Stochastic Trajectory Generation with Diffusion via Implicit Maximizing Likelihood Estimation Distillation

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

[1]University of British Columbia, [2]Vector Institute, [3]Simon Fraser University

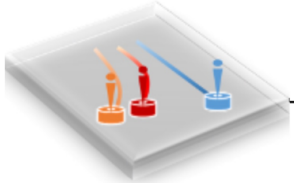
THE UNIVERSITY OF BRITISH COLUMBIA

VECTOR INSTITUTE | INSTITUT VECTEUR

SIMON FRASER UNIVERSITY

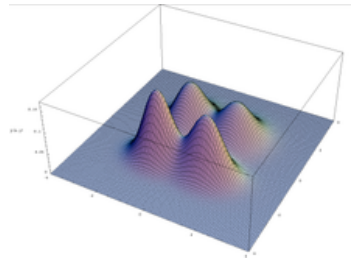
Stochastic Trajectory Prediction

Given the past trajectory history of the agents (only sequences of 2D coordinates), we want to model the distribution of the future trajectories jointly so that the generated samples are socially and physically compliant.



$$\mathbf{x} = [s_{-T_p+1}, s_{-T_p+2}, \ldots, s_0] \in \mathbb{R}^{T_p \times 2}$$

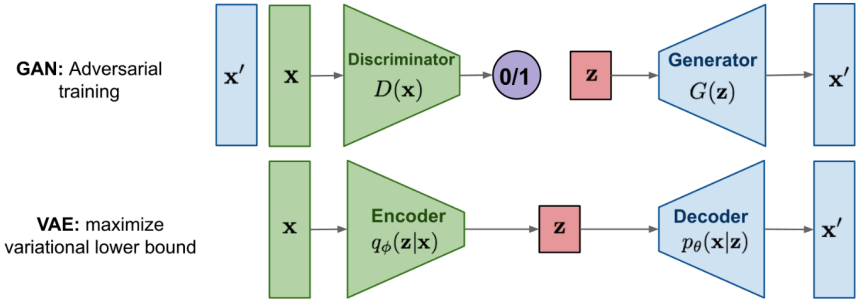
$$\mathbf{X}_N = [\mathbf{x}_{N_1}, \mathbf{x}_{N_2}, \ldots, \mathbf{x}_{N_C}] \in \mathbb{R}^{C \times T_p \times 2}$$



$$\mathbf{y} = [s_1, s_2, \ldots, s_{T_f}] \in \mathbb{R}^{T_f \times 2}$$

Previous works for trajectory generation modelling:

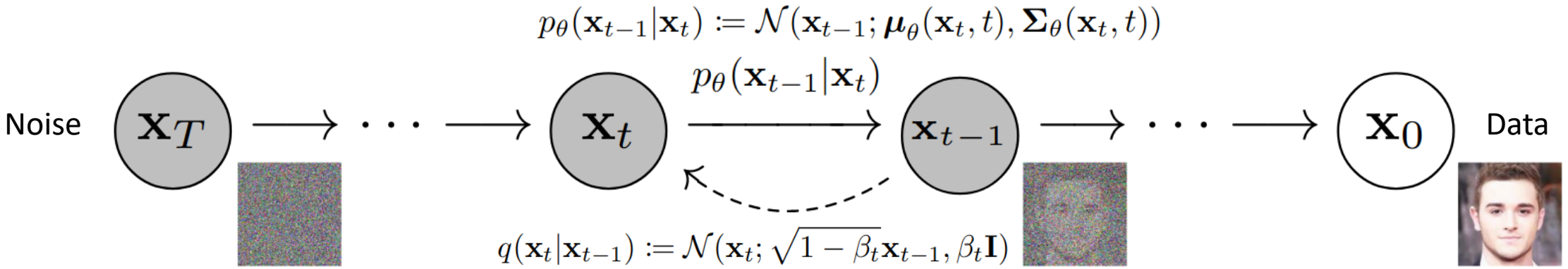
1. GAN [1]
2. Conditional VAEs [2]



[1] P. Dendorfer, S. Elflein, and L. Leal-Taix´e, “Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction,” CVPR’21

[2] M. Lee, S. S. Sohn, S. Moon, S. Yoon, M. Kapadia, and V. Pavlovic, “Muse-vae: Multi-scale vae for environment-aware long term trajectory prediction,” CVPR’22

Denoising Diffusion Probabilistic Model



DDPMs consist of two processes:

- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising

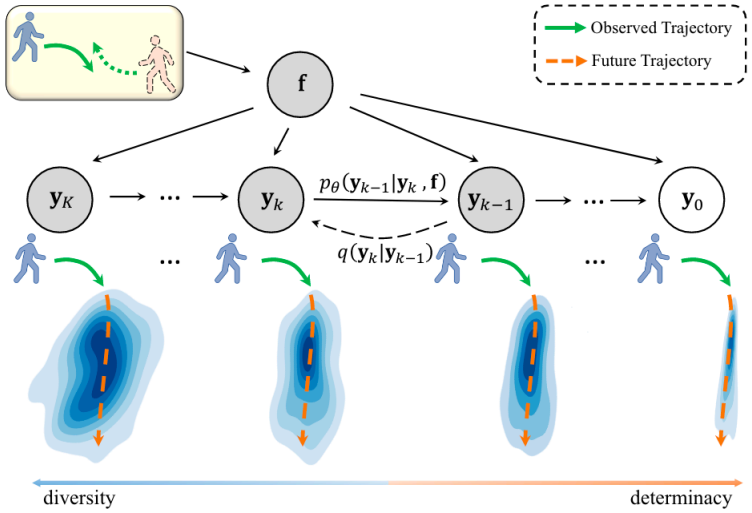
⚠Limitation of DDPM: Inference is very slow due to the iterative denoising steps.

