.76%
E-V^2-Net-SC-a1	×✓✓	6.67/10.73	-1.99%/-3.57%
E-V^2-Net-SC-a2	✓×✓	6.64/10.55	-1.53%/-1.83%
E-V^2-Net-SC-a3	✓✓×	6.59/10.48	-0.76%/-1.16%
E-V^2-Net-SC	✓✓✓	6.54/10.36	(base)

Table 3. Ablation studies on validating SocialCircle meta components on SDD. "V", "D", and "R" indicate whether $\mathbf{f}^i_{\mathrm{vel}}$, $\mathbf{f}^i_{\mathrm{dis}}$, or $\mathbf{f}^i_{\mathrm{dir}}$ are included in the meta vector $\mathbf{f}^i_{\mathrm{meta}}$. Models with "*" are reproduced under the same condition.

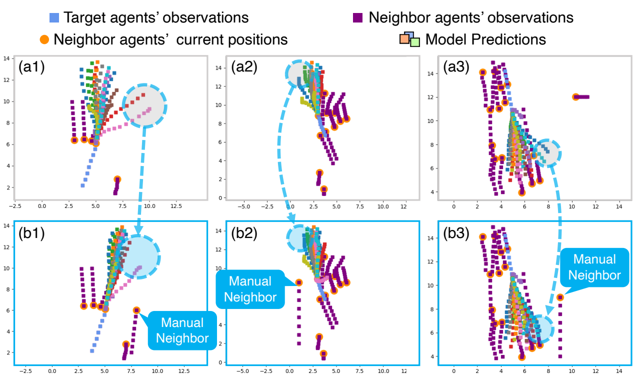

Figure 7. Toy Examples I: Validation of social interactions. The arrows and circles point to the same predicted trajectory that has been changed significantly due to the manual neighbor.

to handle social interactions. For example, predictions in Fig. 5 (a2) to (a4) all present avoidance of the group of pedestrians standing still at the roadside. Similarly, avoiding the upcoming biker about to cross the intersection has also become a major concern in (c1) to (c4). These results indicate the effectiveness of the proposed SocialCircle that works with different backbone prediction models.

Visualized Social Interaction Cases. As shown in Fig. 6, we observe that predictions given by SocialCircle models seem to be more "conservative" with a tendency to avoid others in different social interaction situations, which looks like it is trying to avoid possible collisions or too-close social distances. For example, predictions in (a2) display heavier avoidance of the man coming from

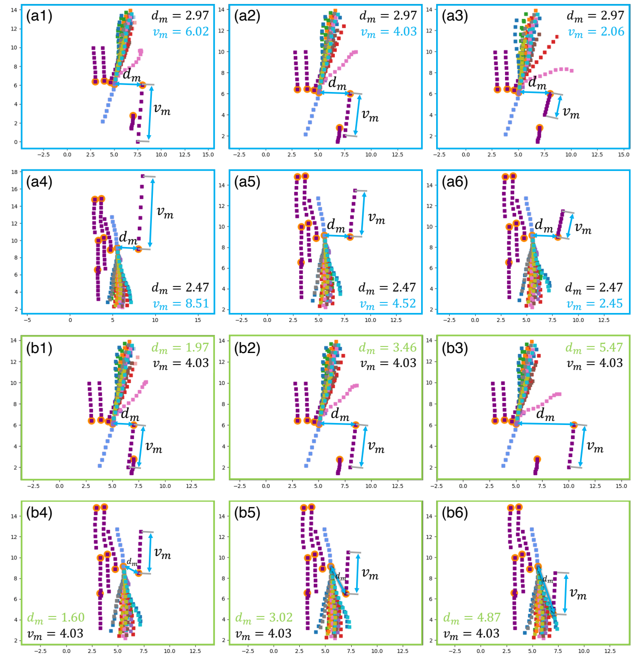

Figure 8. Toy Examples II: Validation of SocialCircle meta components by manually changing the manual neighbor's velocity (denoted by v_m) and its distance to the target agent (d_m).

