Stochastic Trajectory Prediction via Motion Indeterminacy Diffusion
Supplementary Material

A. Detailed Derivations

A.1. Derivations of Loss Function

We give the derivations to obtain our loss function as:

$$\begin{aligned}
L(\theta,\psi) &= \mathbb{E}_q[\sum_{k=1}^{K} -\log \frac{p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f})}{q(\mathbf{y}_k|\mathbf{y}_{k-1})}] \\
&= -\mathbb{E}_q[\sum_{k=2}^{K} \log \frac{p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f})}{q(\mathbf{y}_k|\mathbf{y}_{k-1})} + \log \frac{p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f})}{q(\mathbf{y}_1|\mathbf{y}_0)}] \\
&= -\mathbb{E}_q[\sum_{k=2}^{K} \log p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f}) \frac{q(\mathbf{y}_{k-1}|\mathbf{y}_0)}{q(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{y}_0)q(\mathbf{y}_k|\mathbf{y}_0)} \\
&\quad + \log \frac{p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f})}{q(\mathbf{y}_1|\mathbf{y}_0)}] \\
&= -\mathbb{E}_q[\sum_{k=2}^{K} \log \frac{p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f})}{q(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{y}_0)} + \sum_{k=2}^{K} \log \frac{q(\mathbf{y}_{k-1}|\mathbf{y}_0)}{q(\mathbf{y}_k|\mathbf{y}_0)} \\
&\quad + \log \frac{p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f})}{q(\mathbf{y}_1|\mathbf{y}_0)}] \\
&= \mathbb{E}_q[\sum_{k=2}^{K} -\log \frac{p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f})}{q(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{y}_0)} \\
&\quad - \log p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f}) + \log q(\mathbf{y}_K|\mathbf{y}_0)].
\end{aligned} \tag{1}$$

We ignore the last term because it has no learnable parameters and get the loss function as:

$$\begin{aligned}
L(\theta,\psi) &= \mathbb{E}_q[\sum_{k=2}^{K} D_{KL}(q(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{y}_0)\|p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f})) \\
&\quad - \log p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f})].
\end{aligned} \tag{2}$$

A.2. Derivations of Reparameterization

As shown in the loss function, we should match the reverse transition $p_\theta(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{f})$ and the ground-truth $q(\mathbf{y}_{k-1}|\mathbf{y}_k,\mathbf{y}_0)$, both of which are in Gaussian. We can convert the KL divergence of two Gaussian distributions as the difference of the means. We calculate the mean of posterior in a closed form:

$$\tilde{\boldsymbol{\mu}}_k(\mathbf{y}_k,\mathbf{y}_0) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\mathbf{y}_0 + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\mathbf{y}_k, \tag{3}$$

where $\alpha_k = 1-\beta_k$ and $\bar{\alpha}_k = \prod_{s=1}^{k}\alpha_s$. By the reparameterization, we formulate the $\mathbf{y}_k$ as a function of $\mathbf{y}_0$ and ϵ:

$$\mathbf{y}_k(\mathbf{y}_0,\epsilon) = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{4}$$

where $\epsilon \sim \mathcal{N}(0,\mathbf{I})$ is a random variable, and we have

$$\mathbf{y}_0 = \frac{1}{\sqrt{\bar{\alpha}_k}}(\mathbf{y}_k(\mathbf{y}_0,\epsilon) - \sqrt{1-\bar{\alpha}_k}\epsilon). \tag{5}$$

Then we reformulate $\tilde{\boldsymbol{\mu}}_k(\mathbf{y}_k,\mathbf{y}_0)$:

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_k(\mathbf{y}_k(\mathbf{y}_0,\epsilon),\epsilon) &= (\frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{\sqrt{\bar{\alpha}_k}(1-\bar{\alpha}_k)} + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k})\mathbf{y}_k(\mathbf{y}_0,\epsilon) \\
&\quad + \frac{\sqrt{1-\bar{\alpha}_k}\sqrt{\bar{\alpha}_{k-1}}\beta_k}{\sqrt{\bar{\alpha}_k}(1-\bar{\alpha}_k)}\epsilon \\
&= \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k + \sqrt{\bar{\alpha}_k}\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{\sqrt{\bar{\alpha}_k}(1-\bar{\alpha}_k)}\mathbf{y}_k(\mathbf{y}_0,\epsilon) \\
&\quad - \frac{\beta_k}{\sqrt{\alpha_k}\sqrt{1-\bar{\alpha}_k}}\epsilon \\
&= \frac{\sqrt{\bar{\alpha}_{k-1}}(\beta_k + \alpha_k(1-\bar{\alpha}_{k-1}))}{\sqrt{\bar{\alpha}_k}(1-\bar{\alpha}_k)}\mathbf{y}_k(\mathbf{y}_0,\epsilon) \\
&\quad - \frac{\beta_k}{\sqrt{\alpha_k}\sqrt{1-\bar{\alpha}_k}}\epsilon \\
&= \frac{1}{\sqrt{\alpha_k}}\mathbf{y}_k(\mathbf{y}_0,\epsilon) - \frac{\beta_k}{\sqrt{\alpha_k}\sqrt{1-\bar{\alpha}_k}}\epsilon \\
&= \frac{1}{\sqrt{\alpha_k}}(\mathbf{y}_k(\mathbf{y}_0,\epsilon) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon).
\end{aligned} \tag{6}$$

Therefore, the D_{KL} can be formulated as:

$$\begin{aligned}
D_{KL} &= \mathbb{E}_{\mathbf{y}_0,\epsilon}\left[\lambda\|\tilde{\boldsymbol{\mu}}_k(\mathbf{y}_k(\mathbf{y}_0,\epsilon),\epsilon) - \boldsymbol{\mu}_\theta(\mathbf{y}_k,k,\mathbf{f})\|^2\right] \\
&= \mathbb{E}_{\mathbf{y}_0,\epsilon}\left[\lambda\left\|\frac{1}{\sqrt{\alpha_k}}(\mathbf{y}_k(\mathbf{y}_0,\epsilon) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon) - \boldsymbol{\mu}_\theta(\mathbf{y}_k,k,\mathbf{f})\right\|^2\right],
\end{aligned} \tag{7}$$

Then we show why the last term $-\log p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f})$ is tractable with the same formulation form of D_{KL} at $k=1$. The term $-\log p_\theta(\mathbf{y}_0|\mathbf{y}_1,\mathbf{f})$ means that the outputs of prediction model should follow the distribution of real data.

Algorithm 1 Pseudocode for MID Training Procedure

```
repeat
  Sample trajectory (x, y) ~ q_data
  x : Observed Trajectory
  y : Future Trajectory
  y_0 = y
  k ~ Uniform({1, ..., K})
  ε ~ N(0, I)
  Take gradient descent step on
      ∇_(θ,ψ) ||ε − ε_(θ,ψ)(√(ᾱ_k) y_0 + √(1 − ᾱ_k) ε, k, x)||^2
until converged
```

Algorithm 2 Pseudocode for MID Sampling Procedure

```
Input: Observed Trajectory x
Output: Predicted Trajectory y
Sample y_K ~ N(0, I)
for k = K, ..., 1 do
  z ~ N(0, I) if k > 1, else z = 0
  y_{k−1} = (1/√α_k) (y_k − (β_k/√(1−ᾱ_k)) ε_(θ,ψ)(y_k, k, x)) + √β_k z
end for
y = y_0
return y
```