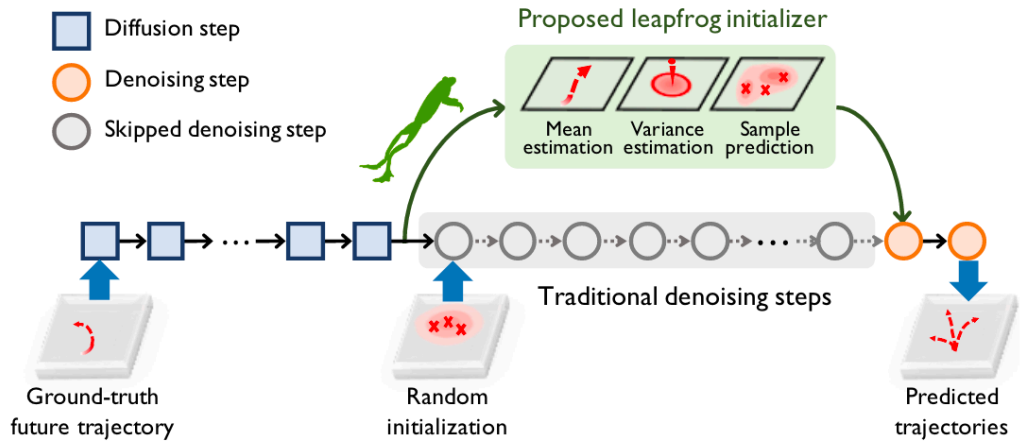
[Ho et al, Denoising diffusion probabilistic models, NeurIPS'20]

Diffusion-based stochastic motion prediction

- **MID**[1] first uses diffusion model on human trajectory dataset

- **LED**[2] proposes an initializer to accelerate the sampling process by "distilling" a large number of denoising steps.



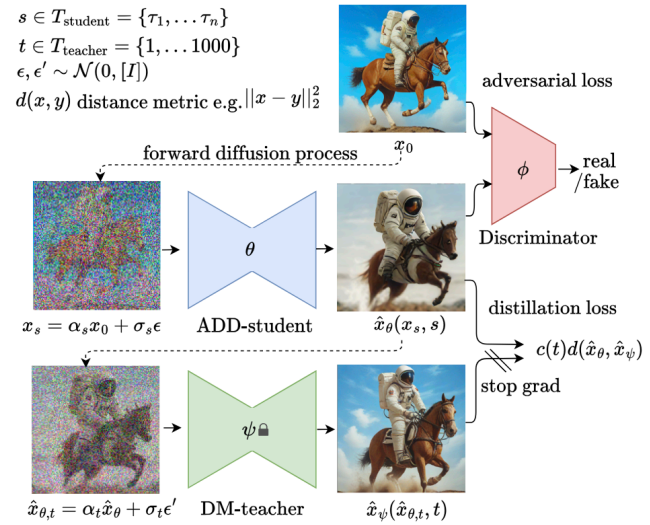


[1] W. Mao, C. Xu, Q. Zhu, S. Chen, and Y. Wang, "Leapfrog diffusion model for stochastic trajectory prediction," CVPR'23

[2] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," CVPR'22

Recent Distillation/Acceleration techniques

- Adversarial Distillation Diffusion model[1]
- Progressive distillation[2] and DDIM[3]
- LED initializer[4]
- On distillation of guided diffusion[5]



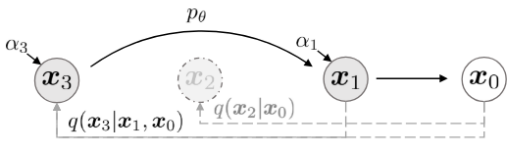


Figure 2: Graphical model for accelerated generation, where $\tau = [1, 3]$.

However, these methods suffer from

- training instability & mode collapse
- multiple retraining phases involved
- failing to attain the quality of samples generated from teacher model
- inappropriate distilling process 😞

[1] ADD Stability AI'23
[2] PD ICLR'22
[3] DDIM ICLR'21
[4] LED CVPR'23
[5] On distillation of guided diffusion CVPR'23

Implicit Maximizing likelihood Estimation

- The implicit maximum likelihood estimator is defined as

$\mathbf{x}_1, \ldots, \mathbf{x}_n$: Data examples

$\tilde{\mathbf{x}}_1^\theta, \ldots, \tilde{\mathbf{x}}_m^\theta$: i.i.d. samples from P_θ

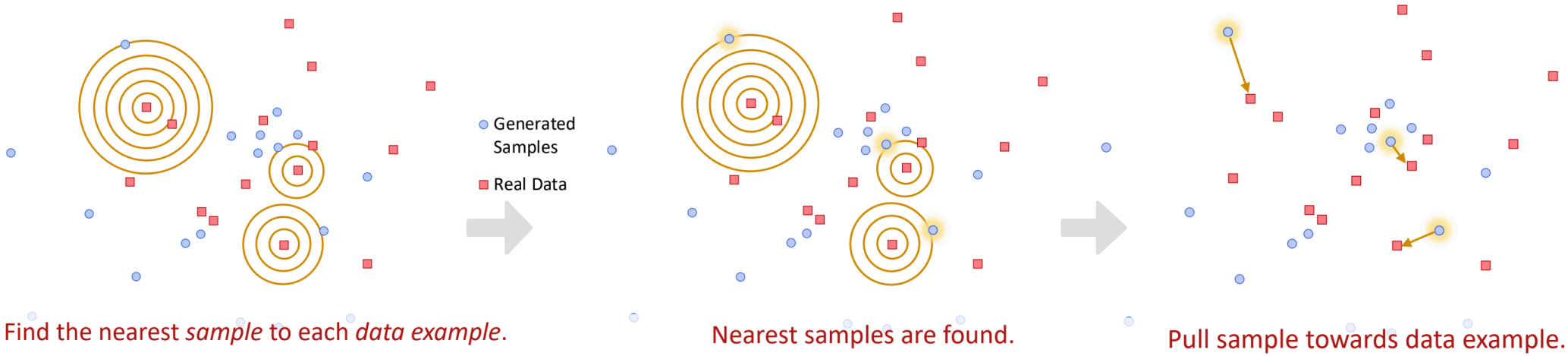
$$\hat{\theta}_{\text{IMLE}} := \arg\min_\theta \mathbb{E}_{\tilde{\mathbf{x}}_1^\theta, \ldots, \tilde{\mathbf{x}}_m^\theta} \left[\sum_{i=1}^{n} \min_{j \in [m]} \left\| \tilde{\mathbf{x}}_j^\theta - \mathbf{x}_i \right\|_2^2 \right]$$

