

Non-Probability Sampling Network for Stochastic Human Trajectory Prediction

Inhwan Bae, Jin-Hwi Park and Hae-Gon Jeon*

AI Graduate School, GIST, South Korea

{inhwanbae, jinhwipark}@gm.gist.ac.kr, haegonj@gist.ac.kr

Abstract

Capturing multimodal natures is essential for stochastic pedestrian trajectory prediction, to infer a finite set of future trajectories. The inferred trajectories are based on observation paths and the latent vectors of potential decisions of pedestrians in the inference step. However, stochastic approaches provide varying results for the same data and parameter settings, due to the random sampling of the latent vector. In this paper, we analyze the problem by reconstructing and comparing probabilistic distributions from prediction samples and socially-acceptable paths, respectively. Through this analysis, we observe that the inferences of all stochastic models are biased toward the random sampling, and fail to generate a set of realistic paths from finite samples. The problem cannot be resolved unless an infinite number of samples is available, which is infeasible in practice. We introduce that the Quasi-Monte Carlo (QMC) method, ensuring uniform coverage on the sampling space, as an alternative to the conventional random sampling. With the same finite number of samples, the QMC improves all the multimodal prediction results. We take an additional step ahead by incorporating a learnable sampling network into the existing networks for trajectory prediction. For this purpose, we propose the Non-Probability Sampling Network (NPSN), a very small network ($\sim 5K$ parameters) that generates purposive sample sequences using the past paths of pedestrians and their social interactions. Extensive experiments confirm that NPSN can significantly improve both the prediction accuracy (up to 60%) and reliability of the public pedestrian trajectory prediction benchmark. Code is publicly available at <https://github.com/inhwanbae/NPSN>.

1. Introduction

The goal of predicting pedestrian trajectories is to infer socially-acceptable paths based on previous steps while considering the social norms of other moving agents. Many earlier works [15, 39, 44, 59] on human trajectory prediction are based on deterministic approaches which yield the

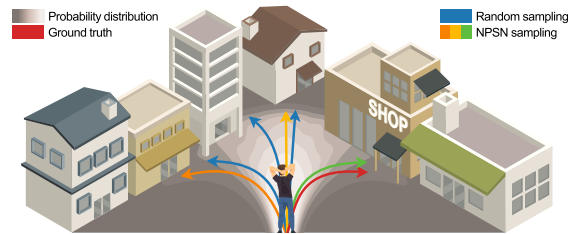


Figure 1. An illustration of a probability distribution of stochastic trajectory prediction and selected paths from each sampling method. While the trajectories from the random sampling are biased in that they do not consider space of all possible distributions, our NPSN purposely generates the accurate route, turning to SHOP, even with its low probability.

most likely single path. One of the earliest works in [15] models a social force using attractive and repulsive forces between pedestrians. Since then, motion time-series and agent interactions have been applied to trajectory forecasting. With the development of recurrent neural networks (RNNs), pioneering works such as, Social-LSTM [1] and Social-Attention [56], have adopted a social pooling and attention mechanisms between spatial neighbors. These approaches have become baseline models in areas such as spatial relation aggregation [13, 17, 40, 48, 50, 53] and temporal future prediction [23, 35, 36, 52, 63, 64].

Recently, generative models, which infer the distribution of potential future trajectories, are likely to inspire a major paradigm shift away from the single best prediction methods [6, 10, 13, 17, 18, 22, 23, 26, 27, 30–32, 35, 40, 47–54, 61, 65]. The generative models represent all possible paths, such that pedestrians may go straight, turn left/right at an intersection or take a roundabout way to avoid obstacles. To efficiently establish this multi-modality, a stochastic process is introduced to the trajectory prediction [13], which models the inferred uncertainty of pedestrians’ movements in every time frame. Stochastic trajectory prediction models start by generating a random hypothesis. Due to the non-deterministic nature of random sampling, the quality of the hypotheses depends on the number of samples. Ideally, an infinite number of hypotheses would be able to characterize all possible movements of pedestrians, but this is infeasible. In prac-

*Corresponding author

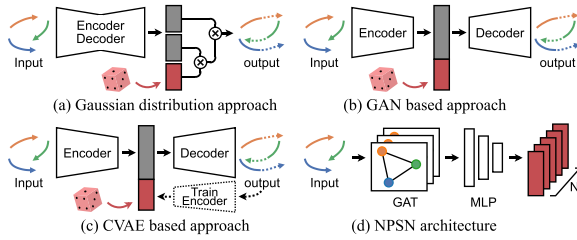


Figure 2. Illustrations of stochastic human trajectory prediction and our NPSN method. The red box indicates the latent vector.

tice, a fixed number of multiple trajectories are randomly sampled using the Monte Carlo (MC) method, and all existing stochastic models follow this random sampling strategy. However, the number of samples is typically too small to represent socially-acceptable pedestrian trajectories because they are biased toward the random sampling, as illustrated in Fig. 1.

In this paper, we revisit the state-of-the-art works which employ the stochastic process for multimodal prediction (Fig. 2-(a)~(c)). We prove that all of the expected values in the generated trajectory distributions with Generative Adversarial Networks (GANs) [6, 13, 17], Conditional Variational Auto-Encoders (CVAEs) [32, 35, 48], and Gaussian methods [40, 50] are biased. Afterward, we introduce a Quasi-Monte Carlo (QMC) sampling method that effectively alleviates this problem using a low-discrepancy sequence, instead of random sampling. Lastly, we push the random sampling forward with a learnable method: Non-Probability Sampling Network (NPSN), a very small network that generates purposive sample sequences using observations and agent interactions in Fig. 2-(d). Without structurally modifying the existing models in any way, we achieve significant improvements in the performance of pedestrian trajectory prediction. This is accomplished by replacing one line of code on random sampling with our NPSN. Interestingly, one of the existing models using our NPSN as an auxiliary module achieves the best performance in all evaluation metrics.

Unlike previous methods, the proposed approach focuses on the sampling method to generate a set of random latent vectors. To the best of our knowledge, our work is the first attempt to adopt QMC sampling and to propose a learnable method for purposive sampling in trajectory forecasting in Fig. 1.

2. Related Works

2.1. Stochastic trajectory prediction

Convolutional neural network (CNN)-based approaches using Gaussian distribution have improved the efficiency of pedestrian trajectory prediction. Social-LSTM [1], a pioneering model in this field, predicts a bivariate Gaussian distribution consisting of five parameters for the observed trajectories of pedestrians. However, it has a limitation

when inferring single paths, since it only selects the best one sample from the distribution in inference time. Follow-up works [40, 50, 51, 56] predict multiple paths by sampling multiple next coordinates based on predicted distributions.

