SocialVAE: Human Trajectory Prediction using Timewise Latents

Pei Xu[1,3], Jean-Bernard Hayet[2], and Ioannis Karamouzas[1]

[1] Clemson University, South Carolina, USA
[2] CIMAT, A.C., México
[3] Roblox
peix@clemson.edu, jbhayet@cimat.mx, ioannis@clemson.edu
https://motion-lab.github.io/SocialVAE

Abstract. Predicting pedestrian movement is critical for human behavior analysis and also for safe and efficient human-agent interactions. However, despite significant advancements, it is still challenging for existing approaches to capture the uncertainty and multimodality of human navigation decision making. In this paper, we propose SocialVAE, a novel approach for human trajectory prediction. The core of SocialVAE is a timewise variational autoencoder architecture that exploits stochastic recurrent neural networks to perform prediction, combined with a social attention mechanism and a backward posterior approximation to allow for better extraction of pedestrian navigation strategies. We show that SocialVAE improves current state-of-the-art performance on several pedestrian trajectory prediction benchmarks, including the ETH/UCY benchmark, Stanford Drone Dataset, and SportVU NBA movement dataset.

Keywords: Human Trajectory Prediction, Multimodal Prediction, Timewise Variational Autoencoder.

1 Introduction

The development of autonomous agents interacting with humans, such as self-driving vehicles, indoor service robots, and traffic control systems, involves human behavior modeling and inference in order to meet both the safety and intelligence requirements. In recent years, with the rise of deep learning techniques, extracting patterns from sequential data for prediction or sequence transduction has advanced significantly in fields such as machine translation [46,8], image completion [45,33], weather forecasting [36,51], and physical simulation [41,25]. In contrast to works performing inference from regularly distributed data conforming to specific rules or physics laws, predicting human behaviors, e.g., body poses and socially-aware movement, still faces huge challenges due to the complexity and uncertainty in the human decision making process.

In this work, we focus on the task of human trajectory prediction from short-term historical observations. Traditional works predict pedestrian trajectories using deterministic models [19,18,34,21,7]. However, human navigation behaviors have an inherent multimodal nature with lots of randomness. Even in the

same scenario, there would be more than one trajectories that a pedestrian could take. Such uncertainty cannot be captured effectively by deterministic models, especially for long-term trajectory prediction with more aleatory influences introduced. Furthermore, individuals often exhibit different behaviors in similar scenarios. Such individual differences are decided by various stationary and dynamical factors such as crowd density and scene lighting, weather conditions, social context, personality traits, etc. As such, the complexity of behaviors is hard to be consistently modeled by rule-based methods, which work under predetermined physical laws and/or social rules [19,18,21,35]. Recent works have promoted data-driven solutions based on deep generative models to perform stochastic predictions or learn the trajectory distribution directly [17,48,38,30,20,40,49,54,56]. Despite impressive results, current approaches still face the challenge of making high-fidelity predictions with a limited number of samples.

In this paper, we exploit recent advances in variational inference techniques and introduce a timewise variational autoencoder (VAE) architecture for human trajectory prediction. Similar to prior VAE-based methods [30,40,49,54,56], we rely on recurrent neural networks (RNNs) to handle trajectory data sequentially and provide stochastic predictions. However, our model uses latent variables as stochastic parameters to condition the hidden dynamics of RNNs at *each time step*, in contrast to previous solutions that condition the prior of latent variables only based on historical observations. This allows us to more accurately capture the dynamic nature of human decision making. Further, to robustly extract navigation patterns from whole trajectories, we use a backward RNN structure for posterior approximation. To cope with an arbitrary number of neighbors during observation encoding, we develop a human-inspired attention mechanism to encode neighbors' states by considering the observed social features exhibited by these neighbors.

Overall, this paper makes the following contributions:

- We propose SocialVAE, a novel approach to predict human trajectory distributions conditioned on short-term historical observations. Our model employs a *timewise* VAE architecture with a conditional prior and a posterior approximated bidirectionally from the whole trajectory, and uses an attention mechanism to capture the social influence from the neighboring agents.
- We introduce *Final Position Clustering* as an optional and easy-to-implement postprocessing technique to help reduce sampling bias and further improve the overall prediction quality when a limited number of samples are drawn from the predicted distribution.
- We experimentally show that SocialVAE captures the mutimodality of human navigation behaviors while reasoning about human-human interactions in both everyday settings and NBA scenarios involving cooperative and adversarial agents with fast dynamics.
- We achieve state-of-the-art performance on the ETH/UCY and SDD benchmarks and SportVU NBA movement dataset, bringing more than 10% improvement and in certain test cases more than 50% improvement over existing trajectory prediction methods.

2   Related Work

Research in pedestrian trajectory prediction can be broadly classified into human-space and human-human models. The former focuses on predicting scene-specific human movement patterns [24,4,39,10,29] and takes advantage of the scene environment information, typically through the use of semantic maps. In this work, we are interested in the latter, which performs trajectory prediction by using dynamic information about human-human interactions.

Mathematical Models. Modeling human movement in human-human interaction settings typically leverages hand-tuned mathematical models to perform deterministic prediction. Such models include rule-based approaches using social forces, velocity-obstacles, and energy-based formulations [19,7,53,21]. Statistical models based on observed data such as Gaussian processes [50,44,22] have also been widely used. By nature, they cope better with uncertainty on overall trajectories but struggle on fine-grained variations due to social interactions.

Learning-based Models. In recent years, data-driven methods using deep learning techniques for trajectory prediction have achieved impressive results. SocialLSTM [1] employs a vanilla RNN structure using long short-term memory (LSTM) units to perform prediction sequentially. SocialAttention [48] introduces an attention mechanism to capture neighbors' influence by matching the RNN hidden state of each agent to those of its neighbors. SocialGAN [17], SoPhie [38] and SocialWays [2] use generative adversarial network (GAN) architectures. To account for social interactions, SocialGAN proposes a pooling process to synthesize neighbors via their RNN hidden states, while SocialWays adopts an attention mechanism that takes into account the neighbors' social features. PECNet [30], Trajectron [20], Trajectron++ [40], AgentFormer [56], BiTraP [54] and SGNet [49] employ a conditional-VAE architecture [43] to predict trajectory distributions, where latent variables are generated conditionally to the given observations. Memory-based approaches for trajectory prediction have also been recently explored such as MANTRA [31] and MemoNet [52]. Though achieving impressive results, such approaches would typically suffer from slow inference speeds and storage issues when dealing with large scenes. Recent works [15,55] have also exploited Transformer architectures to perform trajectory prediction. However, as we will show in Section 4, Transformer-based approaches tend to have worse performance than VAE-based approaches.

Stochastic RNNs. To better model highly dynamic and multimodal data, a number of recent works [5,13,14,16] leverage VAE architectures to extend RNNs with timewisely generated stochastic latent variables. Despite impressive performance on general sequential data modeling tasks, these approaches do not consider the interaction features appearing in human navigation tasks.

