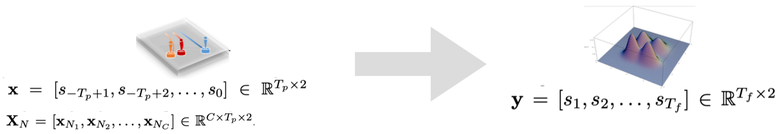
Stochastic Trajectory Generation with Diffusion via Implicit Maximizing Likelihood Estimation Distillation

Yuxiang Fu [1,2] Qi Yan [1,2] Lele Wang [1] Ke Li[4] Renjie Liao[1,2,3]

[1]Univeresity of British Columbia [2]Vector Institute for AI [3]Canada CIFAR AI Chair [4]Simon Fraser University

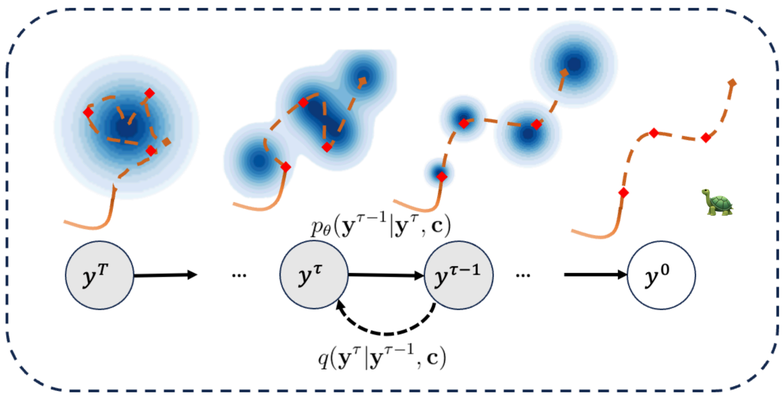
Background

The objective of stochastic trajectory generation: Given the past trajectory history $\mathbf{x}, \mathbf{X}_N$ of the agents (only sequences of 2D coordinates), we want to model the distribution of the future trajectories $\mathbf{y} \sim P_\theta$ jointly so that the generated samples are socially and physically compliant.



Note that C represents the number of agents in a scene and T_p, T_f are the no. of past and future frames resp.

Limitations



- The inference of traditional DDPM [1] is extremely **slow**.
- Current distillation methods suffer from training instability and mode collapse [5], sometimes with multiple retraining phases involved [4].
- Most methods [4, 3] fail to attain good quality of samples from teacher model.
- Some method [2] incorporates deterministic "distilling" step.

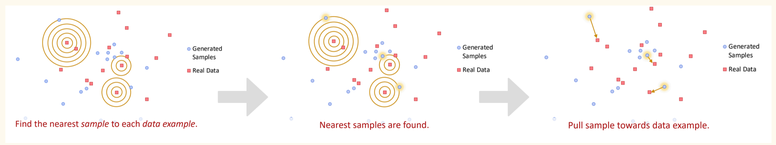
Implicit Maximum Likelihood Estimator

The Implicit Maximum Likelihood Estimator is defined as

$$\hat{\theta}_{\text{IMLE}} := \arg\min_\theta \mathbb{E}_{\tilde{\mathbf{x}}_1^\theta, \ldots, \tilde{\mathbf{x}}_m^\theta} \left[\sum_{i=1} \min_{j \in [m]} \|\tilde{\mathbf{x}}_j^\theta - \mathbf{x}_i\|_2^2 \right]$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are our data examples and $\tilde{\mathbf{x}}_1^\theta, \ldots, \tilde{\mathbf{x}}_m^\theta$ denotes the i.i.d. samples from P_θ.
Key process behind IMLE:

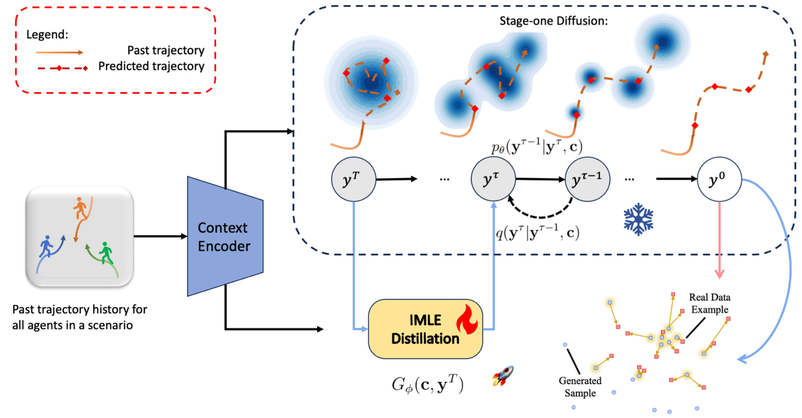
1. Generate a batch of i.i.d. samples so that there are more samples than the number of data examples
2. Search for the nearest sample to **EACH** data example
3. Adjust the parameters so that the nearest sample is pulled by each data example



