Supplementary: Leapfrog Diffusion Model for Stochastic Trajectory Prediction

Weibo Mao[1], Chenxin Xu[1], Qi Zhu[1], Siheng Chen[1,2*], Yanfeng Wang[1,2],
[1]Shanghai Jiao Tong University, [2]Shanghai AI Laboratory
{kirino.mao,xcxwakaka,georgezhu,sihengc,wangyanfeng}@sjtu.edu.cn

1. Detailed Derivations

1.1. Derivation of Standard Diffusion Models

In the paper submission, we present a standard diffusion model for trajectory prediction following the diffusion-denoising process. Here we elaborate on the details of the diffusion-denoising process.

In standard diffusion models, the diffusion process is operated on the future trajectory $\mathbf{Y}$, while the past trajectories $\mathbf{X}$ and $\mathbb{X}$ serve as a condition for the denoising process. Mathematically, let $\mathbf{Y}^\gamma$ be the diffused future trajectory at step γ, being a basic state in the bidirectional Markov chain of the diffusion-denoising process. We have the start state $\mathbf{Y}^0 = \mathbf{Y}$ and the end state $\mathbf{Y}^\Gamma \sim \mathcal{N}(\mathbf{Y}^\Gamma; 0, \mathrm{I})$. We restate the overall procedure of diffusion models for trajectory prediction here, following

$$\mathbf{Y}^0 = \mathbf{Y}, \tag{1a}$$

$$\mathbf{Y}^\gamma = f_{\text{diffuse}}(\mathbf{Y}^{\gamma-1}),\ \gamma = 1, \cdots, \Gamma, \tag{1b}$$

$$\widehat{\mathbf{Y}}_k^\Gamma \overset{i.i.d}{\sim} \mathcal{P}(\widehat{\mathbf{Y}}^\Gamma) = \mathcal{N}(\widehat{\mathbf{Y}}^\Gamma; \mathbf{0}, \mathbf{I}), \text{sample } K \text{ times}, \tag{1c}$$

$$\widehat{\mathbf{Y}}_k^\gamma = f_{\text{denoise}}(\widehat{\mathbf{Y}}_k^{\gamma+1}, \mathbf{X}, \mathbb{X}_\mathcal{N}),\ \gamma = \Gamma-1, \cdots, 0, \tag{1d}$$

where we use the $f_{\text{diffuse}}(\cdot)$ to represent the diffusion process and $f_{\text{denoise}}(\cdot)$ to represent the conditional denoising process. Here we present the details of these two processes.

Forward diffusion process. Let $(\mathbf{Y}^0, \mathbf{Y}^1, \ldots, \mathbf{Y}^\Gamma)$ be the forward Γ-steps Markov chain constructed by the diffusion model where $\mathbf{Y}^\gamma$ is the diffused future trajectory at step γ. The forward diffusion process between two steps is defined as

$$\begin{aligned} q(\mathbf{Y}^\gamma|\mathbf{Y}^{\gamma-1}) &= \mathcal{N}(\mathbf{Y}^\gamma; \sqrt{1-\beta_\gamma}\mathbf{Y}^{\gamma-1}, \beta_\gamma \mathbf{I}), \\ \Rightarrow \mathbf{Y}^\gamma &= \sqrt{1-\beta_\gamma}\mathbf{Y}^{\gamma-1} + \sqrt{\beta_\gamma}\mathbf{z}, \end{aligned}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; 0, \mathbf{I})$ and $\beta_1, \beta_2, \ldots, \beta_\Gamma$ are the diffusion parameters controlling the distortion between two steps. In the forward diffusion process, we can directly sample γth step diffused trajectory $\mathbf{Y}^\gamma$ directly using

$$\begin{aligned} \mathbf{Y}^\gamma &= \sqrt{1-\beta_\gamma}\mathbf{Y}^{\gamma-1} + \sqrt{\beta_\gamma}\mathbf{z} \\ &\overset{\alpha_\gamma := 1-\beta_\gamma}{=\!=\!=} \sqrt{\alpha_\gamma}\,\mathbf{Y}^{\gamma-1} + \sqrt{1-\alpha_\gamma}\mathbf{z} \\ &= \sqrt{\alpha_\gamma}\,(\sqrt{\alpha_{\gamma-1}}\,\mathbf{Y}^{\gamma-2} + \sqrt{1-\alpha_{\gamma-1}}\mathbf{z}') + \sqrt{1-\alpha_\gamma}\mathbf{z} \\ &\overset{\text{reparam.}}{=\!=\!=} \sqrt{\alpha_\gamma}\sqrt{\alpha_{\gamma-1}}\,\mathbf{Y}^{\gamma-2} + \sqrt{1-\alpha_\gamma\alpha_{\gamma-1}}\mathbf{z}'' \\ &= \cdots \\ &= \sqrt{\bar{\alpha}_\gamma}\mathbf{Y}^0 + \sqrt{1-\bar{\alpha}_\gamma}\mathbf{z}, \end{aligned}$$