Following the literature of stochastic RNNs, we propose to use a timewise VAE as the backbone architecture for human trajectory prediction. The main motivation behind our formulation is that human decision making is highly dynamic and can lead to different trajectories at any given time. Additionally, to better extract features in human-human interactions, we employ a backward RNN for posterior approximation, which takes the *whole* ground-truth (GT)
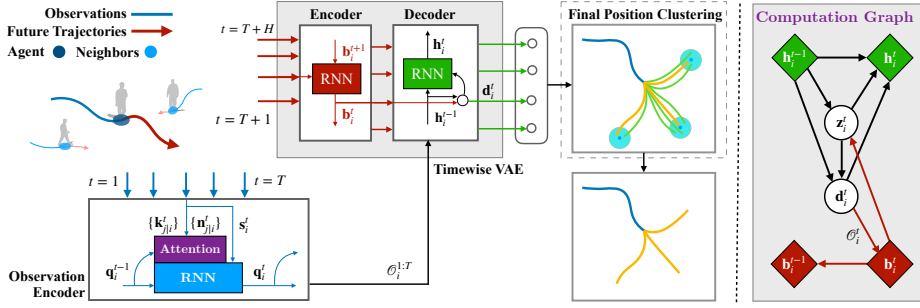
Fig. 1. Overview of SocialVAE that uses an RNN-based timewise VAE with sequentially generated stochastic latent variables for trajectory prediction. SocialVAE can be coupled with a *Final Position Clustering* postprocessing scheme to improve the prediction quality. The observation encoder attention mechanism considers each neighbor's state $\mathbf{n}_{j|i}$ along with its social features $\mathbf{k}_{j|i}$. The computation graph (right) shows the state transfer inside the timewise VAE. Diamonds represent deterministic states and circles represent stochastic states. Red parts are used only at training.

trajectory into account during learning. Neighbors are encoded through an attention mechanism that uses social features similar to [2]. A major advantage of our attention mechanism is that it relies only on the neighbors' observable states (position and velocity). In contrast, previous attention-based works use RNN hidden states as the representation of neighbors' states [17,2,38], which can only take into account neighbors that are consistently tracked during observation. As we show, our timewise VAE with the proposed attention mechanism achieves state-of-the-art performance on ETH/UCY/NBA datasets.

3   Approach

Our proposed SocialVAE approach infers the distribution of future trajectories for each agent in a scene based on given historical observations. Specifically, given a scene containing N agents, let $\mathbf{x}_i^t \in \mathbb{R}^2$ be the 2D spatial coordinate of agent i at time step t. We perform H-frame inference for the distribution over the agent's future positions $\mathbf{x}_i^{T+1:T+H}$ based on a T-frame joint observation, i.e., we estimate $p(\mathbf{x}_i^{T+1:T+H}|\mathcal{O}_i^{1:T})$ where $\mathcal{O}_i^{1:T}$ gathers the local observations from agent i to the whole scene. SocialVAE performs prediction for each agent independently, based on social features extracted from local observations, and can run with scenes having an arbitrary number of agents. This may be of particular interest in real-time and highly dynamic environments where the local neighborhood of a target agent is constantly changing or cannot be tracked consistently.

3.1   Model Architecture

Figure 1 shows the system overview of our model. Its backbone is a timewise VAE. As in prior works [16,5,13,14], we use an RNN structure to condi-

tion the sequential predictions through an auto-regressive model relying on the state variable of the RNN structure. However, while prior works directly perform predictions over time-sequence data, we introduce the past observations as *conditional variables*. Moreover, instead of directly predicting the absolute coordinates $\mathbf{x}_i^{T+1:T+H}$, we generate a displacement sequence $\mathbf{d}_i^{T+1:T+H}$, where $\mathbf{d}_i^{t+1} \triangleq \mathbf{x}_i^{t+1} - \mathbf{x}_i^t$. The target probability distribution of the displacement sequence can be written as

$$p(\mathbf{d}_i^{T+1:T+H}|\mathcal{O}_i^{1:T}) = \prod_{\tau=1}^{H} p(\mathbf{d}_i^{T+\tau}|\mathbf{d}_i^{T+1:T+\tau-1}, \mathcal{O}_i^{1:T}). \tag{1}$$

To generate stochastic predictions, we use a conditional prior over the RNN state variable to introduce latent variables at each time step, and thus obtain a timewise VAE architecture, allowing us to model highly nonlinear dynamics during multi-agent navigation. In the following, we describe our generative model for trajectory prediction and the inference process for posterior approximation.

Generative Model. Let $\mathbf{z}_i^t$ be the latent variables introduced at time step t. To implement the sequential generative model $p(\mathbf{d}_i^t|\mathbf{d}_i^{t-1}, \mathcal{O}_i^{1:T}, \mathbf{z}_i^t)$, we use an RNN in which the state variable $\mathbf{h}_i^t$ is updated recurrently by

$$\mathbf{h}_i^t = \overrightarrow{g}(\psi_{\mathbf{zd}}(\mathbf{z}_i^t, \mathbf{d}_i^t), \mathbf{h}_i^{t-1}), \tag{2}$$

for $t = T+1, \ldots, T+H$. In the recurrence, the initial state is extracted from historical observations, i.e., $\mathbf{h}_i^T = \psi_{\mathbf{h}}(\mathcal{O}_i^{1:T})$, where $\psi_{\mathbf{zd}}$ and $\psi_{\mathbf{h}}$ are two embedding neural networks. Developing Eq. 1 with $\mathbf{z}_i^t$, we obtain the generative model:

$$p(\mathbf{d}_i^{T+1:T+H}|\mathcal{O}_i^{1:T}) = \prod_{t=T+1}^{T+H} \int_{\mathbf{z}_i^t} p(\mathbf{d}_i^t|\mathbf{d}_i^{T:t-1}, \mathcal{O}_i^{1:T}, \mathbf{z}_i^t)p(\mathbf{z}_i^t|\mathbf{d}_i^{T:t-1}, \mathcal{O}_i^{1:T})\mathrm{d}\mathbf{z}_i^t. \tag{3}$$

In contrast to standard VAEs that use a standard normal distribution as the prior, our prior distribution is conditioned and can be obtained from the RNN state variable. The second term of the integral in Eq. 3 can be translated into

$$p(\mathbf{z}_i^t|\mathbf{d}_i^{T:t-1}, \mathcal{O}_i^{1:T}) := p_\theta(\mathbf{z}_i^t|\mathbf{h}_i^{t-1}), \tag{4}$$

where θ are parameters for a neural network that should be optimized. This results in a parameterized conditional prior distribution over the RNN state variable, through which we can track the distribution flexibly.

The first term of the integral in Eq. 3 implies sampling new displacements from the prior distribution p, which is conditioned on the latent variable $\mathbf{z}_i^t$, and on the observations and previous displacements captured by $\mathbf{h}_i^{t-1}$, i.e.

$$\mathbf{d}_i^t \sim p_\xi(\cdot|\mathbf{z}_i^t, \mathbf{h}_i^{t-1}), \tag{5}$$

with parameters ξ. Given the displacement definition $\mathbf{d}_i^t$, we obtain

$$\mathbf{x}_i^t = \mathbf{x}_i^T + \sum_{\tau=T+1}^{t} \mathbf{d}_i^\tau, \tag{6}$$

as a stochastic prediction for the spatial position of agent i at time t.