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. arXiv: 1809.09087, 2018

Key ideas behind IMLE

Simple ideas to conduct principled distillation:

1. Generate a batch of i.i.d. samples (more samples than the no. 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



K. Li and J. Malik. Implicit Maximum Likelihood Estimation. arXiv: 1809.09087, 2018

Natural benefits of IMLE

- **No Mode Collapse ✅**
  - Each data example has a nearby sample
- **No more Vanishing Gradients ✅**
  - Gradient becomes zero when the distance to the nearest sample is zero
- **No more Training Instability ✅**
  - Simple minimization problem

K. Li and J. Malik. Implicit Maximum Likelihood Estimation. arXiv: 1809.09087, 2018

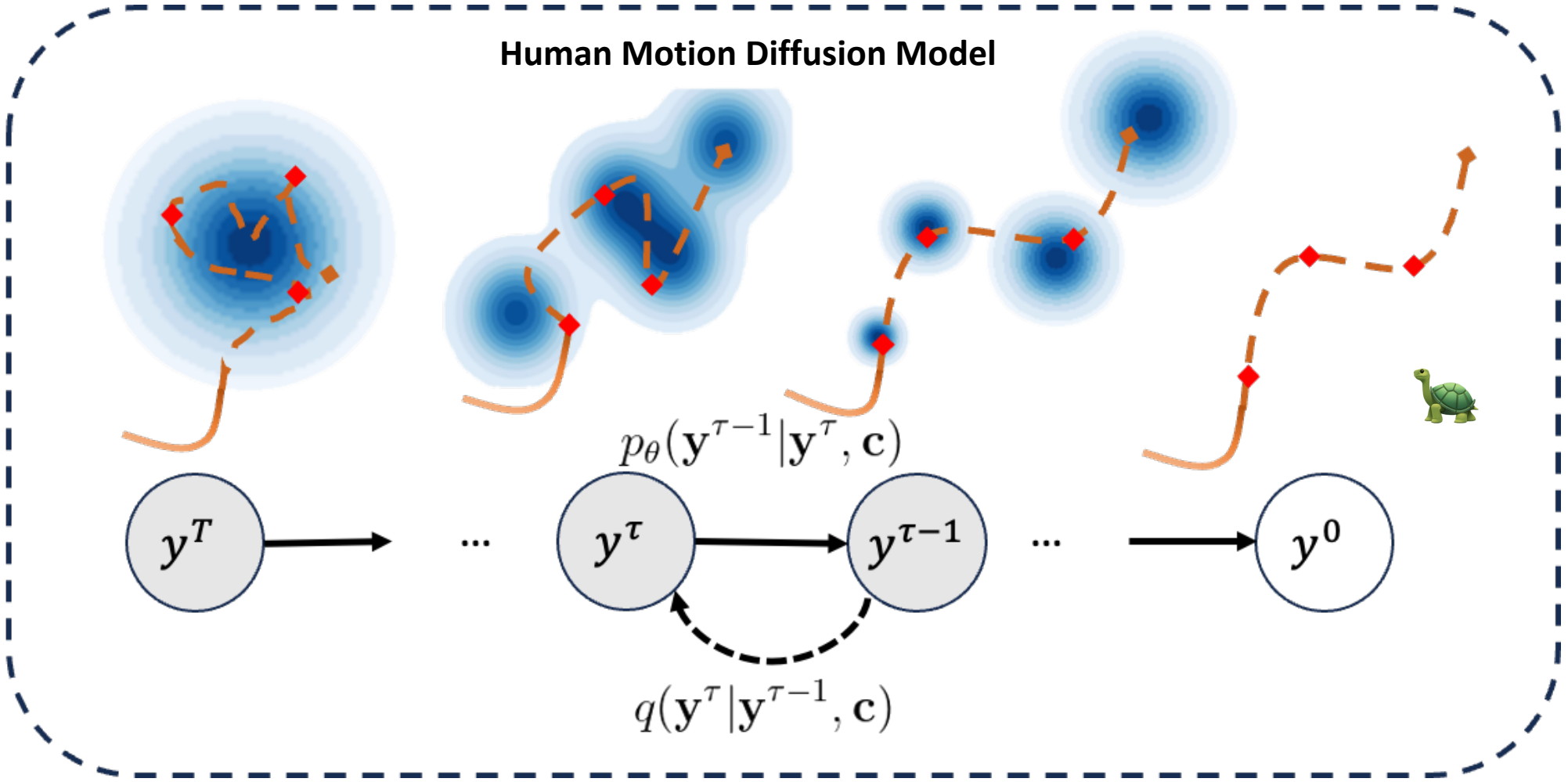
Motivations

- Diffusion models have emerged as powerful generative models, **yet** the practical application of these models is hindered by their high computational costs during inference.