where we set the diffusion parameter $\alpha_\gamma := 1 - \beta_\gamma$ and use the reparameterization to merge two Gaussian distributions. Note that the forward process is non-trainable and with sufficient steps, the final state $\mathbf{Y}^\Gamma \sim q(\mathbf{Y}^\Gamma)$ will be approximate to sample in a normal distribution, i.e., $\mathbf{Y}^\Gamma \sim \mathcal{N}(\mathbf{Y}^\Gamma; \mathbf{0}, \mathbf{I})$.

Conditional denoising process. Conversely, denote $(\widehat{\mathbf{Y}}^\Gamma, \widehat{\mathbf{Y}}^{\Gamma-1}, \ldots, \widehat{\mathbf{Y}}^0)$ as the reverse denoising process conditioned on context information extracting from past trajectories, i.e. $\mathbf{C} = f_{\text{condition}}(\mathbf{X}, \mathbb{X})$. We formulate the conditional denoising process as follows:

$$\begin{aligned} &p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma, \mathbf{C}) = \mathcal{N}(\mathbf{Y}^{\gamma-1}; \boldsymbol{\mu}_\theta^\gamma(\mathbf{Y}^\gamma, \mathbf{C}), \beta_\gamma \mathbf{I}), \\ &\Rightarrow \widehat{\mathbf{Y}}^{\gamma-1} = \boldsymbol{\mu}_\theta^\gamma(\widehat{\mathbf{Y}}^\gamma, \mathbf{C}) + \sqrt{\beta_\gamma}\mathbf{z}, \\ &\boldsymbol{\mu}_\theta^\gamma(\widehat{\mathbf{Y}}^\gamma, \mathbf{C}) = \frac{1}{\sqrt{\alpha_\gamma}}(\widehat{\mathbf{Y}}^\gamma - \frac{\beta_\gamma}{\sqrt{1-\bar{\alpha}_\gamma}}\boldsymbol{\epsilon}_\theta^\gamma(\widehat{\mathbf{Y}}^\gamma, \mathbf{C})) \end{aligned} \tag{2}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, $\alpha_\gamma = 1 - \beta_\gamma$ and $\bar{\alpha}_\gamma = \prod_{\tau=1}^{\gamma} \alpha_\gamma$ are the diffusion parameters at step γ, and $\boldsymbol{\mu}_\theta(\cdot) \in \mathbb{R}^{T_\text{f} \times 2}$ is the core denoising module with the learnable parameters θ. Note that we have specified the mean term and simplified the variance term in Eq.(2) following DDPM [1] so that we can derive the noise estimation loss.

1.2. Derivation of Noise Estimation Loss

Here we elaborate on the derivation of our noise estimation loss, the overall target of diffusion models is to maximize the $p_\theta(\mathbf{Y}^0|\mathbf{C})$.

*Corresponding author.

$$
\begin{aligned}
&-\log p_\theta(\mathbf{Y}^0|\mathbf{C}) \\
&\leq -\log p_\theta(\mathbf{Y}^0|\mathbf{C}) + \mathrm{D}_{\mathrm{KL}}(q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)||p_\theta(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0,\mathbf{C})) \\
&= -\log p_\theta(\mathbf{Y}^0|\mathbf{C}) \\
&\quad + \int_{\mathbf{Y}^{1:\Gamma}} [\log \frac{q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)}{p_\theta(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0,\mathbf{C})}] q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)\, d\mathbf{Y}^{1:\Gamma} \\
&= -\log p_\theta(\mathbf{Y}^0|\mathbf{C}) + \mathbb{E}_{\mathbf{Y}^{1:\Gamma}\sim q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)} [\log \frac{q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)}{p_\theta(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0,\mathbf{C})}] \\
&= -\log p_\theta(\mathbf{Y}^0|\mathbf{C}) + \mathbb{E}_q [\log \frac{q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)}{p_\theta(\mathbf{Y}^{0:\Gamma}|\mathbf{C})} + \log p_\theta(\mathbf{Y}^0|\mathbf{C})] \\
&= \mathbb{E}_{q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)} [\log \frac{q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)}{p_\theta(\mathbf{Y}^{0:\Gamma}|\mathbf{C})}]
\end{aligned}
$$

where we derive the variational lower bound (VLB) to minimize the negative log-likelihood.