Inference Model. To approximate the posterior q over the latent variables, we consider the whole GT observation $\mathcal{O}_i^{1:T+H}$ to shape the latent variable distribution via a backward recurrent network [14,16]:

$$\mathbf{b}_i^t = \overleftarrow{g}(\mathcal{O}_i^t, \mathbf{b}_i^{t+1}), \tag{7}$$

for $t = T+1, \cdots, T+H$ given the initial state $\mathbf{b}_i^{T+H+1} = \mathbf{0}$. The state variable $\mathbf{b}_i^t$ provides the GT trajectory information backward from $T + H$ to t. By defining the posterior distribution as a function of both the backward state $\mathbf{b}_i^t$ and forward state $\mathbf{h}_i^t$, the latent variable $\mathbf{z}_i^t$ is drawn implicitly during inference based on the entire GT trajectory. With ϕ denoting the parameters of the network mapping $\mathbf{b}_i^t$ and $\mathbf{h}_i^{t-1}$ to the posterior parameters, we can sample latent variables as

$$\mathbf{z}_i^t \sim q_\phi(\cdot|\mathbf{b}_i^t, \mathbf{h}_i^{t-1}). \tag{8}$$

The computation graph shown in Fig. 1 gives an illustration of the dependencies of our generative and inference models. Note that the inference model (red parts in Fig. 1) is employed only at training. During testing or evaluation, only the generative model coupled with the observation encoding module is used to perform predictions, and no information from future trajectories is considered.

Training. Similarly to the standard VAE, the learning objective of our model is to maximize the evidence lower bound (ELBO) that sums up over all time steps given the target distribution defined in Eq. 3:

$$\sum_{t=T+1}^{T+H} \mathbb{E}_{\mathbf{z}_i^t \sim q_\phi(\cdot|\mathbf{b}_i^t, \mathbf{h}_i^{t-1})} \left[\log p_\xi(\mathbf{d}_i^t|\mathbf{z}_i^t, \mathbf{h}_i^{t-1})\right] - D_{KL}\left[q_\phi(\mathbf{z}_i^t|\mathbf{b}_i^t, \mathbf{h}_i^{t-1})||p_\theta(\mathbf{z}_i^t|\mathbf{h}_i^{t-1})\right]. \tag{9}$$

where p_ξ, q_ϕ and p_θ are parameterized as Gaussian distributions by networks.

Optimizing Eq. 9 with the GT value of $\mathbf{d}_i^t$ ignores accumulated errors when we project $\mathbf{d}_i^t$ back to $\mathbf{x}_i^t$ to get the final trajectory (Eq. 6). Hence, we replace the log-likelihood term with the squared error over $\mathbf{x}_i^t$ and optimize over $\mathbf{d}_i^t$ through reparameterization tricks [23], for prediction errors in previous time steps to be compensated in next time steps. The final training loss is

$$\mathbb{E}_i \left[\frac{1}{H} \sum_{t=T+1}^{T+H} \mathbb{E}_{\substack{\mathbf{d}_i^t \sim p_\xi(\cdot|\mathbf{z}_i^t, \mathbf{h}_i^{t-1}) \\ \mathbf{z}_i^t \sim q_\phi(\cdot|\mathbf{b}_i^t, \mathbf{h}_i^{t-1})}} \left[\|(\mathbf{x}_i^t - \mathbf{x}_i^T) - \sum_{\tau=T+1}^{t} \mathbf{d}_i^\tau\|^2 + q_\phi(\mathbf{z}_i^t|\mathbf{b}_i^t, \mathbf{h}_i^{t-1}) - p_\theta(\mathbf{z}_i^t|\mathbf{h}_i^{t-1})\right]\right]. \tag{10}$$

For simplicity, at training, we sample $\mathbf{z}_i^t$ and $\mathbf{d}_i^t$ only once every time, and use the reparameterization trick of Gaussian distributions to update q_ϕ, p_ξ and p_θ.

3.2   Observation Encoding

Consider a scene with an agent i being the target of our prediction process and multiple neighboring agents (their number may vary along the observation sequence of agent i). We define the local observation from agent i to the whole

scene at time step $t = 2, \cdots, T$ as the vector containing the observation to the agent itself and the synthesis of all its neighbors:

$$\mathcal{O}_i^t := \left[f_{\mathbf{s}}(\mathbf{s}_i^t), \sum\nolimits_j w_{j|i}^t f_{\mathbf{n}}(\mathbf{n}_{j|i}^t) \right], \tag{11}$$

where

- $\mathbf{s}_i^t := \left[\mathbf{d}_i^t, \mathbf{d}_i^t - \mathbf{d}_i^{t-1} \right] \in \mathbb{R}^4$ is the self-state of agent i, including the agent's velocity and acceleration information represented by position displacement,
- $\mathbf{n}_{j|i}^t := \left[\mathbf{x}_j^t - \mathbf{x}_i^t, \mathbf{d}_j^t - \mathbf{d}_i^t \right] \in \mathbb{R}^4$ is the local state of neighbor agent j relative to agent i, including its relative position and velocity,
- $f_{\mathbf{s}}$ and $f_{\mathbf{n}}$ are learnable feature extraction neural networks,
- $w_{j|i}^t$ is an attention weight through which features from an arbitrary number of neighbors are fused together into a fixed-length vector.

Neighbors are re-defined at every time step: agent j is a neighbor of agent i at time step t if $j \neq i$ and $||\mathbf{x}_j^t - \mathbf{x}_i^t|| < r_i$ where r_i is the maximal observation range of agent i. Non-neighbor agents are ignored when we compose the local observation $\mathcal{O}_i^t$. Note that we use the attention mechanism only for past observations $t \leq T$. In the case of the backward recurrent network used in Eq. 7, we simply set $w_{j|i}^t = 1$ for all neighbors to form $\mathcal{O}_i^t$ for $t > T$.

To represent the past observation sequence while embedding the target agent's navigation strategy, we employ an RNN to encode the observations sequentially via the state variable $\mathbf{q}_i^t$, i.e. $\mathcal{O}_i^{1:t} := \mathbf{q}_i^t$, with $\mathbf{q}_i^t$ updated recurrently through

$$\mathbf{q}_i^{t+1} = g(\mathcal{O}_i^{t+1}, \mathbf{q}_i^t). \tag{12}$$

The initial state $\mathbf{q}_i^1$ is extracted from the agent's and its neighbors' initial positions at $t = 1$:

$$\mathbf{q}_i^1 = \sum\nolimits_j f_{\text{init}}(\mathbf{x}_j^1 - \mathbf{x}_i^1), \tag{13}$$

where f_{init} is a feature extraction neural network.

The attention weights, $w_{j|i}^t$, are obtained by a graph attention mechanism [47], encoding node features by learnable edge weights. To synthesize neighbors at time step t based on observations from agent i, we regard agent i and its neighbors as nodes in a graph with directed edges from the neighbor to the agent. Attention weights corresponding to the edge weights are computed by

$$w_{j|i}^t = \frac{\exp(e_{j|i}^t)}{\sum_{k \neq i} \exp(e_{k|i}^t)}, \tag{14}$$

where $e_{j|i}^t$ is the edge weight from the neighbor node j to the agent node i. To obtain the edge weights, following [2], we define the social features $\mathbf{k}_{j|i}^t$ of a neighbor j observed by the agent i at time step t using three geometric features: (1) the Euclidean distance between agents i and j, (2) the cosine value of the bearing angle from agent i to neighbor j, and (3) the minimal predicted distance [32] from agent i to j within a given time horizon.