As another methodology, a generative model is introduced to predict realistic future paths. Social-GAN [13] firstly uses a generative framework that recursively infers future trajectory. The benefit of GAN is that it generates various outputs according to latent vectors. As a result, inter-personal, socially acceptable and multimodal human behaviors are accounted for in the pedestrian trajectory prediction. Such a research stream encourages to define a variety loss which calculates for the best prediction among multiple samples for diverse sample generation. [6, 10, 17, 22, 47, 52].

Similarly, there have been attempts to predict diverse future generations using CVAE frameworks. DESIRE [23] uses a latent variable to account for the ambiguity of future paths and learns a sampling model to produce multiple hypotheses of future trajectories from given observations. This approach provides a diverse set of plausible predictions without the variety loss, and shares inspiration to objectives in many CVAE-based models [18, 32, 35, 48, 61].

All of these methods include a random sampling process and are sensitive to bias, due to the fixed number of samples, as above mentioned. In addition, current state-of-the-art models with CVAE frameworks outperform Gaussian distribution-based methods [40, 50]. In this study, we analyze these phenomena with respect to the bias of stochastic trajectory prediction, and show that the Gaussian distribution-based approaches achieve noticeable performance improvements by minimizing the bias, even better than the CVAE-based methods. Lastly, we mention a recent deterministic approach [64] that predicts multiple trajectories, which is beyond the scope of this paper.

2.2. Learning latent variables

Some works account for the transformation of latent spaces by using prior trajectory information. PECNet [35] for example uses a truncation trick in latent space to adjust the trade-off between the fidelity and the variety of samples. In their learning approach, both IDL [28] and Trajectron++ [48] predict the mean and standard deviation of a latent distribution in an inference step. Rather than directly predicting the distribution parameters, AgentFormer [62] uses a linear transform of Gaussian noise to produce the latent vector. These methodologies still run the risk of bias because of the random sampling of the latent vectors. In the present work, we aim to reduce the bias using a discrepancy loss of a set of sampled latent vectors.

2.3. Graph-based approaches

Pioneering works have introduced the concepts of social-pooling [1, 13, 52] and social-attention mechanisms [27, 56, 63] to capture the social interactions among pedestrians in

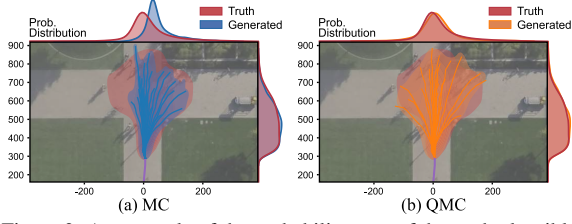


Figure 3. An example of the probability map of the truth plausible trajectory distribution and the generated distributions with $N=20$ samples $\hat{Y}_l^{1:T_{pred}}$ using MC and QMC method on PECNet [35].

scenes. Recently, Graph Neural Network (GNN)-based approaches [2, 17, 22, 27, 30, 40, 50] have been introduced to model agent-agent interactions with graph-based policies. In the GNN-based works, pedestrians are regarded as nodes of the graph, and their social relations are represented as edge weights. Social-STGCNN [40] presents a Graph Convolutional Network (GCN) [21]-based trajectory prediction which aggregates the spatial information of distances among pedestrians. Graph Attention Networks (GATs) [55] implicitly assign more weighting to edges with high social affinity on the pedestrian graph [17, 22, 50, 53, 61]. Multiverse [30] and SimAug [29] utilize GATs on 2D grids to infer feasible trajectories. Unlike these previous works, where GATs are used in the encoding process, we apply a GAT framework to a sampling process on the latent space to make a decoder predict future paths more accurately.

2.4. Monte Carlo Sampling Method

(Quasi-) Monte Carlo is a computational technique for numerical experiment using random numbers. Exploiting the random numbers allows one to approximate integrals, but this is highly error prone. The error directly depends on the random sampling methods from probability distributions. QMC sampling is developed with quasi-random sequences, known as low-discrepancy sequences [41] and is generated in a deterministic manner. It is widely utilized for many computer vision tasks, such as depth completion [57], 3D reconstruction [11, 58], motion tracking [66] and neural architecture search [8, 9]. We firstly apply QMC sampling to ensure uniform coverage of the sampling spaces for pedestrian trajectory prediction. Note that the sequence is uniformly distributed if the discrepancy tends to be zero, as the number of samples goes to infinity.

3. Generated Trajectories Are Biased

In this section, we start with the problem definition for pedestrian trajectory prediction in Sec. 3.1. We then theoretically demonstrate that generated trajectories from stochastic trajectory prediction models are biased toward random sampling in Sec. 3.2. We also introduce a way to alleviate the bias with a low-discrepancy sequence for stochastic prediction in Sec. 3.3.

3.1. Problem Definition

We formulate the pedestrian trajectory prediction task as a multi-agent future trajectory generation problem conditioned on their past trajectories. To be specific, during the observation time frames $1 \leq t \leq T_{obs}$, there are L pedestrians in a scene. The observed trajectory sequence is represented as $X_l^{1:T_{obs}} = \{X_l^t | t \in [1, \dots, T_{obs}]\}$ for $\forall l \in [1, \dots, L]$, where x_l^t is the spatial coordinate of each pedestrian l at time frame t . With the given observed sequence, the goal of the trajectory prediction is to learn potential distributions to generate N plausible future sequences $\hat{Y}_l^{1:T_{pred}} = \{\hat{Y}_{l,n}^t | t \in [1, \dots, T_{pred}], n \in [1, \dots, N]\}$ for all L pedestrians.

3.2. Stochastic Trajectory Prediction is Biased.

The generated trajectory $\hat{Y}_l^{1:T_{pred}}$ comes from a distribution of possible trajectories which are constructed by pedestrians' movements based on social forces (Fig. 3). \mathcal{T}_{truth} is an expectation value computed with a plausible trajectory distribution, and \mathcal{T}_{gen} is calculated with $\hat{Y}_l^{1:T_{pred}}$ of N which are independent and identically distributed (IID) random samples, *i.e.* the term is random if one uses different samples to generate trajectories. The expectation \mathcal{T}_{gen} is a Monte Carlo estimate of integral, *i.e.* relevant expectation.

Suppose that the expectation we want to compute from the trajectory distribution is $I(\tau) = \int_{[0,1]^s} \tau(x)q(x)dx$ which is the expected value of $\tau(x)$ for random variable x with a density q on s -dimensional unit cube $[0, 1]^s$. Then, the Monte Carlo estimator for the generated trajectory distribution with N samples can be formulated as below:

$$\hat{I}(\tau) = \hat{I}_{N,s}(\tau) = \frac{1}{N} \sum_{i=1}^N \tau(x_i), \quad (1)$$

$$Pr(\lim_{n \rightarrow \infty} \hat{I}(\tau) = I(\tau)) = 1, \quad (2)$$

where $Pr(\cdot)$ denotes a probability.