$$
\begin{aligned}
&\Rightarrow \mathbb{E}_{q(\mathbf{Y}^0)} - \log p_\theta(\mathbf{Y}^0|\mathbf{C}) \leq \mathbb{E}_{q(\mathbf{Y}^{0:\Gamma})} [\log \frac{q(\mathbf{Y}^{1:\Gamma}|\mathbf{Y}^0)}{p_\theta(\mathbf{Y}^{0:\Gamma}|\mathbf{C})}] \\
&= \mathbb{E}_q [\sum_{\gamma=1}^{\Gamma} - \log \frac{p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{C})}{q(\mathbf{Y}^\gamma|\mathbf{Y}^{\gamma-1})} - \log p_\theta(\mathbf{Y}^\Gamma)] \\
&= -\mathbb{E}_q [\sum_{\gamma=2}^{\Gamma} \log \frac{p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{C})}{q(\mathbf{Y}^\gamma|\mathbf{Y}^{\gamma-1})} \\
&\quad + \log \frac{p_\theta(\mathbf{Y}^0|\mathbf{Y}^1,\mathbf{C})}{q(\mathbf{Y}^1|\mathbf{Y}^0)} + \log p_\theta(\mathbf{Y}^\Gamma)] \\
&= -\mathbb{E}_q [\sum_{\gamma=2}^{\Gamma} \log \Big(\frac{p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{C})}{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0)} \cdot \frac{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^0)}{q(\mathbf{Y}^\gamma|\mathbf{Y}^0)} \Big) \\
&\quad + \log \frac{p_\theta(\mathbf{Y}^0|\mathbf{Y}^1,\mathbf{C})}{q(\mathbf{Y}^1|\mathbf{Y}^0)} + \log p_\theta(\mathbf{Y}^\Gamma)] \\
&= -\mathbb{E}_q [\sum_{\gamma=2}^{\Gamma} \log \frac{p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{C})}{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0)} + \sum_{\gamma=2}^{\Gamma} \log \frac{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^0)}{q(\mathbf{Y}^\gamma|\mathbf{Y}^0)} \\
&\quad + \log \frac{p_\theta(\mathbf{Y}^0|\mathbf{Y}^1,\mathbf{C})}{q(\mathbf{Y}^1|\mathbf{Y}^0)} + \log p_\theta(\mathbf{Y}^\Gamma)] \\
&= -\mathbb{E}_q [\frac{p_\theta(\mathbf{Y}^\Gamma)}{q(\mathbf{Y}^\Gamma|\mathbf{Y}^0)} + \sum_{\gamma=2}^{\Gamma} \log \frac{p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{C})}{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0)} \\
&\quad + \log p_\theta(\mathbf{Y}^0|\mathbf{Y}^1,\mathbf{C})]
\end{aligned}
$$

where the first term can be ignored since there are no trainable parameters in $p_\theta(\mathbf{Y}^\Gamma)$. Then, we only need to focus on the second term $\mathbb{E}_q[\log \frac{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0)}{p_\theta(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{C})}]$ where $p_\theta(\cdot)$ is given in Equation (2). We can derive the close form for $q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0)$ with the Bayes' rule,

$$
\begin{aligned}
q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0) &= \frac{q(\mathbf{Y}^{\gamma-1},\mathbf{Y}^\gamma|\mathbf{Y}^0)}{q(\mathbf{Y}^\gamma|\mathbf{Y}^0)} \\
&= \frac{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^0) q(\mathbf{Y}^\gamma|\mathbf{Y}^{\gamma-1},\mathbf{Y}^0)}{q(\mathbf{Y}^\gamma|\mathbf{Y}^0)} \\
&= \frac{q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^0) q(\mathbf{Y}^\gamma|\mathbf{Y}^{\gamma-1})}{q(\mathbf{Y}^\gamma|\mathbf{Y}^0)}
\end{aligned}
$$