Given the neighbor's social features $\mathbf{k}_{j|i}^t$ and the agent's navigation features $\mathbf{q}_i^{t-1}$, we compute the edge weight through the cosine similarity:

$$e_{j|i}^t = \text{LeakyReLU}(f_\mathbf{q}(\mathbf{q}_i^{t-1}) \cdot f_\mathbf{k}(\mathbf{k}_{j|i}^t)), \tag{15}$$

where $f_\mathbf{q}$ and $f_\mathbf{k}$ are neural networks. This leads to a dot-product attention using social features to synthesize neighbors while modeling the observations of an agent as a graph. Here, $\mathbf{q}_i^{t-1}$, $\{\mathbf{k}_{j|i}^t\}$ and $\{f_\mathbf{n}(\mathbf{n}_{j|i}^t)\}$ correspond to the query, key and value vectors, respectively, in the vanilla attention mechanism (cf. Fig 1).

In contrast to prior works [17,38,2] that apply attention mechanisms only on the last frame of the given observation sequence, our approach computes attention at every time step. This allows to better extract an agent's navigation strategy while always taking into account the social influence from the neighbors. While prior works use RNN hidden states to represent each neighbor, our approach relies only on the neighbors' observed states, allowing to support sparse interactions where a neighbor is not consistently tracked.

3.3   Final Position Clustering

The combination of the timewise VAE approach from 3.1 and the observation encoding from 3.2 defines our vanilla Social-VAE model. It produces as many trajectory samples from the predictive distribution as required to infer the distribution over an agent's future position. However, when only a limited number of prediction

Fig. 2. An example of FPC to extract 3 predictions (orange) from 9 samples.

samples are drawn from the distribution, bias issues may arise as some samples may fall into low-density regions or too many samples may be concentrated in high-density regions. As such, we propose *Final Position Clustering* (FPC) as an optional postprocessing technique to improve the prediction diversity. With FPC, we sample at a higher rate than the desired number of predictions K. Then we run a K-means clustering on the samples' final positions and for each of the K clusters, we keep only the sample whose final position is the closest to the cluster's mean. This generates K predictions in total as the final result. Figure 2 shows an example of FPC selecting 3 predictions (orange) from 9 samples. Green trajectories depict discarded prediction samples and cyan regions are the clustering result from the final positions of these 9 samples.
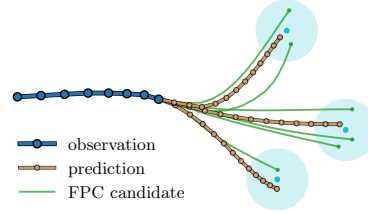
4   Experiments

In the following, we first give details on the implementation of SocialVAE and introduce the metrics used for evaluation. Then, we report and analyze experiments on two standard datasets: ETH/UCY [34,26] and SDD [37], and on the SportVU NBA movement dataset [27,57] which is a more challenging benchmark containing a rich set of complex human-human interactions.

Implementation details. Our approach uses a local observation encoding and ignores agents' global coordinates. To eliminate the global moving direction preferences, we apply two types of data augmentation: flipping and rotation. We employ GRUs [12] as the RNN structure. The slope of the LeakyReLU activation layer is 0.2. The time horizon used for the minimal predicted distance is 7s. The latent variables $\mathbf{z}_i^t$ are modeled as 32-dimensional, Gaussian-distributed variables. All models are trained with 8-frame observations ($T = 8$) and 12-frame predictions ($H = 12$). The choice of sampling rate K provides a tradeoff between evaluation performance and runtime and varies per dataset. An upper bound is set at 50. We refer to our code repository for our implementation, hyperparameters values and pre-trained models. During inference, vanilla SocialVAE runs in real time on a machine equipped with a V100 GPU, performing 20 stochastic predictions within 0.05s for a batch of 1,024 observation samples.

Evaluation Metrics and Baselines. To evaluate our approach, we consider a deterministic prediction method, *Linear*, which considers agents moving with constant velocities, and the current state-of-the-art (SOTA) baselines of stochastic prediction in human-human interaction settings, as shown in Table 1. Following the literature, we use best-of-20 predictions to compute *Average Displacement Error* (ADE) and *Final Displacement Error* (FDE) as the main metrics, and also report the *Negative Log Likelihood* (NLL) estimated from 2,000 samples. We refer to the appendix for details about these metrics. Among the considered baselines, PECNet [30], AgentFormer [56] and MemoNet [52] rely on postprocessing to improve the model performance. The reported numbers for Trajectron++ [40], BiTraP [54] and SGNet-ED [49] were obtained by training the models from scratch using the publicly released code after fixing a recently reported issue [28,58,3] which leads to performance discrepancies compared to the numbers mentioned in the original papers.

4.1 Quantitative Evaluation

Experiments on ETH/UCY. ETH/UCY benchmark [34,26] contains trajectories of 1,536 pedestrians recorded in five different scenes. We use the same preprocessing and evaluation methods as in prior works [17,2] and apply leave-one-out cross-validation. Table 1 shows the ADE/FDE results in meters. As it can be seen, the linear model struggles to capture the complex movement patterns of pedestrians, with high errors in most test cases, though some of its results [42] are better than the GAN-based approaches of SocialGAN [17], SoPhie [38] and SocialWays [2]. STAR [55] and TransformerTF [15] use Transformer architectures as a model backbone. While both Transformer-based approaches outperform the listed GAN-based approaches, with STAR achieving the best performance on ETH, they tend not to be better than the VAE-based models of Trajectron++, BiTraP and SGNet-ED. Compared to conditional-VAE baselines, our model leads to better performance, both with and without FPC. Specifically, without FPC, the improvement of SocialVAE over SGNet-ED is about 12% both on ADE and FDE. With FPC, SocialVAE improves SGNet-ED by 21% and 30% in ADE and FDE, respectively. Compared to approaches that

Table 1. Prediction errors reported as ADE/FDE in meters for ETH/UCY and in pixels for SDD. The reported values are the mean values of ADE/FDE using the best of 20 predictions for each trajectory.