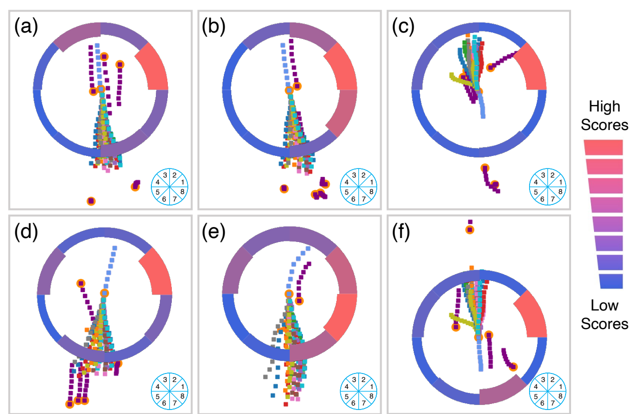

Figure 9. Visualized attention scores of each SocialCircle partition in the colored-ring form on several ETH-UCY prediction scenes. Partitions with higher attention scores (redder and wider) contribute more to the final predicted trajectories.

the car side than in (b2). Similarly, the potential collision between the moving biker and the focused walking pedestrian has been caught in (c3), thus representing a prediction shift (comparing the **blue** prediction in (c3) and the **red** one in (d3)). In addition, two pedestrians walk towards right together in cases (a4) and (b4). Unlike case (b4), SocialCircle model forecasts that the target agent may not crossover the other two's potential trajectories, like it "thought" that the agent would rather keep its relative position in the group.

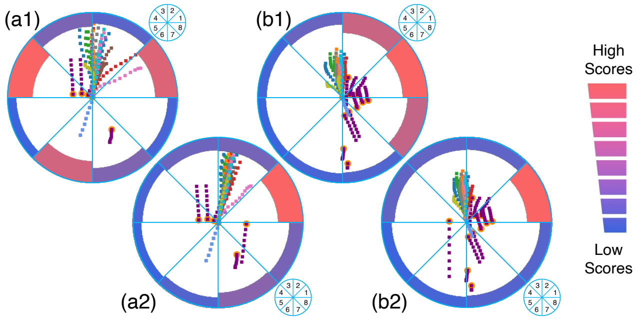

Figure 10. Toy Examples III: Comparisons of attention scores before and after adding manual neighbors in ETH-UCY scenes.

Note that social interactions are not only limited to these collision cases. They are common behavior patterns among agents, including scene-specific interactions like the game-related interactions in the NBA dataset. Please see the qualitative analyses of NBA interactions in the Appendix.

Toy Examples I (Social Interactions). We design a series of toy examples on several real-world prediction scenes from ETH-UCY to verify the capacity of SocialCircle in handling social interactions in Fig. 7. We manually add new "manual neighbors" in different positions (partitions) to test SocialCircle model's response. Each manual neighbor's observed trajectory is simulated by linearly interpolating from the given start and end points. Comparing Fig. 7 (a1) and (b1), several predicted trajectories to the right of the target agent have changed a lot. For example, the **red trajectory** contracts violently to the other side, which seems quite likely to prevent possible collisions with the manual neighbor that walks towards itself. Similar phenomenons also appear in Fig. 7 (b2), whose **blue** and **green** predictions show varying degrees of avoidance tendencies, depending on where they are located. The predicted trajectories colored in **blue** and **cyan** in Fig. 7 (a3) also show a tendency to avoid the manual neighbor. These cases illustrate the adaptivity of the SocialCircle in customizing different predictions. We have provided more toy cases in the Appendix.