where $q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^0), q(\mathbf{Y}^\gamma|\mathbf{Y}^{\gamma-1})$, and $q(\mathbf{Y}^\gamma|\mathbf{Y}^0)$ are all Gaussian distributions, which indicates the target distribution $q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0)$ also has the Gaussian form. Follow [1], suppose $q(\mathbf{Y}^{\gamma-1}|\mathbf{Y}^\gamma,\mathbf{Y}^0) = \mathcal{N}(\mathbf{Y}^{\gamma-1};\boldsymbol{\mu}^\gamma,\beta_\gamma\mathbf{I})$, where

$$
\begin{aligned}
\boldsymbol{\mu}^\gamma &= \frac{\sqrt{\alpha_\gamma}(1-\bar{\alpha}_{\gamma-1})}{1-\bar{\alpha}_\gamma}\mathbf{Y}^\gamma + \frac{\sqrt{\bar{\alpha}_{\gamma-1}}\beta_\gamma}{1-\bar{\alpha}_\gamma}\mathbf{Y}^0 \\
&= \frac{1}{\sqrt{\alpha}_\gamma}(\mathbf{Y}^\gamma - \frac{\beta_\gamma}{\sqrt{1-\bar{\alpha}_\gamma}}\boldsymbol{\epsilon})
\end{aligned}
$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})$. Then, we only need to minimize the means between two distributions and get the noise estimation loss:

$$
\mathcal{L}_{\mathrm{NE}} = \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta^\gamma(\widehat{\mathbf{Y}}^\gamma,\mathbf{C})\|_2,
$$

2. Experiment Details

We apply a standard diffusion model with the diffusion step $\Gamma = 100$, the start value β_1 =1e-4, and the end value β_{100} =5e-2. We use the linear schedule to interpolate the intermediate values $\beta_2, \beta_3, \cdots, \beta_{99}$.

On SDD, following previous destination prediction strategies [2, 3], we first predict the destination of a pedestrian using the proposed leapfrog diffusion model. And then, we fulfill the trajectory using the multi-layer perceptron.

3. Supplementary Experiments

3.1. Influence of Diffusion Parameters

We explore the influence of different parameters in the diffusion model, including the denoising steps Γ, the start value of β_1, the end value of β_Γ, and the schedule to generate β's; see Table 1. We see that i) $\beta_1 = 1e-4, \beta_\Gamma = 5e-2, \Gamma = 100$ provides the best performance for the standard diffusion model; ii) with the fixed β_1 and β_Γ, the schedule to generate the intermediate parameters will not influence the performance lot, also the linear schedule provides the best performance; and iii) when the diffusion step Γ is too small, the denoising step is not equivalent to estimating the Gaussian noise, deteriorating the performance.

3.2. Influence of Different Encoders

In the leapfrog initializer, we use a social encoder to capture social influence, a temporal encoder to learn temporal

Table 1. Influence of different parameters in the standard diffusion models on SDD. We run 5 times for each setting with K=20 and report the average and best performance.

Diffusion Parameters				AVG		Best	
β_1	β_Γ	Γ	schedule	minADE	minFDE	minADE	minFDE
1e-4	5e-2	20	linear	19.27	32.77	10.42	19.19
		50	linear	11.04	17.75	9.94	15.95
		100	linear	**10.36**	16.92	**9.73**	**15.32**
			sigmoid	10.65	**16.87**	9.76	15.52
			quadratic	10.55	17.87	9.84	15.77
		200	linear	10.70	18.03	10.24	16.98
		500	linear	10.94	18.68	10.45	17.68
		1000	linear	11.27	19.01	10.91	18.26
1e-5	5e-2	100	linear	10.43	17.45	9.92	16.03
1e-4	1e-2	100	linear	25.80	48.29	12.70	21.37
1e-5	1e-2	100	linear	26.91	44.85	12.52	21.06

Table 2. Influence of different social-temporal structures in the leapfrog initializer on SDD. We run 5 times for each setting with K=20 and report the average and best performance.

Encoder Structure	AVG		Best	
	minADE	minFDE	minADE	minFDE
without social	8.69	12.07	8.64	11.93
sequential	8.65	11.94	8.60	11.81
parallel	**8.47**	**11.54**	**8.46**	**11.47**

embedding, and an aggregation layer to fuse both social and temporal information. Here we explore the influence of different encoders including without considering the social information (without social), sequential structure to fuse the social-temporal information (sequential), and the parallel structure used in the paper submission (parallel); see Table 2. We see that i) the parallel structure provides the best performance since the social-temporal information is decoupled without influencing each other; and ii) the social force will influence the agent's movement since considering the social embedding outperforms the without social embedding structure.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 1, 2

[2] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, pages 759–776, 2020. 2

[3] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2022. 2