	ETH	Hotel	Univ	Zara01	Zara02	SDD
Linear	1.07/2.28	0.31/0.61	0.52/1.16	0.42/0.95	0.32/0.72	19.74/40.04
SocialGAN	0.64/1.09	0.46/0.98	0.56/1.18	0.33/0.67	0.31/0.64	27.23/41.44
SoPhie	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	16.27/29.38
SocialWays	0.39/0.64	0.39/0.66	0.55/1.31	0.44/0.64	0.51/0.92	-
STAR	**0.36**/0.65	0.17/0.36	0.31/0.62	0.29/0.52	0.22/0.46	-
TransformerTF	0.61/1.12	0.18/0.30	0.35/0.65	0.22/0.38	0.17/0.32	-
MANTRA	0.48/0.88	0.17/0.33	0.37/0.81	0.22/0.38	0.17/0.32	8.96/17.76
MemoNet[†]	0.40/0.61	**0.11/0.17**	0.24/0.43	0.18/0.32	0.14/0.24	8.56/12.66
PECNet[†]	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	9.29/15.93
Trajectron++[*]	0.54/0.94	0.16/0.28	0.28/0.55	0.21/0.42	0.16/0.32	10.00/17.15
AgentFormer[†]	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24	-
BiTraP[*]	0.56/0.98	0.17/0.28	0.25/0.47	0.23/0.45	0.16/0.33	9.09/16.31
SGNet-ED[*]	0.47/0.77	0.21/0.44	0.33/0.62	0.18/0.32	0.15/0.28	9.69/17.01
SocialVAE	0.47/0.76	0.14/0.22	0.25/0.47	0.20/0.37	0.14/0.28	8.88/14.81
SocialVAE+FPC	0.41/**0.58**	0.13/0.19	**0.21/0.36**	**0.17/0.29**	**0.13/0.22**	**8.10/11.72**

[*]: reproduced results with a known issue fixed
[†]: baselines using postprocessing

require postprocessing, SocialVAE+FPC brings an improvement around 9% on ADE and 13% on FDE over AgentFormer, and 5% on FDE over MemoNet while allowing for faster inference speeds and without requiring any extra space for memory storage. The NLL evaluation results in Table 2 further show that the SocialVAE predictive distributions have superior quality, with higher probability (lower NLL) on the GT trajectories. For a fair comparison, we restrict here our analysis on SOTA methods from Table 1 that do not rely on postprocessing.

Experiments on SDD. SDD [37] includes trajectories of 5,232 pedestrians in eight different scenes. We used the TrajNet [6] split to perform the training/testing processes and converted original pixel coordinates into spatial coordinates defined in meters for training. The related ADE/FDE and NLL results are reported in Tables 1 and 2, respectively. Following previous works, ADE and FDE are reported in pixels. As shown in Table 2, SocialVAE leads to more accurate trajectory distributions as compared to other VAE-based baselines. In addition, SocialVAE+FPC provides a significant improvement over existing baselines in terms of FDE as reported in Table 1. Given that SDD has different homographies at each scene, to draw a fair comparison with results reported in meters, we refer to additional results in the appendix.

Experiment on NBA Dataset. We tested SocialVAE on the SportVU NBA movement dataset focusing on NBA games from the 2015-2016 season [27,57]. Due to the large size of the original dataset, we extracted two sub-datasets to use as benchmarks named Rebounding and Scoring, consisting of 257,230 and

Table 2. NLL estimation on tested datasets.

	ETH	Hotel	Univ	Zara01	Zara02	SDD	Rebounding	Scoring
Trajectron++	2.26	-0.52	0.32	-0.05	-1.00	1.76	2.41	2.69
BiTraP	3.68	0.48	0.71	0.49	-0.69	0.87	2.92	3.23
SGNet-ED	2.65	0.92	1.36	0.13	-0.77	1.53	3.28	3.05
SocialVAE	**0.96**	**-1.41**	**-0.49**	**-0.65**	**-2.67**	**-0.43**	**1.90**	**1.67**

2,958,480 20-frame trajectories, respectively. We refer to the appendix for details on data acquisition. The extracted trajectories capture a rich set of agent-agent interactions and highly non-linear motions. Note that the overall frequency and the adversarial and cooperative nature of the interactions are significantly different from those

Table 3. ADE/FDE on NBA Datasets.

Unit: feet	Rebounding	Scoring
Linear	2.14/5.09	2.07/4.81
Trajectron++	0.98/1.93	0.73/1.46
BiTraP	0.83/1.72	0.74/1.49
SGNet-ED	0.78/1.55	0.68/1.30
SocialVAE	0.72/1.37	0.64/1.17
SocialVAE+FPC	**0.66/1.10**	**0.58/0.95**

in ETH/UCY and SDD, which makes trajectory prediction much more challenging [28]. This is confirmed by the rather poor performance of the Linear baseline in such scenes (Table 3). SocialVAE achieves low ADE/FDE on both datasets, much better than the ones reported in prior work [28,52], though it is unclear what training/testing data such works have used. Hence, we report our own comparisons to other VAE baselines in Tables 2 and 3. Similar to ETH/UCY and SDD, SocialVAE exhibits SOTA performance on the two NBA datasets.

4.2 Qualitative Evaluation

Case Study on ETH/UCY. Figure 3 compares trajectories generated by SocialVAE with and without FPC in the Zara scene. We show heatmaps of the predictive distributions in the 3rd row. They cover the GT trajectories very well in the first three scenarios. In the 4th scenario, the agent takes a right turn that can be hardly captured from the 8-frame observation in which he keeps walking on a straight line. Though the GT trajectory is rather far from the average prediction, SocialVAE's output distribution still partially covers it. Using FPC improves the predictions diversity and helps to eliminate outliers. For example, the topmost predictions in the 2nd and 3rd columns and the rightmost one in the 4th column are drawn from low-probability regions and eliminated by FPC.

To better understand how the social-feature attention mechanism impacts the prediction model, we show the attention maps of scenes from *students003* in Fig. 4. The maps are generated by visualizing the attention weights (Eq.14) with respect to each neighbor as a white circle drawn on the location of that neighbor. The opacity and the radius of the circle increase with the weight of its associated neighbor. As it can be seen, in the 1st frame, while monitoring the
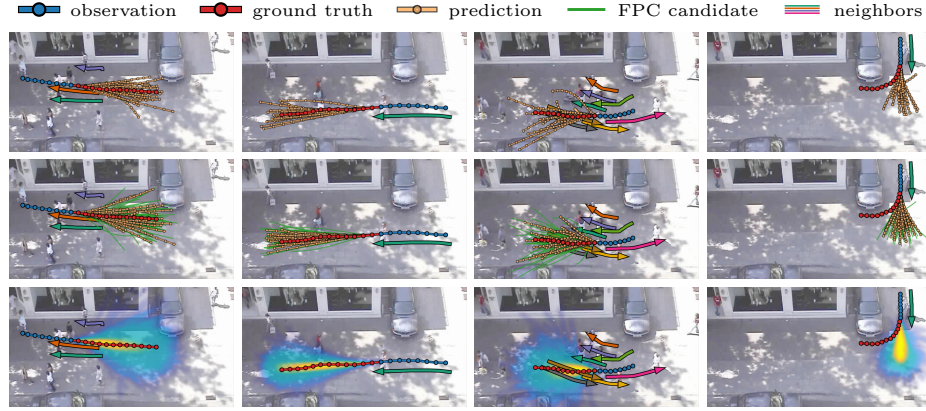
Fig. 3. Examples of predictions from SocialVAE in the UCY *Zara* scene. Observed trajectories appear in blue, predicted trajectories appear in orange, and GT is shown in red. From top to bottom: SocialVAE without FPC, SocialVAE+FPC, and heatmaps of the predicted trajectory distribution. Heatmaps are generated using 2,000 samples.
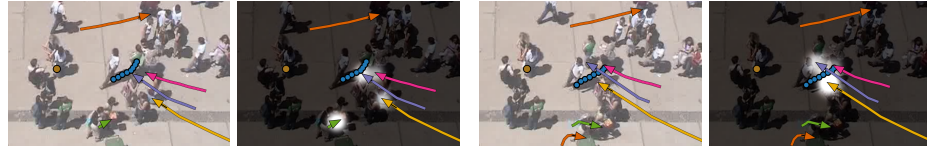

Fig. 4. Attention maps at the 1st (left) and 20th (right) frames in UCY *students003*. Trajectories of the pedestrian under prediction appear in blue. The other colored lines with arrows are the observed neighbors. Yellow dots denote stationary neighbors.

three neighbors on the right, much attention is paid to the green neighbor at the bottom who seems to be headed toward the pedestrian. In the 20th frame, the model ignores the green agent as it has changed its direction, and shifts its attention to the three nearby neighbors (red, purple, and yellow). Among these neighbors, more attention is paid to the yellow agent walking toward the target pedestrian and less to the ones behind. In both scenes, the top-left, faraway neighbor is ignored along with the idle neighbors (yellow dots).

