6. Appendix

6.1. Training and Inference Procedure

During the training process, we input the whole trajectories $X_{-T_p:0}, X_{1:T_f}$ and recover them. The procedure is illustrated as follows:

Algorithm 1 Training Procedure

Input: $X_{-T_p:0}, X_{1:T_f}, N$

for $i = 1$ to N **do**
  Construct interactive graphs $G_{-T_p:T_f} = f_g(X_{-T_p:0}, X_{1:T_f})$ in real-world space as Eq 1.
  Generate latent vectors for the whole trajectories $\mu_q, \sigma_q = g_{enc}(X_{-T_p:0}, X_{1:T_f})$ and only previous trajectories $\mu_p, \sigma_p = g_{enc}(X_{-T_p:0})$ as Eq 2.
  Generate initial values h_{-T_p}, latent trajectories and real-wolrd trajectories $\hat{X}_{-T_p:T_f}$ as Eq 3 $\sim$ 6 using μ_q, σ_q and interactive graphs $G_{-T_p:T_f}$.
  Compute ELBO loss based on $\mu_q, \sigma_q, \mu_p, \sigma_p, \hat{X}_{1:T_f}, X_{1:T_f}$. Compute MSE loss based on $\hat{X}_{-T_p:0}, X_{-T_p:0}$. Compute Graph consistency loss based on $G_{-T_p:T_f}$.
  Update the parameters ϕ and θ by optimizing the loss function.
end for
return network parameters ϕ and θ

During the inference process, we only input the previous trajectories $X_{-T_p:0}$ to estimate the future trajectories $X_{1:T_f}$. The procedure is illustrated as follows:

Algorithm 2 Inference Procedure

Input: $X_{-T_p:0}$

Construct interactive graphs $G_{-T_p:0} = f_g(X_{-T_p:0})$ in real-world space as Eq 1.
Generate latent vectors for the previous trajectories $\mu_p, \sigma_p = g_{enc}(X_{-T_p:0})$ as Eq 2.
Generate initial values h_{-T_p}, latent trajectories and real-wolrd trajectories $\hat{X}_{-T_p:T_f}$ as Eq 3 $\sim$ 6 using μ_p, σ_p and interactive graphs $G_{-T_p:0}$.
return estimated future trajectories $\hat{X}_{1:T_f}$

6.2. More Visualization Results

Static Obstacle. In Figure 4, we provide additional visualization results of our Agent Graph ODE approach when faced with sudden static obstacles. These results demonstrate that our proposed method can effectively handle such obstacles.

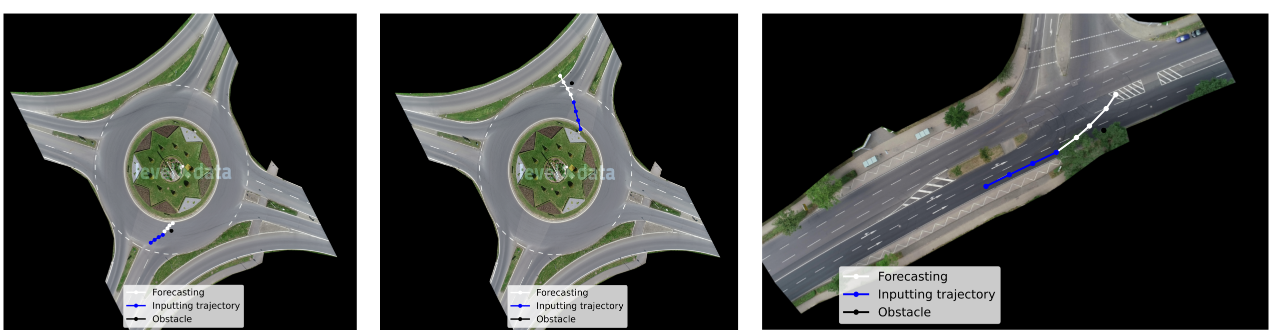

Figure 4. More visualization of Agent Graph ODE in a sudden obstacle scenario. The figure displays input trajectories in blue and predicted trajectories in white, with a black point representing the sudden obstacle. The Agent Graph ODE approach successfully avoids the static obstacle in each scenario.

Moving Obstacle. We also conducted experiments where we placed one obstacle in the original trajectory and made it move against the original trajectory. Videos of these experiments can be found in the "**moving_obstacle**" folder. The videos demonstrate that our approach can successfully avoid moving obstacles, which is more challenging than static obstacles.

6.3. Loss Function

We have the loss function as follows:

$$L = \alpha_1 L_{elbo} + \alpha_2 L_{mse} + \alpha_3 L_g,$$

where L_{elbo} is the ELBO loss in CVAE, L_{mse} is the loss for recovering input previous trajectories and L_g is the loss for consistency for graph construction in real-world space and latent space.