By the Strong Law of large numbers [12], the MC estimate converges to $I(\tau)$ as the number of samples N increases without bound. Now, we assume that $\tau(x)$ has a finite variance $K(\tau)$ and define the error α as below:

$$\alpha = \hat{I}_{N,s}(\tau) - I(\tau), \quad (3)$$

$$\mathbb{E}[\alpha] = 0, \quad var(\alpha) = \frac{K(\tau)}{N}, \quad (4)$$

where \mathbb{E} is an expectation and $K(t)$ is $\int (\tau(x) - I(\tau))^2 q(x)dx$. Note that the $K(\tau)$ is non-negative and depends on the function being integrated. The algorithmic goal is to specify the procedure that results in lower variance estimates of the integral.

Now consider a function of the generator F , which is sufficiently smooth, in a Monte Carlo integral $I(\tau)$. We apply the Taylor series expansion of $F(I(\tau) + \alpha)$ as follows:

$$\begin{aligned} F(\hat{I}_{N,s}(\tau)) &= F(I(\tau) + \alpha) \\ &\approx F(I(\tau)) + \alpha F'(I(\tau)) + \alpha^2 \frac{F''(I(\tau))}{2} + O(\alpha^3). \end{aligned} \quad (5)$$

Therefore, the expectation value of $F(\hat{I}_{N,s}(\tau))$ can be formulated as below:

$$\mathbb{E}[F(\hat{I}_{N,s}(\tau))] = F(I(\tau)) + \frac{M}{N} + O(\frac{1}{N^2}), \quad (6)$$

where $M = K(\tau)(F''(I(\tau))/2)$ and the $\frac{M}{N}$ is a bias. Since the term \mathcal{T}_{gen} is estimated with an MC integration, the estimate must have a bias of $\frac{M}{N} + O(\frac{1}{N^2})$. Note that the bias in the generated trajectories vanishes for $N \rightarrow \infty$, however, it is infeasible to utilize all infinite possible paths in practice. Since M depends on the generator, the generated trajectories are differently biased depending on the number of generated samples as well as the generators, which is validated in Sec. 5.2.

3.3. Quasi-Monte Carlo for Trajectory Prediction

The QMC method utilizes a low discrepancy sequence including the Halton sequence [14] and the Sobol sequence [19]. Inspired by [43], we select a Sobol sequence which not only shows consistently better performances than the Halton sequence, but also is up to 5 times faster than the MC method, even with lower error rates.

From the view of numerical analysis, an inequality in [42] proves that low-discrepancy sequences guarantees more advanced sampling in Eq. (2) with fewer integration errors as below:

$$|\alpha| \leq V(\tau) D_N^*, \quad (7)$$

where $V(\tau)$ is a total variation of function τ which is bounded variation, and D_N^* is the discrepancy of a sequence for the number of samples N . The inequality shows that a deterministic low-discrepancy sequence can be much better than the random one, for a function with finite variation. In the mathematics community, it has been proven that the Sobol sequences have a rate of convergence close to $O((\log N)^s/N)$; for a random sequence it is $O(\sqrt{\log(\log N)/N})$ in [42, 43]. For faster convergence, s needs to be small and N large (e.g., $N > 2^s$). As a result, the low discrepancy sequences have lower errors for the same number of points ($N = 20$) as shown in Tab. 1.

As an example, since x_i are IID samples from a uniformly distributed unit box for MC estimates, the samples tend to be irregularly spaced. For QMC, as x_i comes from a deterministic quasi-random sequence whose point samples are independent, they can be uniformly spaced. This guarantees a suitable distribution for pedestrian trajectory prediction by successively constructing finer uniform partitions. Fig. 4 displays a plot of a moderate number of pseudo-random points in 2-dimensional space. We observe regions of empty space where there are no points generated from the uniform distribution, which produce results skewed towards the specific destinations. However, the Sobol sequence yields evenly distributed points to enforce prediction results close to socially-acceptable paths.

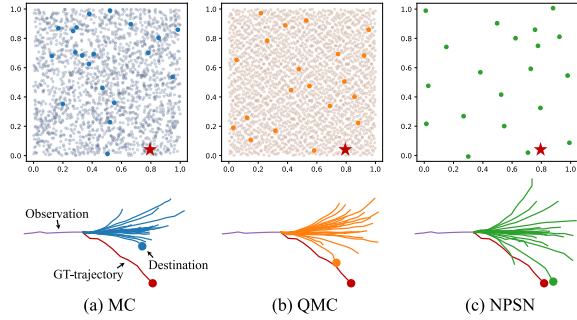


Figure 4. (Top): 2D scatter plots of 1,000 points with MC, QMC and NPSN. Stars indicate coordinates of a GT destination in the sampling space. (Bottom): Stochastic trajectory prediction results with the first 20 samples of the 1,000 points from each method.

Unfortunately, low-discrepancy sequences such as the Sobol sequence are deterministically generated and make the trajectory prediction intractable when representing an uncertainty of pedestrians' movements with various social interactions. Adding randomness into the Sobol sequence by scrambling the sequence's base digits [3] is a solution to this problem. The resultant sequence retains the advantage of QMC method, even with the same expected value. Accordingly, we utilize the scrambled Sobol sequence to generate pedestrian trajectories to account for the feasibility, the diversity, and the randomness of human behaviors.

4. Non-Probability Sampling Network

In this section, we propose NPSN, which extends the sampling technique for pedestrian trajectory prediction based on observed trajectory. Unlike the previous methods, which sample N paths in a stochastic manner, we construct a model that effectively chooses target samples using a non-probabilistic sampling technique illustrated in Fig. 2-(d).

4.1. Non-Probability Sampling on Multimodal Trajectory Prediction

In contrast to stochastic sampling, purposive sampling, one of the most common non-probability sampling techniques [4], relies on the subjective judgment of an expert to select the most productive samples rather than random selection. This approach is advantageous when studying complicated phenomena in in-depth qualitative research [38].

Since most people walk to their destinations using the shortest path, a large portion of labeled attributes in public datasets [25, 44] are straight paths. Generative attribute models learn the probabilistic distributions of social affinity features for the attribute of straight paths. However, due to the multimodal nature of human paths, the models must generate as many diverse and feasible paths as possible, using only a fixed number of samples. As a possible solution, we can purposively include a variety of samples on turning left/right and detouring around obstacles. In purposive

sampling, a maximum variation is beneficial for multimodal trajectory prediction, when examining the diverse ranges of pedestrians' movements. We make this process a learnable method, aiming to generate N heterogeneous trajectory samples with prior knowledge of past trajectories.