Case Study on NBA Datasets. The NBA scenarios contain rich interactions between players in both close and remote ranges. Players show distinct movement patterns with fast changes in heading directions. Despite such complex behaviors, SocialVAE provides high-quality predictions, with distributions covering the GT trajectories closely. In the challenging example shown in the 1st row of Fig. 5, our model successfully predicts the player's intention to change his moving direction sharply and to shake his marker and catch the ball. The predictive distribution gives several directions consistent with this intention and covers well the one that the player actually took. From the attention map, we see that the model pays more attention to the defensive player who follows the one under prediction,
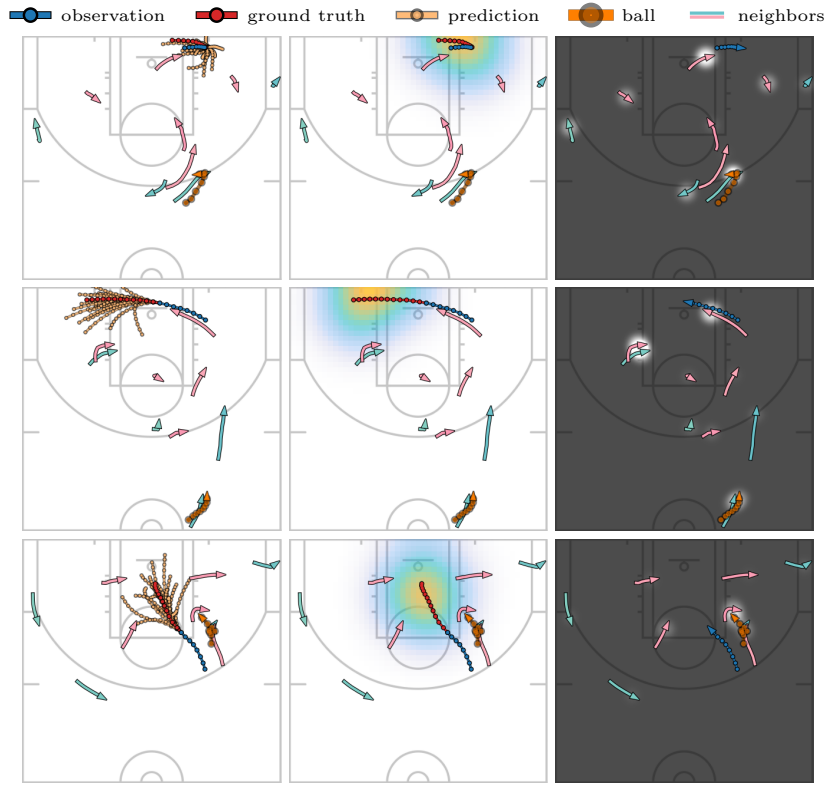
Fig. 5. Examples of predictions from SocialVAE on NBA datasets. From left to right: predicted trajectories, distribution heatmaps and attention maps. The azure lines with arrows denote trajectories of players who are in the same team as the one under prediction. The pink ones denote opposing players. The orange indicates the ball.

and, across the range from the baseline to the three-point line, to the ball. Though we do not have any semantic information about the ball, our observation encoding approach helps the model identify its importance based on historical observations. Similar behaviors are found in the examples of the 2nd and 3rd rows, where the player performs a fast move to create a passing lane and to crash the offensive board, respectively. Note that our model pays more attention to teammates and opponents who influence the predicted agent's decision making. Furthermore, the predicted trajectories clearly exhibit multimodality, reflecting multiple responding behaviors that a player could take given the same scenario.

4.3 Ablation Study

SocialVAE has four key components: timewisely generated latent variables (TL), backward posterior approximation (BP), neighborhood attention using social features (ATT), and an optional final position clustering (FPC). To understand

Table 4. Ablation studies. ETH/UCY results are the average over five tested scenes.

$\uparrow$[a] (%)	TL[b]	BP[c]	ATT[d]	FPC	ETH/UCY	SDD	Rebounding	Scoring
	-	-	-	-	0.35/0.50	13.43/17.81	1.02/1.48	1.03/1.43
14/3	✓	-	-	-	0.32/0.48	11.08/17.50	0.81/1.46	0.84/1.40
20/6	-	✓	-	-	0.30/0.47	9.45/16.01	0.78/1.46	0.74/1.39
26/11	✓	✓	-	-	0.26/0.44	9.31/15.09	0.76/1.43	0.70/1.32
33/16	✓	✓	✓	-	0.24/0.42	8.88/14.81	0.72/1.37	0.64/1.17
38/32	✓	✓	✓	✓	0.21/0.33	8.10/11.72	0.66/1.10	0.58/0.95

[a] $\uparrow$: average performance improvement related the top row as baseline;
[b] TL: timewise latents; [c] BP: backward posterior; [d] ATT: neighborhood attention.

the contributions of these components, we present the results of ablation studies in Table 4. For reference, the first row of the table shows the base performance of SocialVAE without using any of the key components. Using either the TL scheme or the BP formulation reduces the ADE/FDE values, with the combination of the two leading to an average improvement of 26%/11% on ADE/FDE, respectively. Adding the ATT mechanism can further bring the error down. In the last row, we also report the performance when FPC is applied, highlighting the value of prediction diversity. As it can be seen, using all of the four components leads to a considerable decrease in FDE and SOTA performance (cf. Tables 1 and 3). We refer to the supplementary material for an analysis on the FPC's sampling rate and explanatory visualizations of the latent space.

5   Conclusion and Future Work

We introduce SocialVAE as a novel approach for human trajectory prediction. It uses an attention-based mechanism to extract human navigation strategies from the social features exhibited in short-term observations, and relies on a timewise VAE architecture using RNN structures to generate stochastic predictions for future trajectories. Our backward RNN structure in posterior approximation helps synthesize whole trajectories for navigation feature extraction. We also introduce FPC, a clustering method applied on the predicted trajectories final positions, to improve the quality of our prediction with a limited number of prediction samples. Our approach shows state-of-the-art performance in most of the test cases from the ETH/UCY and SDD trajectory prediction benchmarks. We also highlighted the applicability of SocialVAE to SportVU NBA data. To further improve the prediction quality and generate physically acceptable trajectories, an avenue for future work is to introduce semantic scene information as a part of the model input. By doing so, our model could explicitly take both human-space and human-agent interactions into account for prediction. This would also allow us to further evaluate SocialVAE on heterogeneous datasets [9,11].