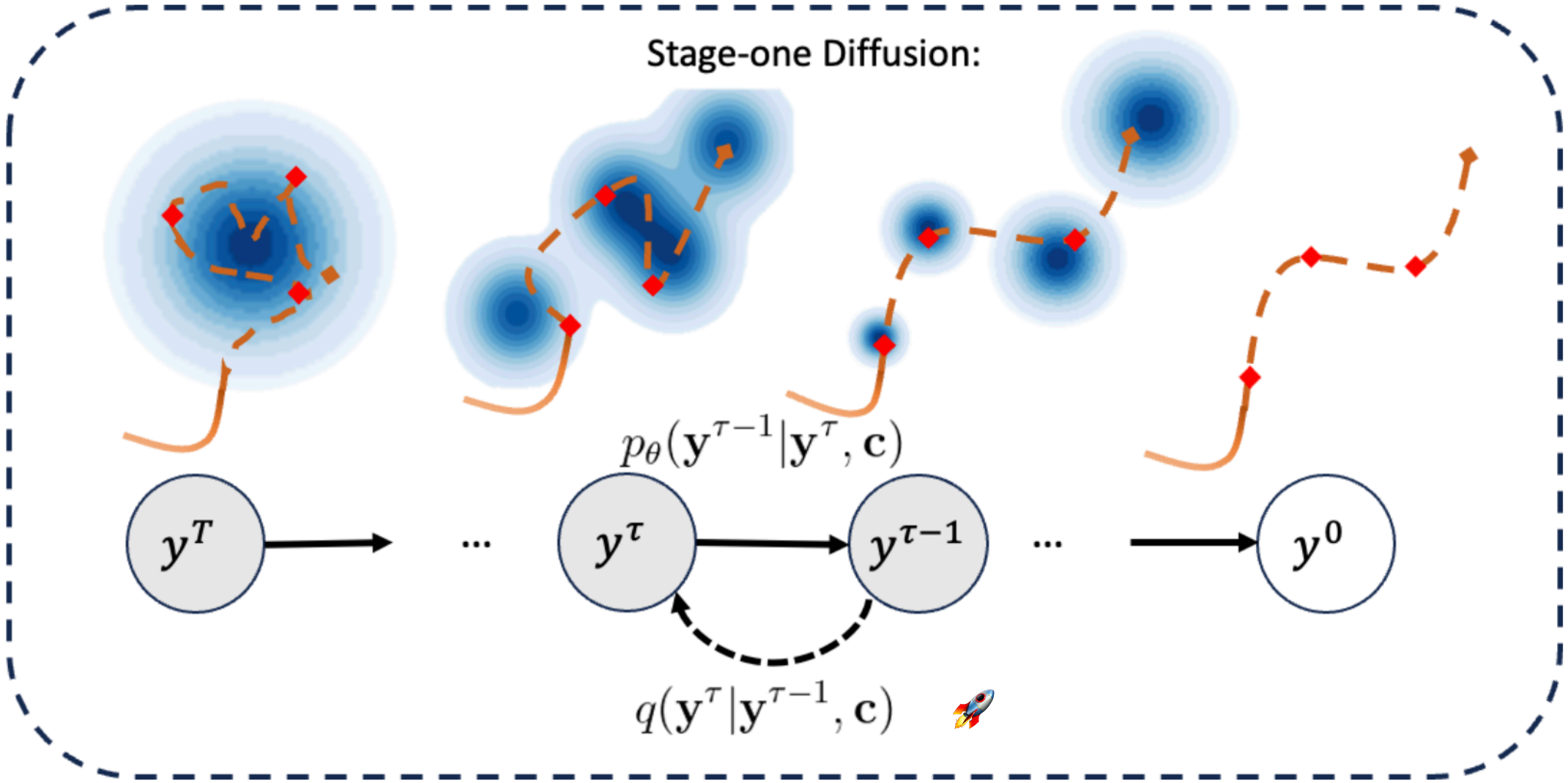
- Current distillation techniques have notable drawbacks for the trajectory prediction task, and sometimes involve deterministic process that undermines their reliability.

- IMLE is a **simple** and **strong** method to match distributions without any knowledge of the likelihood function.