4.2. NPSN Architecture

We propose NPSN which substitutes the random sampling process of existing models with a learnable method. NPSN works as purposive sampling, which relies on the past trajectories of pedestrians when selecting samples in the distribution. As a result, when predicting a feasible future trajectory, a past trajectory can be used for the sampling process while also embedding informative features as a guidance. Unlike existing works [28, 35, 48, 62] that impose a restriction in the sampling space by limiting a distribution, we design all of the processes in a learnable manner.

Pedestrian graph representation. NPSN first captures the social relations using a GAT to generate socially-acceptable samples. For input trajectory $X_l^{1:T_{obs}}$, a pedestrian graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as a set of pedestrian nodes $\mathcal{V} = \{v_l | l \in [1, \dots, L]\}$ and their relation edges $\mathcal{E} = \{e_{i,j} | i, j \in [1, \dots, L]\}$. With the node features $H = \{h_l | l \in [1, \dots, L]\}$, learned feature maps for the social relation are shared across different pedestrian nodes in a scene. We utilize an attention mechanism for modeling the social interaction, whose effectiveness is demonstrated in previous works [17, 50]. The GAT allows NPSN to aggregate the features for neighbors by assigning different importance to their edge $e_{i,j}$. Here, the importance value is calculated using the attention score between two node features (h_i, h_j) .

Purposive sampling. With the interaction-aware node features, we predict N samples for each pedestrian. In particular, we use three MLP layers after the GAT layer for NPSN. By learning more prior information about samples of interest, prediction models using NPSN generate better samples. Each trajectory prediction model additionally receives an s -dimensional random latent vector along with the observed trajectory. Therefore, the NPSN must predict a set of output $S_l = [S_{l,1}, \dots, S_{l,N}]$. The output passes through a prediction model to generate N final trajectories for each pedestrian. For temporal consistency, we use the same set of purposive samples for all prediction time frames $[1, \dots, T_{pred}]$ of each pedestrian node. This process is repeated for all pedestrian nodes, and the output shape of the NPSN is $S = \mathbb{R}^{L \times s \times N}$.

Loss function. To optimize trajectory prediction models with our NPSN, we use two loss functions to generate well-distributed purposive samples. First, a winner-takes-all process [46], which generates a path closest to its ground truth, is trained to regress the accurate positions of pedestrians. Similar to [13], we measure a L_2 distance between the N prediction paths and the ground-truth, and use only one path with the smallest error for training:

$$\mathcal{L}_{dist} = \frac{1}{L} \sum_{l=1}^L \min_{n \in [1, \dots, N]} \|\hat{Y}_{l,n}^{1:T_{pred}} - Y_l^{1:T_{pred}}\|. \quad (8)$$

However, we observe that all N sample points are sometimes closely located near its ground-truth as learning progresses. This is a common problem in purposive sampling, because certain samples can be over-biased due to data imbalance, *i.e.* a large portion of the trajectory moving along one direction of the walkway. For this reason, we introduce a novel discrepancy loss to keep the N sample points with low-discrepancy, as below:

$$\mathcal{L}_{disc} = \frac{1}{LN} \sum_{l=1}^L \sum_{i=1}^N -\log \min_{\substack{j \in [1, \dots, N] \\ j \neq i}} \|S_{l,i} - S_{l,j}\|. \quad (9)$$

The objective of discrepancy loss is to maximize distances among the closest neighbors of N samples. If the distance is closer, the loss imposes a higher penalty to ensure their uniform coverage of the sampling space.

The final loss function is a linear combination of both the distance and the discrepancy loss $\mathcal{L} = \mathcal{L}_{dist} + \lambda \mathcal{L}_{disc}$. We set $\lambda = 1e-2$ to balance the scale of both terms.

4.3. Implementation Details

Transformation of one distribution to another. While most human trajectory prediction models use a normal distribution, the Sobol sequence and our NPSN are designed to produce a uniform distribution. We bridge the gap by transforming between the uniform distribution and the normal distribution. There are some representative methods including Ziggurat method [37], Inverse Cumulative Distribution Function (ICDF), and Box-Muller Transform [5]. In this work, we utilize the Box-Muller transform which is differentiable and enables an efficient execution on a GPU with the lowest QMC error, as demonstrated in [16, 67]. The formula of the Box-muller transform is as follows:

$$\begin{aligned} Z_{odd} &= \sqrt{-2 \ln(U_{even})} \cos(2\pi U_{odd}), \\ Z_{even} &= \sqrt{-2 \ln(U_{even})} \sin(2\pi U_{odd}), \end{aligned} \quad (10)$$

where U is an independent sample set from a uniform distribution and Z is an independent random variable from a standard normal distribution.

Training Procedure. Our NPSN is embedded into the state-of-the-art pedestrian trajectory prediction models [6, 13, 17, 32, 35, 40, 48, 50] by simply replacing their random sampling part. The parameters of the models are initialized using the weights provided by the authors, except for two models [17, 35] which use weights reproduced from the authors' source codes. Our NPSN has only 5,128 learnable parameters on $s = 2$ and $N = 20$. We train the prediction models with NPSN using an AdamW optimizer [34] with a batch size of 128 and a learning rate of $1e-3$ for 128 epochs. We step down the learning rate with a gain of 0.5 at every 32 epochs. Training time takes about three hours on a machine with an NVIDIA 2080TI GPU.

Social-STGCNN [40] ($s = 2$)													SGCN [50] ($s = 2$)												
MC (Baseline)				QMC				NPSN					MC (Baseline)				QMC				NPSN				
ADE ↓	FDE ↓	TCC ↑		ADE ↓	FDE ↓	TCC ↑	Gain ↑	ADE ↓	FDE ↓	TCC ↑	Gain ↑		ADE ↓	FDE ↓	TCC ↑		ADE ↓	FDE ↓	TCC ↑	Gain ↑	ADE ↓	FDE ↓	TCC ↑	Gain ↑	
ETH	0.650	1.097	0.510	0.611	1.025	0.579	6.5%	0.443	0.652	0.565	40.6%		0.567	0.997	0.545	0.495	0.810	0.596	18.8%	0.357	0.588	0.624	41.0%		
HOTEL	0.496	0.858	0.270	0.342	0.517	0.289	39.8%	0.213	0.342	0.298	60.2%		0.308	0.533	0.295	0.212	0.309	0.314	42.0%	0.159	0.253	0.355	52.6%		
UNIV	0.441	0.798	0.637	0.364	0.628	0.725	21.3%	0.278	0.443	0.762	44.5%		0.374	0.668	0.689	0.310	0.555	0.737	16.9%	0.229	0.394	0.820	41.0%		
ZARA1	0.341	0.532	0.710	0.315	0.526	0.775	1.1%	0.248	0.430	0.802	19.1%		0.285	0.508	0.746	0.245	0.446	0.803	12.1%	0.182	0.318	0.854	37.3%		
ZARA2	0.305	0.482	0.394	0.288	0.497	0.467	-3.2%	0.217	0.379	0.439	21.4%		0.225	0.422	0.491	0.193	0.359	0.503	14.9%	0.138	0.245	0.735	41.8%		
AVG	0.447	0.753	0.504	0.384	0.639	0.567	15.2%	0.280	0.449	0.573	37.2%		0.352	0.626	0.553	0.291	0.496	0.591	20.7%	0.213	0.360	0.678	42.5%		
SDD	20.76	33.18	0.471	19.21	31.81	0.498	4.1%	11.80	18.43	0.551	44.5%		25.00	41.52	0.570	21.97	38.04	0.604	8.4%	17.12	28.97	0.650	30.2%		
GCS	14.72	23.87	0.698	13.42	22.18	0.724	7.1%	9.72	15.69	0.760	34.3%		11.18	20.65	0.777	10.10	18.69	0.795	9.5%	7.66	13.41	0.789	35.1%		