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–971 (2016)
2. Amirian, J., Hayet, J.B., Pettré, J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019)
3. Bae, I., Park, J.H., Jeon, H.G.: Non-probability sampling network for stochastic human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6477–6487 (2022)
4. Ballan, L., Castaldo, F., Alahi, A., Palmieri, F., Savarese, S.: Knowledge transfer for scene-specific motion prediction. In: European Conference on Computer Vision. pp. 697–713. Springer (2016)
5. Bayer, J., Osendorfer, C.: Learning stochastic recurrent networks. arXiv preprint arXiv:1411.7610 (2014)
6. Becker, S., Hug, R., Hübner, W., Arens, M.: An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. arXiv preprint arXiv:1805.07663 (2018)
7. van den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: International Symposium of Robotics Research. pp. 3–19 (2011)
8. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems **33**, 1877–1901 (2020)
9. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11618–11628 (2020)
10. Cao, Z., Gao, H., Mangalam, K., Cai, Q.Z., Vo, M., Malik, J.: Long-term human motion prediction with scene context. In: European Conference on Computer Vision. pp. 387–404. Springer (2020)
11. Chandra, R., Bhattacharya, U., Bera, A., Manocha, D.: Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
12. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
13. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. Advances in Neural Information Processing Systems **28** (2015)
14. Fraccaro, M., Sønderby, S.K., Paquet, U., Winther, O.: Sequential neural models with stochastic layers. Advances in Neural Information Processing Systems **29** (2016)
15. Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: IEEE International Conference on Pattern Recognition. pp. 10335–10342 (2021)
16. Goyal, A., Sordoni, A., Côté, M.A., Ke, N.R., Bengio, Y.: Z-forcing: Training stochastic recurrent networks. Advances in Neural Information Processing Systems **30** (2017)

17. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2255–2264 (2018)
18. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**(6803), 487–490 (2000)
19. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995)
20. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2375–2384 (2019)
21. Karamouzas, I., Skinner, B., Guy, S.J.: Universal power law governing pedestrian interactions. Physical Review Letters **113**(23), 238701 (2014)
22. Kim, K., Lee, D., Essa, I.: Gaussian process regression flow for analysis of motion trajectories. In: IEEE International Conference on Computer Vision. pp. 1164–1171 (2011)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Bengio, Y., Le-Cun, Y. (eds.) International Conference on Learning Representations (2014)
24. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: European Conference on Computer Vision. pp. 201–214. Springer (2012)
25. Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S.: Machine learning–accelerated computational fluid dynamics. Proceedings of the National Academy of Sciences **118**(21) (2021)
26. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum. vol. 26, pp. 655–664. Wiley Online Library (2007)
27. Linou, K., Linou, D., de Boer, M.: NBA Player Movements. `https://github.com/linouk23/NBA-Player-Movements` (2016)
28. Makansi, O., Kügelgen, J.V., Locatello, F., Gehler, P.V., Janzing, D., Brox, T., Schölkopf, B.: You mostly walk alone: Analyzing feature attribution in trajectory prediction. In: International Conference on Learning Representations (2022)
29. Mangalam, K., An, Y., Girase, H., Malik, J.: From goals, waypoints & paths to long term human trajectory forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15233–15242 (2021)
30. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: European Conference on Computer Vision. pp. 759–776. Springer (2020)
31. Marchetti, F., Becattini, F., Seidenari, L., Bimbo, A.D.: Mantra: Memory augmented networks for multiple trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7143–7152 (2020)
32. Olivier, A.H., Marin, A., Crétual, A., Pettré, J.: Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers. Gait & Posture **36**(3), 399–404 (2012)
33. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Advances in Neural Information Processing Systems **29** (2016)
34. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: IEEE International Conference on Computer Vision. pp. 261–268 (2009)

35. Pradhan, N., Burg, T., Birchfield, S.: Robot crowd navigation using predictive position fields in the potential function framework. In: Proceedings of the 2011 American control conference. pp. 4628–4633. IEEE (2011)
36. Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., et al.: Skilful precipitation now-casting using deep generative models of radar. Nature **597**(7878), 672–677 (2021)
37. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European Conference on Computer Vision. pp. 549–565. Springer (2016)
38. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1349–1358 (2019)
39. Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., Savarese, S.: Car-net: Clairvoyant attentive recurrent network. In: European Conference on Computer Vision. pp. 151–167 (2018)
40. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: European Conference on Computer Vision. pp. 683–700. Springer (2020)
41. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P.: Learning to simulate complex physics with graph networks. In: International Conference on Machine Learning. pp. 8459–8468 (2020)
42. Schöller, C., Aravantinos, V., Lay, F., Knoll, A.C.: What the constant velocity model can teach us about pedestrian motion prediction. IEEE Robotics and Automation Letters **5**(2), 1696–1703 (2020)
43. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. Advances in Neural Information Processing Systems **28**, 3483–3491 (2015)
44. Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 797–803 (2010)
45. Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International Conference on Machine Learning. pp. 1747–1756 (2016)
46. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
47. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
48. Vemula, A., Muelling, K., Oh, J.: Social attention: Modeling attention in human crowds. In: IEEE international Conference on Robotics and Automation. pp. 4601–4607 (2018)
49. Wang, C., Wang, Y., Xu, M., Crandall, D.: Stepwise goal-driven networks for trajectory prediction. IEEE Robotics and Automation Letters (2022)
50. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 283–298 (2007)
51. Weyn, J.A., Durran, D.R., Caruana, R.: Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. Journal of Advances in Modeling Earth Systems **11**(8), 2680–2693 (2019)

52. Xu, C., Mao, W., Zhang, W., Chen, S.: Remember intentions: Retrospective-memory-based trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6488–6497 (2022)
53. Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1345–1352 (2011)
54. Yao, Y., Atkins, E., Johnson-Roberson, M., Vasudevan, R., Du, X.: Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. IEEE Robotics and Automation Letters **6**(2), 1463–1470 (2021)
55. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: European Conference on Computer Vision. pp. 507–523. Springer (2020)
56. Yuan, Y., Weng, X., Ou, Y., Kitani, K.: Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. arXiv preprint arXiv:2103.14023 (2021)
57. Yue, Y., Lucey, P., Carr, P., Bialkowski, A., Matthews, I.: Learning fine-grained spatial models for dynamic sports play prediction. In: IEEE International Conference on Data Mining. pp. 670–679 (2014)
58. Zamboni, S., Kefato, Z.T., Girdzijauskas, S., Norén, C., Dal Col, L.: Pedestrian trajectory prediction with convolutional neural networks. Pattern Recognition **121**, 108252 (2022)

A   Evaluation Metrics

Below are the computation details of the evaluation metrics:

- *Average Displacement Error* (ADE), the Euclidean distance between a prediction trajectory $\{\mathbf{x}_i^t\}$ and the GT value $\{\hat{\mathbf{x}}_i^t\}$ averaged over all prediction frames for $t = T + 1, \cdots, T + H$:

$$\text{ADE}(\{\mathbf{x_i}^t\}, \{\hat{\mathbf{x}}_i^t\}) = \frac{1}{H} \sum_{t=T+1}^{T+H} ||\mathbf{x}_i^t - \hat{\mathbf{x}}_i^t||. \tag{16}$$

- *Final Displacement Error* (FDE), the Euclidean distance between the predicted position in the final frame and the corresponding GT value:

$$\text{FDE}(\{\mathbf{x_i}^t\}, \{\hat{\mathbf{x}}_i^t\}) = ||\mathbf{x}_i^{T+H} - \hat{\mathbf{x}}_i^{T+H}||. \tag{17}$$

- *Negative Log Likelihood* (NLL), the negative logarithm of the value of the predictive PDF at GT trajectories. The predictive distribution is obtained by Gaussian kernel density estimation from 2,000 samples. For simplicity, distributions at each time step are estimated independently and we use the joint distributions to compute PDF values.