Trajectory Generation Diffusion Stage One

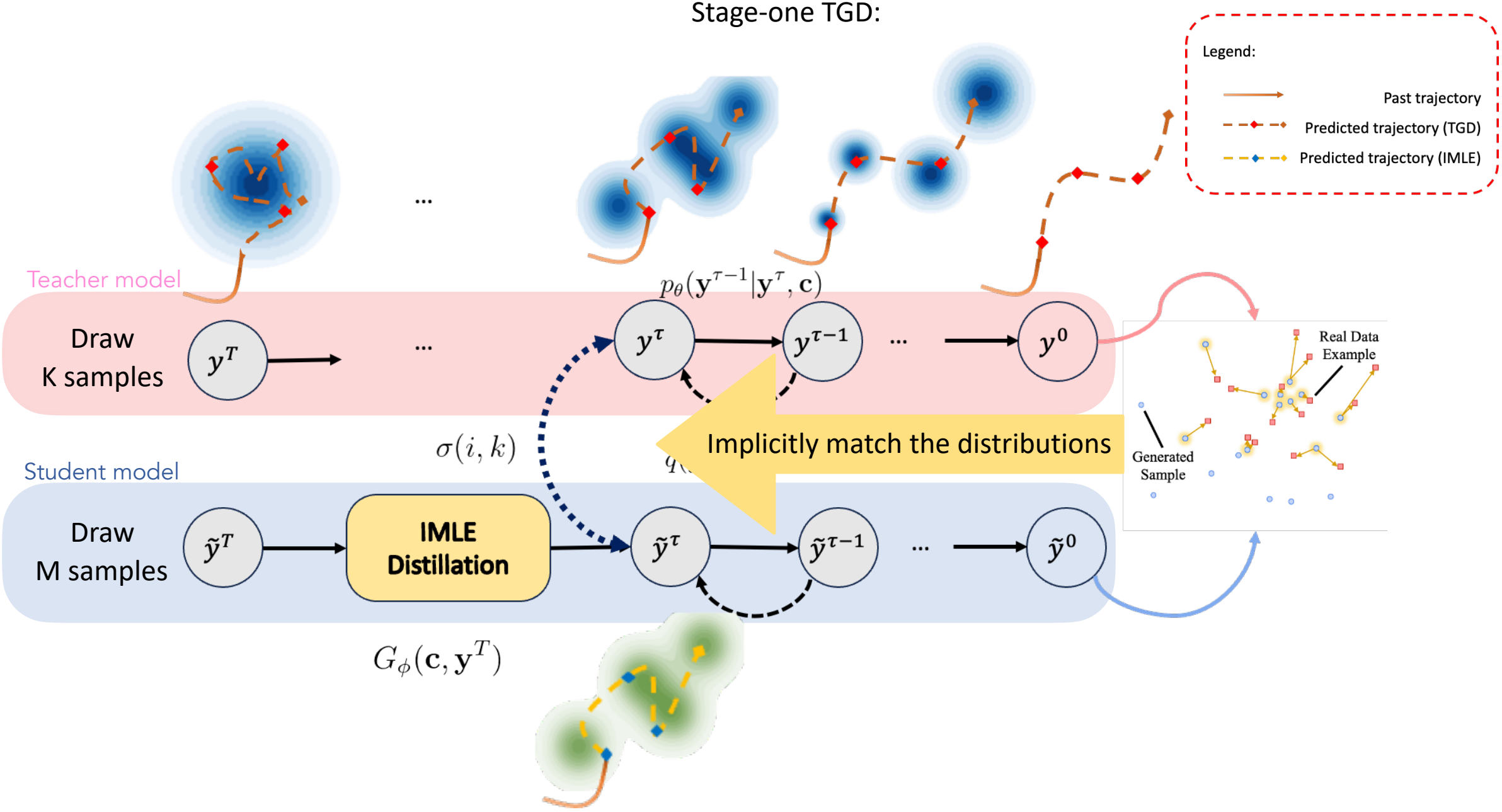


IMLE distillation process (stage-two)

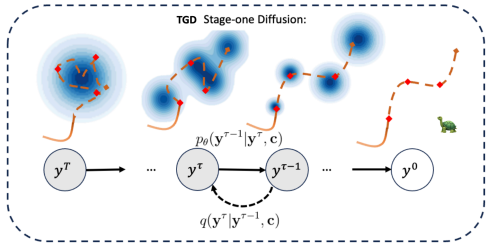


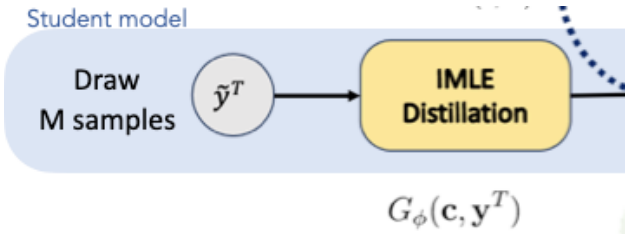
Approach

Stage-one TGD:



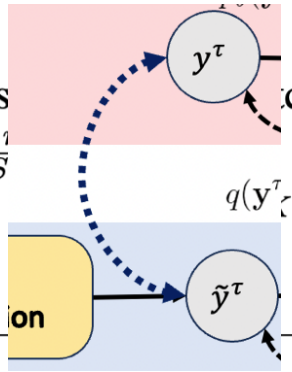
IMLE algorithm explained

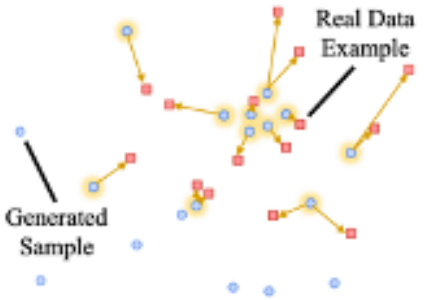




Algorithm 2 Distilling pre-trained DDPM with τ-IMLE

Require: Number of samples $m > K$ that IMLE needs to draw, the set of observed context embeddings $C = \{\mathbf{c}_i\}_{i=1}^n$, time step τ, the set of future clear samples from TGD stage-one model $\hat{Y}^0 = \{\mathbf{y}_{i,k}^0\}_{i,k=1}^{n,K}$ based on C, time step τ with future noisy trajectory targets $\hat{Y}^\tau$
Initialize the parameters ϕ of the network G_ϕ
repeat
  for $i = 1, \ldots, n$ **do**
    Generate m i.i.d vectors $\mathbf{z}_1, \ldots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\tilde{\mathbf{y}}_{i,j}^\tau \leftarrow G_\phi(\mathbf{c}_i, \mathbf{z}_j)$ for all $j \in [m]$
    Denoise $\tilde{\mathbf{y}}_{i,j}^\tau$ for remaining τ steps to obtain $\tilde{\mathbf{y}}_{i,j}^0$
    $\sigma(i,k) \leftarrow \arg\min_j \mathcal{L}(\mathbf{y}_{i,k}^0, \tilde{\mathbf{y}}_{i,j}^0)$ for all $k \in [K]$
  end for
  loop ω times
    Randomly s ch $S_m \subseteq [n]$
    $\phi \leftarrow \phi - \frac{}{|S}$ $\mathcal{L}(\mathbf{y}_{i,k}^\tau, \tilde{\mathbf{y}}_{i,\sigma(i,k)}^\tau)$
  end loop
until converged