Social-GAN [13] ($s = 8$)													STGAT [17] ($s = 16$)												
MC (Baseline)				QMC				NPSN					MC (Baseline)				QMC				NPSN				
ADE ↓	FDE ↓	TCC ↑		ADE ↓	FDE ↓	TCC ↑	Gain ↑	ADE ↓	FDE ↓	TCC ↑	Gain ↑		ADE ↓	FDE ↓	TCC ↑		ADE ↓	FDE ↓	TCC ↑	Gain ↑	ADE ↓	FDE ↓	TCC ↑	Gain ↑	
ETH	0.767	1.397	0.592	0.760	1.379	0.596	1.3%	0.718	1.264	0.539	9.5%		0.679	1.203	0.576	0.668	1.175	0.576	2.4%	0.612	1.020	0.602	15.2%		
HOTEL	0.434	0.876	0.322	0.431	0.870	0.323	0.7%	0.385	0.720	0.311	17.8%		0.346	0.661	0.338	0.342	0.663	0.351	-0.3%	0.308	0.566	0.301	14.4%		
UNIV	0.745	1.497	0.686	0.744	1.494	0.690	0.2%	0.711	1.427	0.715	4.7%		0.545	1.164	0.759	0.541	1.152	0.763	1.0%	0.535	1.133	0.795	2.6%		
ZARA1	0.346	0.693	0.801	0.343	0.686	0.805	0.9%	0.344	0.683	0.798	1.4%		0.345	0.687	0.791	0.343	0.683	0.818	0.6%	0.338	0.678	0.822	1.4%		
ZARA2	0.356	0.721	0.474	0.354	0.716	0.476	0.7%	0.345	0.696	0.491	3.5%		0.304	0.620	0.508	0.288	0.588	0.484	5.2%	0.304	0.621	0.557	-0.1%		
AVG	0.530	1.037	0.575	0.526	1.029	0.578	0.8%	0.501	0.958	0.571	7.6%		0.444	0.867	0.594	0.436	0.852	0.598	1.7%	0.419	0.804	0.616	7.3%		
SDD	13.58	24.59	0.598	13.41	24.24	0.601	1.4%	13.03	23.04	0.630	6.3%		14.85	28.17	0.590	14.82	28.12	0.594	0.2%	13.67	25.24	0.613	10.4%		
GCS	15.85	32.57	0.783	15.80	32.44	0.785	0.4%	15.78	32.17	0.798	1.2%		15.57	31.82	0.798	15.55	31.80	0.798	0.1%	15.71	32.12	0.798	-0.9%		

Trajectron++ [48] ($s = 25$)													PECNet [35] ($s = 16$)												
MC (Baseline)				QMC				NPSN					MC (Baseline)				QMC				NPSN				
ADE ↓	FDE ↓	TCC ↑		ADE ↓	FDE ↓	TCC ↑	Gain ↑	ADE ↓	FDE ↓	TCC ↑	Gain ↑		ADE ↓	FDE ↓	TCC ↑		ADE ↓	FDE ↓	TCC ↑	Gain ↑	ADE ↓	FDE ↓	TCC ↑	Gain ↑	
ETH	0.432	0.857	0.567	0.425	0.841	0.572	2.0%	0.397	0.761	0.576	11.2%		0.610	1.073	0.596	0.601	1.036	0.602	3.5%	0.550	0.882	0.618	17.8%		
HOTEL	0.121	0.191	0.350	0.119	0.186	0.352	2.8%	0.116	0.176	0.357	7.9%		0.222	0.390	0.335	0.214	0.369	0.336	5.3%	0.188	0.288	0.359	26.1%		
UNIV	0.220	0.428	0.789	0.216	0.420	0.790	1.8%	0.215	0.414	0.787	3.3%		0.335	0.558	0.752	0.326	0.533	0.759	4.5%	0.289	0.439	0.765	21.4%		
ZARA1	0.169	0.321	0.801	0.165	0.312	0.802	2.8%	0.166	0.314	0.801	2.2%		0.250	0.448	0.808	0.241	0.425	0.816	5.1%	0.209	0.333	0.839	25.8%		
ZARA2	0.124	0.251	0.667	0.122	0.244	0.669	2.7%	0.121	0.241	0.672	4.0%		0.186	0.332	0.596	0.178	0.310	0.609	6.7%	0.159	0.252	0.641	24.0%		
AVG	0.213	0.410	0.635	0.209	0.401	0.637	2.2%	0.203	0.381	0.639	7.1%		0.321	0.560	0.617	0.312	0.535	0.624	4.6%	0.279	0.439	0.644	21.7%		
SDD	11.40	20.12	0.652	11.22	19.69	0.656	2.1%	11.12	18.95	0.653	5.8%		9.97	15.89	0.647	9.72	15.22	0.652	4.2%	8.56	11.85	0.665	25.4%		
GCS	12.75	24.23	0.802	12.47	23.50	0.802	3.0%	12.36	22.98	0.805	5.2%		17.08	29.30	0.708	17.09	29.23	0.711	0.2%	10.13	17.36	0.717	40.8%		

Table 1. Comparison results of MC, QMC and NPSN w.r.t. STGCNN [40], SGCN [50], SGAN [13], STGAT [17], Trajectron++ [48], and PECNet [35] in $N = 20$. Models are evaluated on the ETH [44], UCY [25], SDD [45], and GCS [60] datasets. (Gain: performance improvement w.r.t. FDE over the baseline models, Unit for ADE and FDE: meter, **Bold**: Best, Underline: Second best)

5. Experiments

In this section, we conduct comprehensive experiments on public benchmark datasets to verify how the sampling strategy contributes to pedestrian trajectory prediction. We first briefly describe our experimental setup (Sec. 5.1), and then provide comparison results with various baselines and state-of-the-art models (Sec. 5.2). Moreover, we run an extensive ablation study to demonstrate the effect of each component of our method (Sec. 5.3).