Toy Examples II (SocialCircle Meta Components). Fig. 8 further shows the qualitative effect on predicted trajectories caused by the other two SocialCircle factors, velocity and distance. We first focus on the velocity factor validated in (a1) to (a3) or (a4) to (a6). When the manual neighbor has a higher velocity, the predicted trajectories will be affected by it to a greater extent, and vice versa, especially for the **pink predictions** in (a1) to (a3). Except for the velocity, as the distance d_m increases in Fig. 8 (b1) to (b3) or (b4) to (b6), the influence of manual neighbors gradually becomes smaller. In contrast, the predicted trajectory may produce a heavier shift when d_m decreases step by step. These "social-like" behaviors brought by different SocialCircle meta components are learned adaptively during training. It illustrates SocialCircle's capacity and explainability to represent potential social interaction cases and finally modify predicted trajectories socially.

Toy Examples III (SocialCircle Partitions). In Fig. 9, we visualize how each SocialCircle partition contributes by squaring-sum the feature $\mathbf{f}^i_{\text{pad}}$ in each partition, which we call the *Attention Score*. Note that agent i itself will be treated as its self-neighbor located in partition 1 (see Eq. (4)). We can see from these figures that SocialCircle acts far from simply locating all neighbors but tells which partitions should be paid more attention to when forecasting trajectories. In Fig. 9 (a) to (d), partition 1 catches more attention, and others present different trends according to different neighbor distributions. Similar to the phenomenon shown in Fig. 8, neighbors that are closer to the target agent elicit higher levels of attention (like partition 3 against partition 2 in (a)), and neighbors that move faster attract more attention (partition 4 compared to partition 6 in (d)). Unlike these cases, partition 8 owns a higher score than partition 1 in case (e), indicating that the SocialCircle cares more about the neighbor walking along with the target agent rather than the agent itself. It is interesting in case (e) that some partitions have also been assigned scores even though there are no neighbors.

Furthermore, Fig. 10 shows how the attention scores change after adding manual neighbors. In case (a1), SocialCircle focuses more on partition 4. However, the scores have changed significantly in case (a2) due to the manual neighbor located between partitions 1 and 8, and partition 4 has been less focused. Cases (b1) and (b2) also show the similar trends. All these qualitative results illustrate SocialCircle's ability to describe different interactions and the effectiveness of partitioned modeling social interactions.

Limitations. SocialCircle does not contain the directions where neighbor agents came from. It also does not directly consider the interactions among neighbor agents and how they affect the target agent, *i.e.*, the *high-order* interactions. We will further study it in our subsequent works.

5. Conclusion

This work focuses on the modeling of social interactions when forecasting pedestrian trajectories. Like marine animals localizing and communicating with others through echolocation, we bring a simple priori rule to construct the angle-based social interaction representation SocialCircle, which aims to learn how three meta components modify agent trajectories. Experiments on multiple datasets show its competitiveness, and additional toy experiments also prove its effectiveness in handling social interactions.

Acknowledgement This project is supported by National Natural Science Foundation of China (62172177), and National Key R&D Program of China (2022YFC3301000).

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, 2, 4

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[3] Catarina Barata, Jacinto C. Nascimento, João M. Lemos, and Jorge S. Marques. Sparse motion fields for trajectory prediction. *Pattern Recognition*, 110:107631, 2021. 1, 2

[4] Defu Cao, Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Spectral temporal graph neural network for trajectory prediction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1839–1845. IEEE, 2021. 2

[5] Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Scept: Scene-consistent, policy-based trajectory predictions for planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17103–17112, 2022. 1

[6] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11467–11476, 2021. 3

[7] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022. 5

[8] 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, 3, 4

[9] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[10] 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. 5

[11] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 3

[12] Parth Kothari, Brian Sifringer, and Alexandre Alahi. Interpretable social anchors for human trajectory forecasting in crowds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15556–15566, 2021. 1

[13] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007. 4

[14] Cunyan Li, Hua Yang, and Jun Sun. Intention-interaction graph based hierarchical reasoning networks for human trajectory prediction. *IEEE Transactions on Multimedia*, 2022. 2

[15] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from simulation for trajectory prediction. In *Proceedings of the European conference on computer vision (ECCV)*, 2020. 4