SportUV NBA dataset 🏀

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

- Number of agents is 11
- Past trajectory: 10 frames (2.0s) [2D coords sequence in Euclidean space]
- Future ground truth: 20 frames (4.0s)

It has ~358K trajectories for training and ~137K trajectories for testing

Metrics

- Minimum Final Displacement Error (minADE): The L2 distance between the endpoint of the ***best*** forecasted trajectory (among K samples) and the ground truth.
- Minimum Average Displacement Error (minFDE): The average L2 distance between the ***best*** forecasted trajectory and the ground truth. The ***best*** here refers to the trajectory that has the minimum endpoint error.
- Average ADE: Now the ***best*** here is the average of the predicted trajectories
- Average FDE: Same definition of ***best*** above

Results on NBA dataset

Table I: Comparison with baseline models on NBA dataset. $\min_{20}$ADE/$\min_{20}$FDE (meters) are reported. Bold/underlined fonts represent the best/second-best result. The methods in bold are TGD with distinct backbones, training targets, and variance schedule. For LED†, we use the stage one result from their network.

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**

TGD stage-one

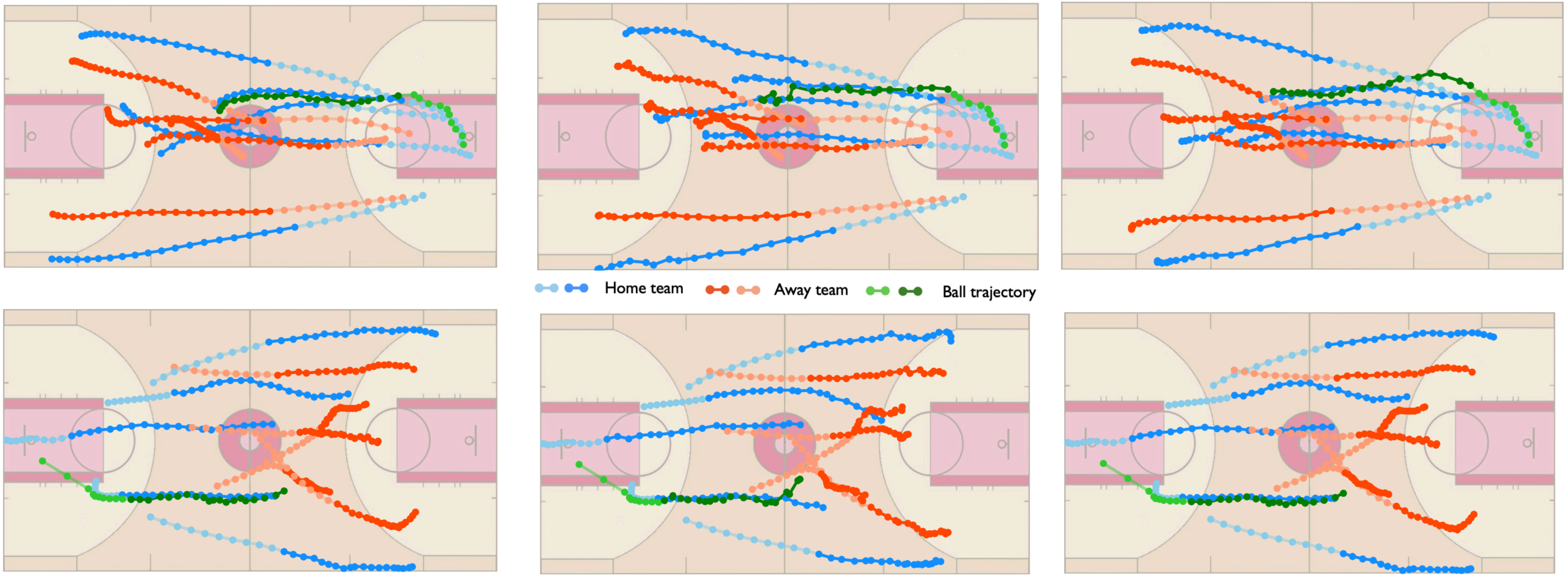
Results on SportUV NBA dataset

Table II: Comparison with baseline distillation models on NBA dataset. min_{20}ADE/min_{20}FDE/avg_{20}ADE/avg_{20}FDE (meters) are reported. The asterisk* denotes that the method uses our stage one diffusion model, which is on par with the SOTA stage one model. All the models learn to generate samples conforming to the marginal distribution from teacher diffusion latents at time step $\tau = 5$.