5.1. Experimental Setup

Dataset. We evaluate the effectiveness of the QMC method and our NPSN on various benchmark datasets [25, 44, 45, 60] over state-of-the-art methods. ETH [44] and UCY dataset [25] include ETH and HOTEL, and UNIV, ZARA1 and ZARA2 scenes, respectively. Both datasets consist of various movements of pedestrians with complicated social interactions. The Stanford Drone Dataset (SDD) [45] contains secluded scenes with various object types (*e.g.* pedestrian, biker, skater, and cart), and the Grand Central Station (GCS) [60] dataset consists of highly congested scenes where pedestrians walk. We observe a trajectory for 3.2 seconds ($T_{obs} = 8$), and then predict future paths for the next 4.8 seconds ($T_{pred} = 12$). We follow a leave-one-out cross-validation evaluation strategy, which is the standard evaluation protocol used in many works [13, 17, 35, 40, 48, 50]. **Evaluation metric.** We measure the performance of the trajectory prediction models using three metrics: 1) Aver-

age Displacement Error (ADE) - average Euclidean distance between a prediction and ground-truth trajectory; 2) Final Displacement Error (FDE) - Euclidean distance between a prediction and ground-truth final destination; 3) Temporal Correlation Coefficient (TCC) [54] - Pearson correlation coefficient of motion patterns between a prediction and ground-truth trajectory. These metrics assess the best one of $N = 20$ trajectory outputs, and we report average values for all agents in each scene. In addition, to reduce the variance in the prediction results of stochastic models, we repeat the evaluation 100 times and then average them for each metric. **Baseline.** We evaluate QMC and NPSN sampling methods with representative stochastic pedestrian trajectory prediction models: 1) Gaussian distribution-based model - Social-STGCNN [40], SGCN [50]; 2) GAN-based model - Social-GAN [13], STGAT [17], Causal-STGAT [6]; 3) CVAE-based model - Trajectron++ [48], PECNet [35], and NCE-Trajectron++ [32]. To validate the effectiveness of QMC and NPSN, we replace their random sampling parts in the authors' provided codes with our QMC and NPSN sampling method.

5.2. Results from QMC and NPSN method

Comparison of MC and QMC. We compare MC with the QMC method by incorporating them into the sampling part of the baseline models. As shown in Tab. 1 and Fig. 5, the QMC method significantly outperforms the MC method on all the evaluation metrics. In Fig. 5, we report the error

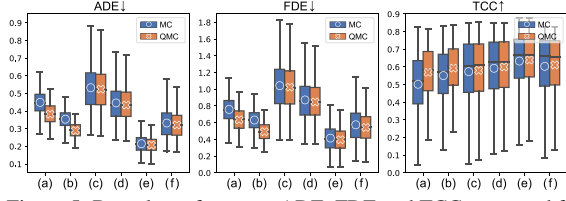


Figure 5. Box plots of average ADE, FDE and TCC measured for each stochastic model on both MC and QMC. (a) STGCNN [40], (b) SGCN [50], (c) SGAN [13], (d) STGAT [17], (e) Trajectron++ [48], and (f) PECNet [35].

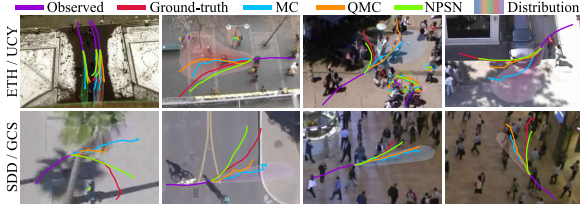


Figure 6. Visualization of probabilistic distributions and the best predictions among sampled trajectories with MC, QMC, and NPSN in SGCN [50].

distributions of the baseline models in the test phase. The QMC method achieves consistently lower errors and variations by alleviating the bias problem mentioned in Sec. 3.2.

We also observe that the Gaussian-based models show a large performance gain over the GAN- and CVAE-based models. There are two reasons for the performance gains induced by the QMC method: 1) The dimension of the sampling space ($s=2$) in the Gaussian-based models is relatively smaller than other models (*i.e.* $s=8, 16$ or 25). According to [24], for large dimensions s and a small number of samples N , the sampling results from a low-discrepancy generator may not be good enough over randomly generated samples. The Gaussian-based model thus yields promising results compared to one which has larger sampling dimensions. 2) The performance improvements depend on the number of layers in networks (shallower is better): The CVAE and GAN-based models are composed of multiple layers. By contrast, the Gaussian-based models have only one layer which acts as a linear transformation between the predicted trajectory coordinates and final coordinates. To be specific, in the transformation, sampled independent 2D points are multiplied with the Cholesky decomposed covariance matrix and shifted by the mean matrix. Here, the shallow layer of the Gaussian-based models directly reflects the goodness of the QMC sampling method, rather than deeper layers which can barely be influenced by the random latent vector in the inference step.

Evaluation of NPSN. We apply NPSN to all three types of stochastic trajectory prediction models. As shown in Tab. 1, there are different performance gains according to the types. Particularly, the Gaussian distribution approaches (Social-

	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Social-GAN [13]	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21
STGAT [17]	0.65 / 1.12	0.35 / 0.66	0.52 / 1.10	0.34 / 0.69	0.29 / 0.60	0.43 / 0.83
Social-STGCNN [40]	0.64 / 1.11	0.49 / 0.85	0.44 / 0.79	0.34 / 0.53	0.30 / 0.48	0.44 / 0.75
PECNet [35]	0.61 / 1.07	0.22 / 0.39	0.34 / 0.56	0.25 / 0.45	0.19 / 0.33	0.32 / 0.56
Trajectron++ [48]	0.43 / 0.86	0.12 / 0.19	0.22 / 0.43	0.17 / 0.32	0.12 / 0.25	0.21 / 0.41
SGCN [50]	0.57 / 1.00	0.31 / 0.53	0.37 / 0.67	0.29 / 0.51	0.22 / 0.42	0.35 / 0.63
Causal-STGAT [6]	0.60 / 0.98	0.30 / 0.54	0.52 / 1.10	0.32 / 0.64	0.28 / 0.58	0.40 / 0.77
NCE-Trajectron++ [33]	0.39 / 0.79	0.11 / 0.18	0.20 / 0.44	0.15 / 0.33	0.11 / 0.26	0.19 / 0.40
NPSN-SGAN	0.72 / 1.26	0.38 / 0.72	0.71 / 1.43	0.34 / 0.68	0.34 / 0.70	0.50 / 0.96
NPSN-STGAT	0.61 / 1.02	0.31 / 0.57	0.53 / 1.13	0.34 / 0.68	0.30 / 0.62	0.42 / 0.80
NPSN-STGCNN	0.44 / 0.65	0.21 / 0.34	0.28 / 0.44	0.25 / 0.43	0.22 / 0.38	0.28 / 0.45
NPSN-PECNet	0.55 / 0.88	0.19 / 0.29	0.29 / 0.44	0.21 / 0.33	0.16 / 0.25	0.28 / 0.44
NPSN-Trajectron++	0.40 / 0.76	0.12 / 0.18	0.21 / 0.41	0.17 / 0.31	0.12 / 0.24	0.20 / 0.38
NPSN-SGCN	0.36 / 0.59	0.16 / 0.25	0.23 / 0.39	0.18 / 0.32	0.14 / 0.25	0.21 / 0.36
NPSN-Causal-STGAT	0.56 / 0.90	0.25 / 0.40	0.51 / 1.09	0.32 / 0.65	0.27 / 0.56	0.38 / 0.72
NPSN-NCE-Trajectron++	0.36 / 0.68	0.11 / 0.18	0.18 / 0.39	0.14 / 0.29	0.11 / 0.23	0.18 / 0.35