We derive the L_{elbo}. In CVAE, we aim to approximate the conditional probability $p(X_{1:T_f}|X_{-T_p:0})$. We have

$$\begin{aligned}
&\log p(X_{1:T_f}|X_{-T_p:0}) \\
&= E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log p(X_{1:T_f}|X_{-T_p:0})] \\
&= E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log \frac{p(X_{1:T_f}, h_{-T_p}|X_{-T_p:0}) q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}{p(h_{-T_p}|X_{-T_p:0}, X_{1:T_f}) q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}] \\
&= E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log \frac{p(X_{1:T_f}, h_{-T_p}|X_{-T_p:0})}{q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})} + \log \frac{q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}{p(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}] \\
&= E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log \frac{p(X_{1:T_f}, h_{-T_p}|X_{-T_p:0})}{q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}] + D_{KL}(q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})||p(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})) \\
&\geq E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log \frac{p(X_{1:T_f}, h_{-T_p}|X_{-T_p:0})}{q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}] \\
&= E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log \frac{p_\theta(X_{1:T_f}|h_{-T_p}, X_{-T_p:0}) p_\theta(h_{-T_p}|X_{-T_p:0})}{q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}] \\
&= E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})} \log p_\theta(X_{1:T_f}|h_{-T_p}, X_{-T_p:0}) - D_{KL}(q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})||p_\theta(h_{-T_p}|X_{-T_p:0})).
\end{aligned}$$

Therefore, we use negative evidence lower bound (ELBO) L_{elbo} as follows:

$$L_{elbo} = - E_{q_\phi(h_{-T_p}|X_{-T_p:0},X_{1:T_f})}[\log p_\theta(X_{1:T_f}|X_{-T_p:0}, h_{-T_p})] \quad + KL(q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})||p_\theta(h_{-T_p}|X_{-T_p:0})).$$

The first term in the ELBO loss is the reconstruction term, which means that we input the whole trajectories $X_{-T_p:0}, X_{1:T_f}$ to generate initial values h_{-T_p}. Then we generate the future trajectories given previous trajectories and initial values. Because the future trajectories are determined by the initial values and ODEs, we have $\log p_\theta(X_{1:T_f}|X_{-T_p:0}, h_{-T_p}) = \log p_\theta(X_{1:T_f}|h_{-T_p})$. We use Gaussian distribution to model $X_{1:T_f} \sim N(\hat{X}_{1:T_f}, \Sigma)$, where $\hat{X}_{1:T_f}$ is the estimated future trajectories by solving ODEs given initial values h_{-T_p} and Σ is a diagonal matrix. Then we have

$$\begin{aligned}
p(X_{1:T_f}) &= \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}[(X_{1:T_f} - \hat{X}_{1:T_f})^T \Sigma^{-1} (X_{1:T_f} - \hat{X}_{1:T_f})]} \\
\log p_\theta(X_{1:T_f}|X_{-T_p:0}, h_{-T_p}) &= \log p_\theta(X_{1:T_f}|h_{-T_p}) \\
&= \log p_\theta(X_{1:T_f}|\hat{X}_{1:T_f}) \\
&= -\frac{1}{2|\Sigma|}||X_{1:T_f} - \hat{X}_{1:T_f}||_2^2 - \log[(2\pi)^{n/2}|\Sigma|^{1/2}] \\
&\approx k \sum_i ||\hat{x}^i_{1:T_f} - x^i_{1:T_f}||_2^2,
\end{aligned}$$

where k is the coefficient, which we set as the hyperparameter in training. Here, the term $-\log[(2\pi)^{n/2}|\Sigma|^{1/2}]$ is a constant and does not require optimization.

The second term in the ELBO loss is the prior matching term, which makes the posterior probability of h_{-T_p} given the whole trajectories $X_{-T_p:0}, X_{1:T_f}$ approach to the prior probability of h_{-T_p} only given the previous trajectories $X_{-T_p:0}$.

The reason is that the whole trajectories and only the previous trajectories should have similar values. We assume $q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f}) = N(\mu_q, \sigma_q)$ and $p_\theta(h_{-T_p}|X_{-T_p:0}) = N(\mu_p, \sigma_p)$, where μ_q, σ_q are the output of the encoder given the whole trajectories and μ_p, σ_p are the output given only previous trajectories. The KL divergence between two Gaussian distributions is

$$
\begin{aligned}
&KL(q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})||p_\theta(h_{-T_p}|X_{-T_p:0})) \\
&= KL(N(\mu_q, \sigma_q)||N(\mu_p, \sigma_p)) \\
&= -\frac{1}{2}\sum_{j=1}^{J}[\log\frac{\sigma_{q,j}^2}{\sigma_{p,j}^2} - \frac{\sigma_{q,j}^2}{\sigma_{p,j}^2} - \frac{(\mu_{q,j}-\mu_{p,j})^2}{\sigma_{p,j}^2} + 1],
\end{aligned}
$$

where J is the dimension of the latent initial value h_{-T_p} and $\mu_{q,j}$ is the value in jth dimension.

Therefore, the ELBO loss in CVAE should be

$$
L_{elbo} = -E_{q_\phi(h_{-T_p}|X_{-T_p:0}, X_{1:T_f})}[k\sum_i ||\hat{x}_{1:T_f}^i - x_{1:T_f}^i||_2^2] - \frac{1}{2}\sum_{j=1}^{J}[\log\frac{\sigma_{q,j}^2}{\sigma_{p,j}^2} - \frac{\sigma_{q,j}^2}{\sigma_{p,j}^2} - \frac{(\mu_{q,j}-\mu_{p,j})^2}{\sigma_{p,j}^2} + 1]
$$