Considering the reverse transition $p_\theta(\mathbf{y}_0|\mathbf{y}_1, \mathbf{f})$ is Gaussian, we also revert this loss as the difference between the mean of Gaussian transition $\boldsymbol{\mu}_\theta(\mathbf{y}_k, k, \mathbf{f})$ and the ground truth $\mathbf{y}_0$ as $\mathbb{E}\left[\lambda\|\mathbf{y}_0 - \boldsymbol{\mu}_\theta(\mathbf{y}_1, 1, \mathbf{f})\|^2\right]$. Moreover, for the D_{KL} under $k = 1$, we have

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_1(\mathbf{y}_1(\mathbf{y}_0, \epsilon), \epsilon) &= \frac{1}{\sqrt{\alpha_1}}(\mathbf{y}_1(\mathbf{y}_0, \epsilon) - \frac{\beta_1}{\sqrt{1-\bar{\alpha}_1}}\epsilon) \\ &= \frac{1}{\sqrt{\bar{\alpha}_1}}(\mathbf{y}_1(\mathbf{y}_0, \epsilon) - \sqrt{1-\bar{\alpha}_1}\epsilon).\end{aligned} \tag{8}$$

With (5), we get $\tilde{\boldsymbol{\mu}}_1(\mathbf{y}_1(\mathbf{y}_0, \epsilon), \epsilon) = \mathbf{y}_0$, which demonstrates the of losses with both $k = 1$ and $k \geq 2$ are in the same form.

As shown in (6) and (7), the loss function expects the model to predict $\frac{1}{\sqrt{\alpha_k}}(\mathbf{y}_k(\mathbf{y}_0, \epsilon) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon)$ given the inputs $\mathbf{y}_k(\mathbf{y}_0, \epsilon)$ and $\mathbf{f}$. Since the $\mathbf{y}_k(\mathbf{y}_0, \epsilon)$ is the input, we only need a network to predict ϵ as $\epsilon_\theta(\mathbf{y}_k(\mathbf{y}_0, \epsilon), k, \mathbf{f})$. Thus, the final loss function is formulated as:

$$L(\theta, \psi) = \mathbb{E}_{\epsilon, \mathbf{y}_0, k}\|\epsilon - \epsilon_{(\theta,\psi)}(\mathbf{y}_k, k, \mathbf{x})\|, \tag{9}$$

where ψ denotes we further consider the encoder network in the loss function. Once the network $\epsilon_{(\theta,\psi)}(\mathbf{y}_k, k, \mathbf{x})$ is trained, we can use this network to obtain the mean of Gaussian transition.

$$\boldsymbol{\mu}_\theta(\mathbf{y}_k, k, \mathbf{f}) = \frac{1}{\sqrt{\alpha_k}}(\mathbf{y}_k(\mathbf{y}_0, \epsilon) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_{(\theta,\psi)}(\mathbf{y}_k, k, \mathbf{x})). \tag{10}$$

Table 1. Ablation studies on sampling number on SDD.

Sampling	ADE	FDE
20	7.61	14.30
40	6.84	12.00
20×20	5.42	9.47

Furthermore, the trajectory in next step is predicted as:

$$\mathbf{y}_{k-1} = \frac{1}{\sqrt{\alpha_k}}(\mathbf{y}_k(\mathbf{y}_0, \epsilon) - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_{(\theta,\psi)}(\mathbf{y}_k, k, \mathbf{x})) + \sqrt{\beta_k}\mathbf{z}, \tag{11}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

B. Implementation Details

In this section, we introduce the implementation details of our method, including the hyper-parameters for training, the network architecture, the algorithms of training and inference, and the attached code.

Diffusion Process and Hyper-parameters: We set the lower bound of variance scheduler β_1 to 0.0001 and upper bound β_K to 0.05, and β_k is uniformly sampled between the bounds. For the main Transformer network in diffusion model ϵ_θ, we devise three Transformer Encoder layers where each has the dimension of 512, feedforward dimension of 1024 and 4 attention heads. For the encoder $\mathcal{F}_\psi$, we utilize the default configuration provided by Trajectron++ [2].