Table 2. Comparison of NPSN with other state-of-the-art stochastic model (ADE/FDE, Unit: meter). The evaluation results are directly referred from [6, 33, 48]. **Bold**: Best, Underline: Second Best.

STGCNN [40], SGCN [50]) show the highest performance improvement (up to 60%), which can be analyzed by the advantages of the QMC method when $s=2$. So far, the performance of the Gaussian distribution approaches has been underestimated due to the disadvantage of being easily affected by the sampling bias. Our NPSN maximizes the capability of the Gaussian distribution approaches through a purposive sampling technique.

In the CVAE based approaches, PECNet [35] shows a larger performance improvement (up to 41%) than that of Trajectron++ [48]. Since PECNet directly predicts a set of destinations through the latent vector, NPSN is compatible with its inference step. On the other hand, NPSN seems to produce less benefit with the inference step of Trajectron++ because it predicts the next step recurrently and its sample dimension is relatively large ($s=25$).

The generative models with variety loss, Social-GAN and STGAT, show relatively small performance improvements, compared to the others. For some datasets, the FDE values of STGAT are lower than those of MC and QMC when using our NPSN. This seems to suggest that NPSN fails to learn samples close to ground-truth trajectories due to the common entanglement problem of latent space [7, 20].

Qualitative results. Fig. 6 shows several cases where there are differences between the predictions of NPSN and other methods. Since NPSN takes an observation trajectory along with the low-discrepancy characteristics of the QMC method, the predicted paths from NPSN are closer to socially-acceptable paths compared to other methods.

As we described in the Fig. 4, the QMC method generates a more realistic trajectory distribution than the MC method. However, due to the limitations of the dataset, the generated trajectories of the baseline network are biased toward a straight path. On the other hand, NPSN sampling method alleviates the problem by selecting the point near the ground-truth in the latent space. As a result, the human trajectory model with NPSN not only generates well-distributed samples with finite sampling pathways, but also represents the feasible range of human’s movements.

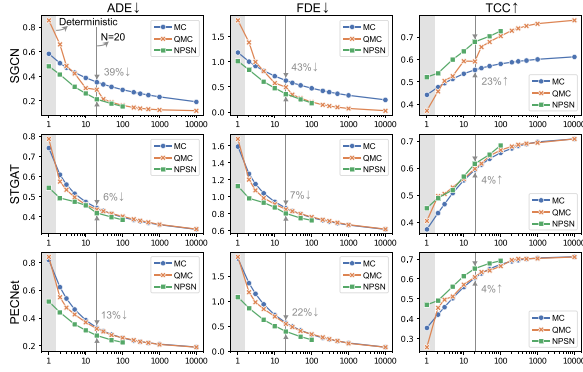


Figure 7. Averaged ADE/FDE/TCC results on ETH/UCY datasets w.r.t. the number of samples for SGCN [50] (Gaussian distribution approach), STGAT [17] (GAN-based approach) and PECNet [35] (CVAE-based approach). We quantify the performance change w.r.t. the sampling methods including MC, QMC and NPSN. We also report the results of deterministic trajectory prediction when $N = 1$ in gray colored regions.

Comparison with the state-of-the-art models. We push the state-of-the-art models with our NPSN, a purposive sampling technique. As shown in Tab. 1, our NPSN shows a significant performance improvement on all the baseline networks. NPSN provides better overall accuracy by taking fully advantage of the low-discrepancy characteristics of the QMC method.

In addition, we report a benchmark result on ETH/UCY dataset in Tab. 2. It is noticeable that all the baseline models exhibit better performances with our NPSN. In particular, when NPSN is incorporated into the combinational approach of Trajectron++ [48] and NCE [32], it achieves the best performances on the benchmark. Our NPSN is trained to only control the latent vector samples for the baseline models, and synergizes well with the inference step that comes after both the initial prediction of Trajectron++ and the collision avoidance of NCE.

5.3. Ablation Studies

Evaluation of different number of samples. To check the effectiveness of the density of sampled paths in human trajectory prediction, we randomly generate trajectories by changing the number of samples N . As shown in Fig. 7, the performance gap between the MC and the QMC method is marginal when the number of samples goes to infinity. As mentioned above, it follows the Strong Law of large numbers in the MC integration. The Gaussian-based model, SGCN [50], achieves superior performance and improves more than 30% performance gain over the classic policy ($N = 20$). Since the sample dimension is small, the effectiveness and convergence of our NPSN are enlarged. Note that a performance drop over sparse conditions due to the discrepancy property: For small N and a comparably large sample space dimension (*i.e.*, $N < 2^8$), the discrepancy of

	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Baseline	0.57 / 1.00	0.31 / 0.53	0.37 / 0.67	0.29 / 0.51	0.22 / 0.42	0.35 / 0.63
w/o \mathcal{L}_{dist}	0.39 / 0.61	0.23 / 0.45	0.26 / 0.47	0.20 / 0.36	0.16 / 0.31	0.25 / 0.44
w/o \mathcal{L}_{disc}	0.38 / 0.61	0.16 / 0.25	0.23 / 0.39	0.18 / 0.32	0.14 / 0.25	0.22 / 0.37
w/o GAT	0.36 / 0.57	0.17 / 0.28	0.23 / 0.39	0.18 / 0.32	0.14 / 0.26	0.22 / 0.37
+NPSN	0.36 / 0.59	0.16 / 0.25	0.23 / 0.39	0.18 / 0.32	0.14 / 0.25	0.21 / 0.36

Table 3. Ablation study on each component of NPSN in SGCN.

the QMC method may not be less than that of a random sequence. We overcome these limitations with a learnable sampling method by sampling a feasible latent vector with low-discrepancy characteristics.