B   Social Features

SocialVAE employs three social features for attention computation as shown in Fig. 6. Given an agent i at time step t and its neighbor j, these features are:

Fig. 6. Demonstration of social features used for attention computation.

- the Euclidean distance between agents i and j, i.e. $||\mathbf{p}_{ji}^t||$ where $\mathbf{p}_{ji}^t = \mathbf{x}_j^t - \mathbf{x}_i^t$;
- the cosine value of the bearing angle from agent i to neighbor j, i.e. $\cos(\mathbf{p}_{ji}^t, \mathbf{d}_i^t)$;
- the minimal predicted distance [32] from agent i to j within a time horizon h (7s by default), i.e., $||\mathbf{p}_{ji}^t + \min(\tau, h)\mathbf{v}_{ji}^t||$, where $\mathbf{v}_{ji}^t = (\mathbf{d}_j^t - \mathbf{d}_i^t)/\Delta t$, $\tau = -(\mathbf{p}_{ji}^t \cdot \mathbf{v}_{ji}^t)/||\mathbf{v}_{ji}^t||^2$, and Δt is the sampling interval between two frames.
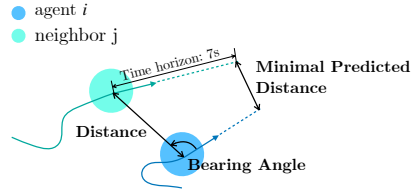
C   Data Acquisition of SportVU NBA Dataset

To test our approach on scenarios with complex and intensive human-human interactions, we have extracted two sub-datasets from the SportVU basketball movement dataset [57,28] focusing on games from the 2015-2016 NBA regular season:

Table 5. Statistical information on SportVU NBA Datasets.

	Scoring	Rebounding
# of Training/Testing Scenes	2,979/744	3,754/938
Avg. Play Duration (s)	11.82	2.94
# of Trajectories (20-frame)	2,958,480	257,230
Avg. Trajectory Length (m)	4.55	3.87

- **Rebounding dataset**. This dataset focuses on scenes involving a missed shot with players moving to grab the rebound. The dataset contains a number of interesting interactions, including players boxing out their opponents to allow a team member to grab a rebound, players moving toward the basket, and players starting to run on the other side of the half court for offensive or defensive purposes.
- **Scoring dataset**. This dataset focuses on scenes involving a team scoring a basket. The resulting dataset contains a rich set of player-player interactions, both cooperative and adversarial, including highly non-linear player motions, set plays employed by different teams, and different offensive and defensive schemes.

We refer to Table 5 for detailed characteristics of the two datasets. For each dataset, scenes are randomly split into testing and training sets using a 1:4 ratio. The original data were recorded at 25 FPS with a time interval of 0.04s between frames. In consideration that basketball players move much faster than normal pedestrians, we downsample the data to the time interval of 0.12s (instead of 0.4s that we use on ETH/UCY and SDD benchmarks). We employ the same network structure that we have used for the ETH/UCY and SDD benchmarks, and do 12-frame predictions for players (excluding the ball) based on 8-frame observations. This leads to training and testing trajectories having 20 frames, with the average length around 4m, as reported in Table 5. The neighborhood radius is set such that the whole arena is covered, which means that all the players and the ball are taken into account during observation encoding.

D   Additional Results on SDD

Table 6. ADE/FDE in meters on SDD. The reported numbers are the mean value of the best-of-20 predictions.

	Trajectron++	BiTraP	SGNet-ED	SocialVAE	SocialVAE+FPC
SDD	0.34/0.58	0.32/0.57	0.33/0.58	0.30/0.50	**0.27/0.39**

E   Sensitivity Analysis on FPC

Figure 7 plots the ADE and FDE values when FPC is applied with varying sampling rate on the ETH/UCY benchmark. As shown in the figure, the errors
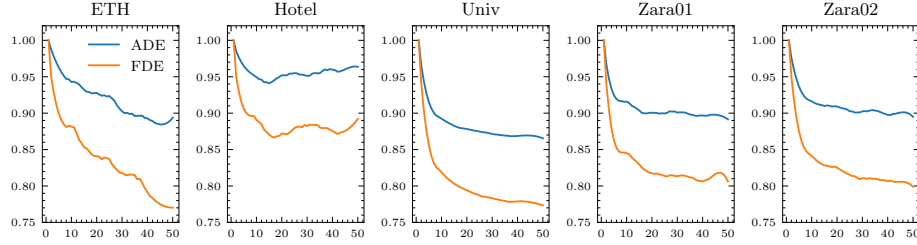
Fig. 7. Performance of FPC with respect to different sampling rates (1-50). All values are normalized by that of sampling rate 1 (no FPC).

decrease roughly as the sampling rate increases. Typically, FPC can lead to a significant improvement about 10% on ADE and 18% on FDE within a sampling rate around 20. Further increasing the sampling rate can only bring about a 2% extra improvement (with the exception of a 5% FDE improvement on ETH), at the cost though of higher running time.
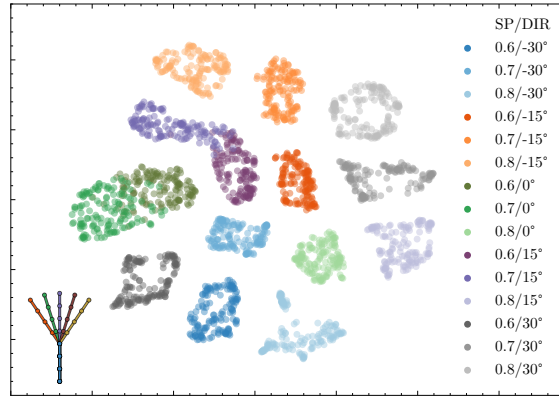
F   Latent Space Analysis

Fig. 8. t-SNE visualization of latent variable distributions given varying observation trajectories with different speeds (SP) and turn directions (DIR). The left bottom corner gives an example of the observed trajectories with different turn directions at the 5th frame from $-30°$ to $30°$. For each of the five trajectory shapes, we consider observations with three constant speeds from 0.6m to 0.8m. This gives us a combination of 15 observations, as shown in the legend.

To show that our model can learn a structured embedding of the observed trajectories, we plot the latent variable distributions in Fig. 8. To do so, we run a model pre-trained using the ETH/UCY datasets on 15 different 8-frame observations, which are the combinations of five distinct trajectory headings and three distinct speeds. For each observation, we draw 150 samples of the latent variables from the prior at the first time step of prediction, i.e. $\mathbf{z}_i^{T+1}$. As it can be seen, our model can clearly distinguish observations with semantically different features.