Time	LED [10] (initializer)	**IMLE-TF (Ours)**	**IMLE-UNet1D (Ours)**	**WGAN***	**DCGAN***
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

IMLE distillation method

Qualitative results

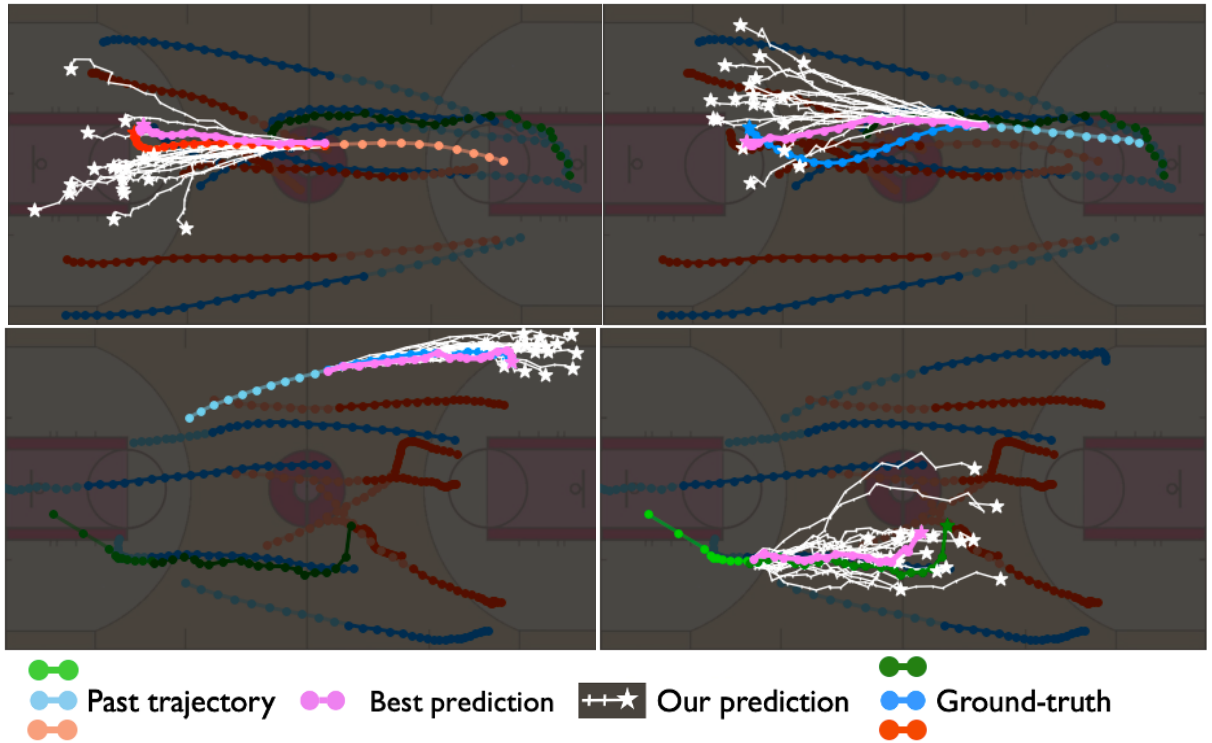


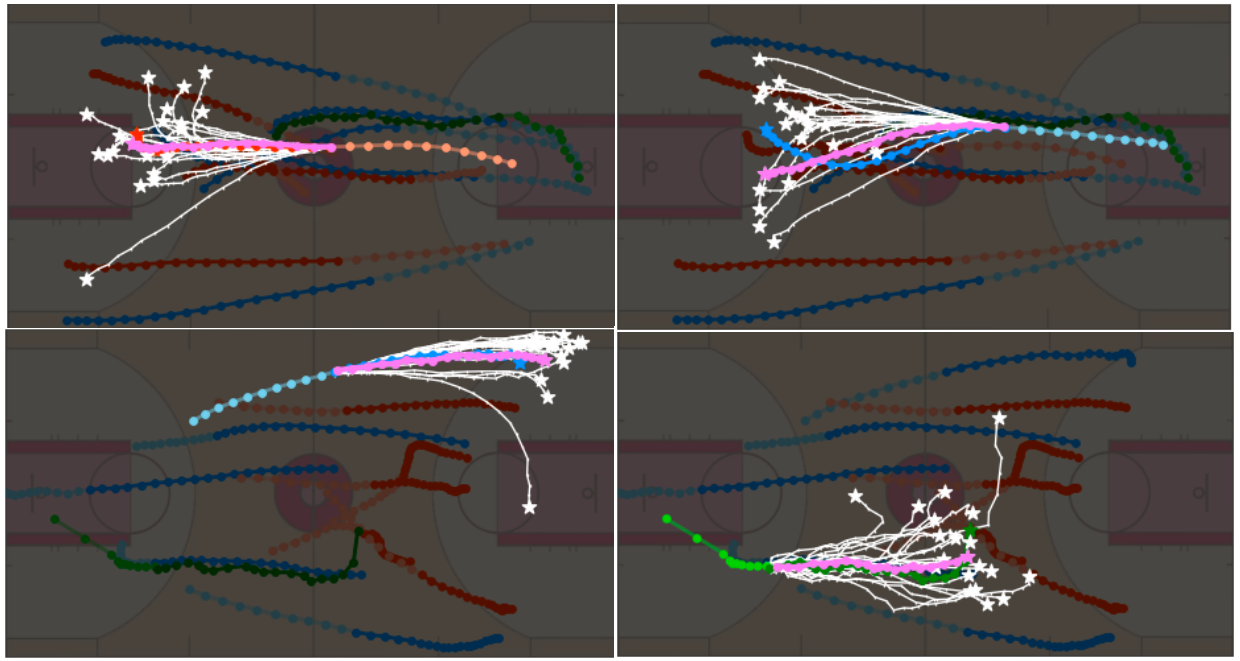
Ground Truth

LED

Ours

Diversity qualitative results

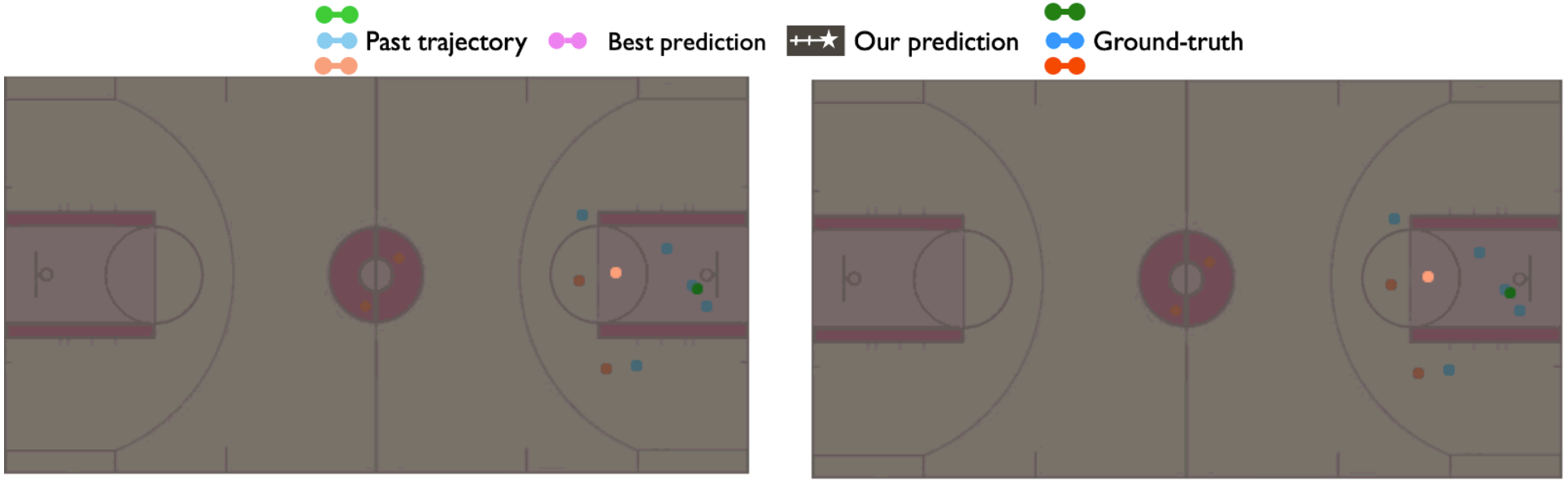




LED

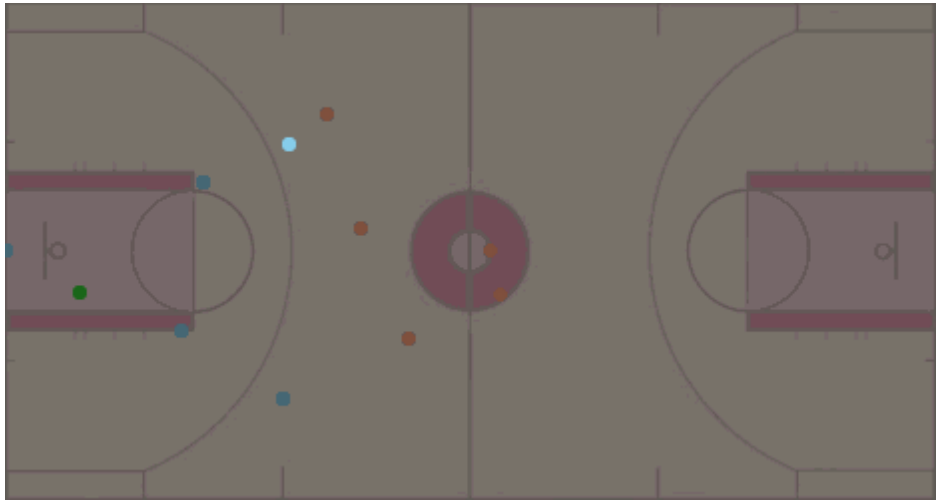
Ours

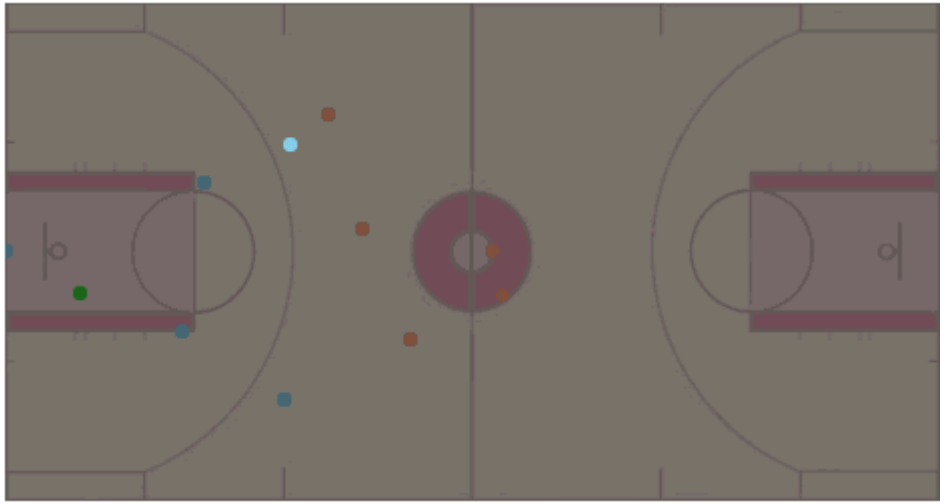
Some Animations



LED

Ours





Conclusion

- In this paper, we introduce a flexible IMLE distillation model for trajectory generation task, which efficiently matching the multi-modal distribution of future trajectories at any intermediate diffusion timestamp.

- We extensively study its performances on the real-world NBA SportUV dataset. Experimental results show that our stage one TGD and IMLE model deliver results that are competitive with SOTA methods.

Future Work

- Future works incorporate validating this idea on other pedestrian trajectory datasets and self-driving datasets. Some erroneous cases can be avoided by encoding the map information.

- Record the accelerated inference time and benchmark other methods on the same GPU hardware.

Thank you!