Deterministic trajectory prediction. Since the stochastic model is trained to predict multi-modal future paths, it outputs diverse paths at each execution, which is undesirable for deterministic human trajectory prediction, which infers only one feasible pathway ($N = 1$). By replacing the conventional probability process with a learnable sampling, NPSN allows the stochastic models to infer the most feasible trajectory in a deterministic manner. As shown in Fig. 7 (gray colored regions), NPSN outperforms QMC and the conventional methods on all the metrics at $N = 1$.

Effectiveness of each component. Lastly, we examine the effectiveness of each component in our NPSN, whose result is reported in Tab. 3. Here, SGCN [50] is selected as the baseline model because it shows the most significant performance improvements with NPSN. First, our two loss functions work well. Particularly, the discrepancy loss guarantees sample diversity by generating low-discrepancy samples, and the distance loss enforces generating samples close to the ground-truth trajectory. The GAT captures the agent-aware interaction for socially-acceptable trajectory prediction, except for the secluded ETH scene.

6. Conclusion

In this work, we numerically analyze the limitations of the conventional sampling process in stochastic pedestrian trajectory prediction, by using the concept of discrepancy as a measure of the sampling quality. To overcome this limitation, we then introduce a novel, light-weight and learnable sampling strategy, inspired by the Quasi-Monte Carlo method. Unlike conventional random sampling, our learnable method considers both observations and the social norms of pedestrians in scenes. In addition, our method can be inserted into stochastic pedestrian trajectory predictions as a plug-and-play module. With the proposed learnable method, all of the state-of-art models achieve performance improvements. In particular, the Gaussian-based models show the best results on the benchmark.

Acknowledgement This work is in part supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST), No.2021-0-02068, Artificial Intelligence Innovation Hub), Vehicles AI Convergence Research & Development Program through the National IT Industry Promotion Agency of Korea (NIPA) funded by the Ministry of Science and ICT(No.S1602-20-1001), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2020R1C1C1012635), and the GIST-MIT Collaboration grant funded by the GIST in 2022.

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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2
- [2] Inhwan Bae and Hae-Gon Jeon. Disentangled multi-relational graph convolutional network for pedestrian trajectory prediction. *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021. 3
- [3] Peter Beerli, Deidre Evans, and Michael Mascagni. On the scrambled sobol sequence. *Lecture Notes in Computer Science*, 2005. 4
- [4] Ken Black. *Business statistics: for contemporary decision making*. John Wiley & Sons, USA, 2019. 4
- [5] George E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 1958. 5
- [6] Guangyi Chen, Junlong Li, Jiwen Lu, and Jie Zhou. Human trajectory prediction via counterfactual analysis. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 5, 6, 7
- [7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2016. 7
- [8] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, and Joseph E. Gonzalez. Fbnetv3: Joint architecture-recipe search using predictor pretraining. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [9] Xiaoliang Dai, Peizhao Zhang, Bichen Wu, Hongxu Yin, Fei Sun, Yanghan Wang, Marat Dukhan, Yunqing Hu, Yiming Wu, Yangqing Jia, Peter Vajda, Matt Uyttendaele, and Niraj K. Jha. Chamnet: Towards efficient network design through platform-aware model adaptation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [10] Patrick Dendorfer, Sven Elfle, and Laura Leal-Taixé. Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 1, 2
- [11] Zhi Deng, Yuxin Yao, Bailin Deng, and Juyong Zhang. A robust loss for point cloud registration. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 3
- [12] William Feller. *An introduction to probability theory and its applications. Vol. II*. John Wiley & Sons Inc., New York, 1971. 3
- [13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 5, 6, 7
- [14] J. H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM*, 1964. 4
- [15] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 1
- [16] Lee Howes and David Thomas. *GPU Gems 3 - Efficient Random Number Generation and Application Using CUDA*. Pearson Education, Inc, 2007. 5
- [17] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 5, 6, 7, 8
- [18] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 1, 2
- [19] Stephen Joe and Frances Y. Kuo. Constructing sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 2008. 4
- [20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7
- [21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 3
- [22] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezaeifighi, and Silvio Savarese. Socialbigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 3
- [23] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2
- [24] Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer, Dordrecht, 2009. 7
- [25] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007. 4, 6
- [26] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2019. 1
- [27] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 3
- [28] Yuke Li. Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5
- [29] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from simulation for trajectory prediction. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 3

- [30] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3
- [31] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [32] Yuejiang Liu, Qi Yan, and Alexandre Alahi. Social nce: Contrastive learning of socially-aware motion representations. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 5, 6, 8
- [33] Yuejiang Liu, Qi Yan, and Alexandre Alahi. Social nce: Contrastive learning of socially-aware motion representations. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 7
- [34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2018. 5
- [35] 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 *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 5, 6, 7, 8
- [36] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1
- [37] George Marsaglia and Wai Wan Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 2000. 5
- [38] Martin N Marshall. Sampling for qualitative research. *Family Practice*, 1996. 4
- [39] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1
- [40] Abdullah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 5, 6, 7
- [41] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, USA, 1992. 3
- [42] Art B. Owen. Quasi-monte carlo sampling. *Elsevier*, 2003. 4
- [43] Spassimir H. Paskov and Joseph F. Traub. Faster valuation of financial derivatives. *The Journal of Portfolio Management*, 1995. 4
- [44] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2009. 1, 4, 6
- [45] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. 6
- [46] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 5
- [47] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezaatofghi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2
- [48] Tim Salzman, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 5, 6, 7, 8
- [49] Nasim Shafiee, Taskin Padir, and Ehsan Elhamifar. Introvert: Human trajectory prediction via conditional 3d attention. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1
- [50] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcnn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 5, 6, 7, 8
- [51] Xiaodan Shi, Xiaowei Shao, Zipei Fan, Renhe Jiang, Haoran Zhang, Zhiling Guo, Guangming Wu, Wei Yuan, and Ryosuke Shibasaki. Multimodal interaction-aware trajectory prediction in crowded space. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 1, 2
- [52] Hao Sun, Zhiqun Zhao, and Zhihai He. Reciprocal learning networks for human trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2
- [53] Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3
- [54] Chaofan Tao, Qinhong Jiang, and Lixin Duan. Dynamic and static context-aware lstm for multi-agent motion prediction. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 6
- [55] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. 3
- [56] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 1, 2
- [57] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 3

- [58] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 3
- [59] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1
- [60] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Understanding pedestrian behaviors from stationary crowd groups. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6
- [61] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3
- [62] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agent-former: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 2, 5
- [63] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2
- [64] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning (CoRL)*, 2020. 1, 2
- [65] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [66] Xiuzhuang Zhou and Yao Lu. Abrupt motion tracking via adaptive stochastic approximation monte carlo sampling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 3
- [67] Giray Ökten and Ahmet Göncü. Generating low-discrepancy sequences from the normal distribution: Box-muller or inverse transform? *Mathematical and Computer Modelling*, 2011. 5