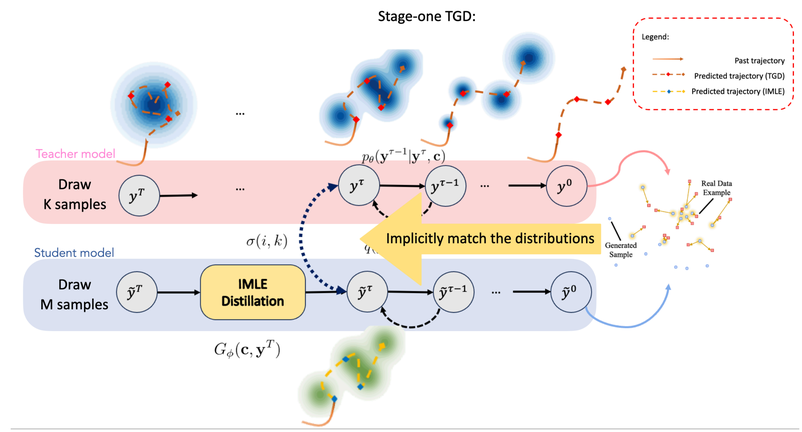
Contributions

1. We design a stage-one trajectory stage-one trajectory generation diffusion model (TGD) tailored for stochastic trajectory generation tasks.
2. We propose a trainable student model to directly match the intricate distribution through the IMLE scheme at any intermediate diffusion timestamp, improving prediction performance and inference time.
3. Due to the IMLE scheme, **no more Mode Collapse; Vanishing Gradients; Training Instability** occur.
4. Our methods deliver results that are competitive with SOTA methods.

IMLE distillation



We distill a large number of denoising steps $(T - \tau)$ with a flexible IMLE module where $T >> \tau$. Let's take a closer look at the IMLE module.



Our IMLE distillation model is able to reconstruct the distribution of diffusion latent at timestamp τ (shown in Green) in the teacher model.

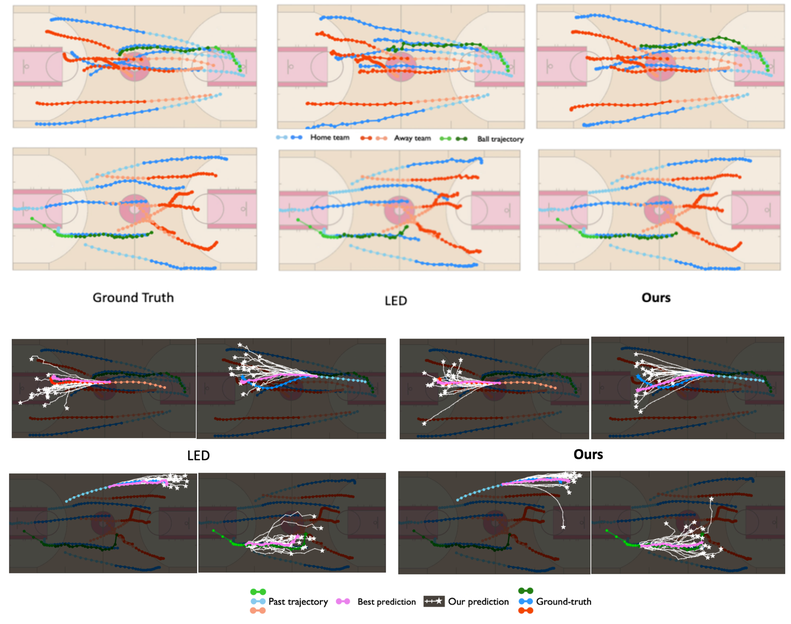
NBA dataset

The dataset is collected by NBA officially via SportVU tracking system, which records the trajectories of 10 players and a ball in a real basketball play-off.

- Past trajectory: 10 frames (2.0s) [2D coord sequence in Euclidean space]. Future ground truth: 20 frames (4.0s)
- It has ~358K trajectories for training and ~137K trajectories for testing

Qualitative Assessments

The first row and second row in each plot represent two distinct scenes.



Teacher TGD Results

Time	MemoNet [33] CVPR'22	NPSN [34] CVPR'22	GroupNet [6] CVPR'22	MID [5] CVPR'22	LED† [10] CVPR'23	TGD (Ours) Transformer-ϵ-cos	TGD UNet1D-y^0-cos	TGD UNet2D-y^0-cos
1.0s	0.38/0.56	0.35/0.58	0.26/0.34	0.28/0.37	0.21/**0.28**	0.19/0.29	**0.189**/0.29	0.19/0.29
2.0s	0.71/1.14	0.68/1.23	0.49/0.70	0.51/0.72	0.44/0.64	0.42/0.65	**0.41**/0.64	0.41/**0.63**
3.0s	1.00/1.57	1.01/1.76	0.73/1.02	0.71/0.98	0.69/0.95	0.68/1.01	**0.65**/0.94	0.66/**0.93**
Total (4.0s)	1.25/1.47	1.31/1.79	0.96/1.30	0.96/1.27	0.94/1.21	0.95/1.38	**0.89**/1.19	0.91/**1.19**

Figure 1. The methods in bold are TGD with distinct backbones, prediction objectives and variance schedule.

Student IMLE distillation Results

Time	LED [10] (initializer)	IMLE-TF (Ours)	IMLE-UNet1D (Ours)	WGAN [35]*	DCGAN [36]*
1.0s	0.18/0.27/2.49/3.15	0.19/0.30/0.50/0.95	0.20/0.30/**0.48/0.93**	0.33/0.52/0.50/0.86	0.32/0.58/0.47/0.87
2.0s	0.37/0.56/2.51/2.41	0.41/0.63/1.16/2.43	0.42/0.64/**1.13/2.36**	0.75/1.43/0.99/1.89	0.78/1.57/1.00/1.98
3.0s	0.58/0.84/2.75/3.73	0.64/0.96/1.87/3.82	0.65/0.97/**1.82**/3.70	1.23/2.34/1.49/2.84	1.28/2.47/1.53/2.94
Total (4.0s)	0.81/1.14/3.13/4.55	0.89/1.31/2.51/4.78	0.89/1.31/**2.44**/4.65	1.69/2.91/1.95/3.63	1.75/3.05/2.00/3.74

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

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

[3] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020.

[4] Jonathan Ho Tim Salimans. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021.

[5] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022.