[16] Kostya Linou, Dzmitryi Linou, and Martijn de Boer. Nba player movements. https://github.com/linouk23/NBA-Player-Movements, 2016. 4

[17] Matteo Lisotto, Pasquale Coscia, and Lamberto Ballan. Social and scene-aware trajectory prediction in crowded spaces. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[18] Congcong Liu, Yuying Chen, Ming Liu, and Bertram E Shi. Avgcn: Trajectory prediction using graph convolutional networks guided by human attention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14234–14240. IEEE, 2021. 3

[19] Matthias Luber, Johannes A Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *2010 IEEE international conference on robotics and automation*, pages 464–469. IEEE, 2010. 2

[20] Takahiro Maeda and Norimichi Ukita. Fast inference and update of probabilistic density estimation on trajectory prediction. *arXiv preprint arXiv:2308.08824*, 2023. 5

[21] 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. 5

[22] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. pages 15233–15242, 2021. 5

[23] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5517–5526, 2023. 5

[24] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942. IEEE, 2009. 2

[25] Mancheng Meng, Ziyan Wu, Terrence Chen, Xiran Cai, Xiang Sean Zhou, Fan Yang, and Dinggang Shen. Forecasting human trajectory from scene history. *arXiv preprint arXiv:2210.08732*, 2022. 1, 5

[26] 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. 3

[27] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca

Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 1

[28] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009. 1, 2, 4

[29] 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. Springer, 2016. 4

[30] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 1, 3

[31] 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. 5

[32] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8994–9003, 2021. 3

[33] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. *arXiv preprint arXiv:2205.13296*, 2022. 1

[34] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Wei Tang, Nanning Zheng, and Gang Hua. Representing multimodal behaviors with mean location for pedestrian trajectory prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 5

[35] Eliot R Smith and Frederica R Conrey. Agent-based modeling: A new approach for theory building in social psychology. *Personality and social psychology review*, 11(1): 87–104, 2007. 2

[36] Yuchao Su, Jie Du, Yuanman Li, Xia Li, Rongqin Liang, Zhongyun Hua, and Jiantao Zhou. Trajectory forecasting based on prior-aware directed graph convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2022. 2, 3

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4, 5

[38] 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. 4, 5

[39] 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. 4, 5

[40] Conghao Wong, Beihao Xia, Qinmu Peng, Wei Yuan, and Xinge You. Msn: multi-style network for trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems*, 24:9751 – 9766, 2023. 4, 5

[41] Beihao Xia, Conghao Wong, Qinmu Peng, Wei Yuan, and Xinge You. Cscnet: Contextual semantic consistency network for trajectory prediction in crowded spaces. *Pattern Recognition*, page 108552, 2022. 1, 2

[42] Dan Xie, Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. Learning and inferring "dark matter" and predicting human intents and trajectories in videos. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1639–1652, 2017. 2

[43] 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. 4, 5

[44] 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. 4, 5

[45] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. *arXiv preprint arXiv:2303.10876*, 2023. 5

[46] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. *arXiv preprint arXiv:2203.08207*, 2022. 1

[47] Hao Xue, Du Q Huynh, and Mark Reynolds. Scene gated social graph: Pedestrian trajectory prediction based on dynamic social graphs and scene constraints. *arXiv preprint arXiv:2010.05507*, 2020. 1

[48] 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. 5

[49] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *European Conference on Computer Vision*, pages 376–394. Springer, 2022. 1, 2, 5

[50] Francesco Zanlungo, Tetsushi Ikeda, and Takayuki Kanda. Social force model with explicit collision prediction. *Europhysics Letters*, 93(6):68005, 2011. 2

[51] Pu Zhang, Jianru Xue, Pengfei Zhang, Nanning Zheng, and Wanli Ouyang. Social-aware pedestrian trajectory prediction via states refinement lstm. *IEEE transactions on pattern*

analysis and machine intelligence, 44(5):2742–2759, 2022.
5