Upsample-Downsample Layers: We employ a MLP-based sub-network to upsample the raw trajectory from 2d to 512d, and downsample the output of the Transformer such that 512d-256d-2d as the final output of the network. Each sub-network, denoted by M and parameterized by ϕ, contains three MLP layers which we can formulate as:

$$M_\phi(\mathbf{h}, k, \mathbf{f}) = (\mathbf{W}_1\mathbf{h}+\mathbf{b}_1)\odot\sigma(\mathbf{W}_2\mathbf{c}+\mathbf{b}_2)+(\mathbf{W}_3\mathbf{c}+\mathbf{b}_3). \tag{12}$$

$\mathbf{c}$ is the concatenation of step number embedding and state embedding such that $\mathbf{c} = [k, \sin(k), \cos(k), \mathbf{f}]$ and $\mathbf{h}$ denotes the input trajectory feature of the sub-network. $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ and $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ are the trainable parameters of the MLP layers. σ corresponds to a sigmoid function.

Reproducibility: For better understanding and reproduction, we provide Algorithm 1 and Algorithm 2 showing the training and inference procedure of our MID framework. Furthermore, the code can be found in `https://github.com/gutianpei/MID`.

C. Additional Experiments

We also respectively report the ADE (left) and FDE (right) curves of min_3/min_5 metrics within reverse diffusion steps from 0 to 100 in Figure 1. We can observe

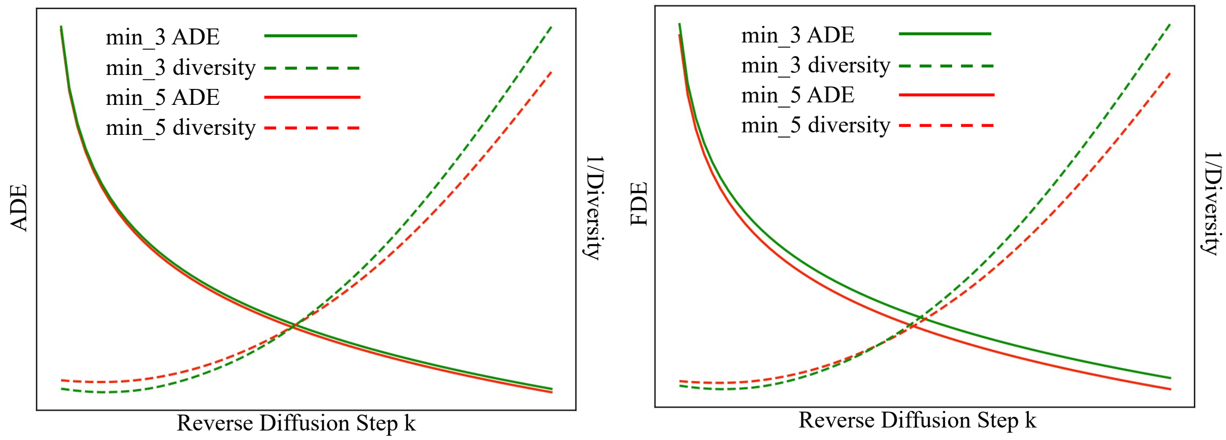

Figure 1. Trade-off between min_3/min_5 and diversity.

reducing the diversity also leads to better predictions with fewer samples, which demonstrates that diversity and determinacy are still contradictory with few samples.

Additionally, we found that the sampling trick is very effective to improve model performance. Sampling tricks usually add the number of sampling and do the post-processing (clustering in YNet [1] and choosing best in Expert [3]). As shown in Table 1, the performance is improved significantly when we add the number of sampling like Expert. However, we don't encourage to use more samplings since more samplings indicate more computation cost.

References

[1] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *ICCV*, pages 15233–15242, 2021. 3

[2] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 2

[3] He Zhao and Richard P Wildes. Where are you heading? dynamic trajectory prediction with expert goal examples. In *ICCV*, pages 7629–7638, 2021. 3