A SUPPLEMENTARY TO WEIGHTING FUNCTION

We give intuition for our choice of weighting function (main text, Eq. (6)). Since we approximate the integrals w.r.t. $q(\mathbf{s}_{t-1} \mid \mathbf{x}_{\leq t})$ (main text, Eqs. (4) and (7)) with samples from $\tilde{q}(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})$ [1] instead of samples from $q(\mathbf{s}_{t-1} \mid \mathbf{x}_{\leq t})$, importance sampling tells us that the weigths should be

$$\omega(\mathbf{s}_{t-1}, \mathbf{x}_t) = \frac{q(\mathbf{s}_{t-1} \mid \mathbf{x}_{\leq t})}{\tilde{q}(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})} = \frac{q(\mathbf{x}_t \mid \mathbf{s}_{t-1})}{q(\mathbf{x}_t \mid \mathbf{x}_{<t})} \frac{\tilde{q}(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})}{\tilde{q}(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})} = \frac{q(\mathbf{x}_t \mid \mathbf{s}_{t-1})}{q(\mathbf{x}_t \mid \mathbf{x}_{<t})} \propto q(\mathbf{x}_t \mid \mathbf{s}_{t-1}) \quad (1)$$

This is consistent with out earlier definition of $q(\mathbf{s}_{t-1} \mid \mathbf{x}_{\leq t}) = \omega(\mathbf{s}_{t-1}, \mathbf{x}_t)\tilde{q}(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})$. The weights are proportional to the likelihood of the variational model $q(\mathbf{x}_t \mid \mathbf{s}_{t-1})$. We choose to parametrize it using the likelihood of the generative model $p(\mathbf{x}_t \mid \mathbf{s}_{t-1})$ and get

$$\omega_t^{(i)} = \omega(\mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t)/k := \mathbb{1}(i = \arg\max_j p(\mathbf{x}_t \mid \mathbf{s}_{t-1}^{(j)})). \quad (2)$$

With this choice of the weighting function, only the mixture component with the highest likelihood is selected to be in charge of modeling the current observation $\mathbf{x}_t$. As a result, other mixture components have the capacity to focus on different modes. This helps avoid the effect of mode-averaging. An alternative weight function is given in Appendix G.

B SUPPLEMENTARY TO LOWER BOUND

Claim. *The ELBO (main text, Eq. (8)) is a lower bound on the log evidence* $\log p(\mathbf{x}_t \mid \mathbf{x}_{<t})$*,*

$$\log p(\mathbf{x}_t \mid \mathbf{x}_{<t}) \geq \mathcal{L}_{\text{ELBO}}(\mathbf{x}_{\leq t}, \phi)\,. \quad (3)$$

Proof. We write the data evidence as the double integral over the latent variables $\mathbf{z}_t$, and $\mathbf{z}_{<t}$.

$$\log p(\mathbf{x}_t \mid \mathbf{x}_{<t}) = \log \iint p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{<t}) p(\mathbf{z}_{<t} \mid \mathbf{x}_{<t}) \mathrm{d}\mathbf{z}_t \mathrm{d}\mathbf{z}_{<t} \quad (4)$$

We multiply the posterior at the previous time step $p(\mathbf{z}_{<t} \mid \mathbf{x}_{<t})$ with the ratio $\frac{f(\mathbf{a},\mathbf{b})}{f(\mathbf{a},\mathbf{b})}$, where f is any suitable function of two variables $\mathbf{a}$ and $\mathbf{b}$. The following equality holds, since the ratio equals to one.

$$\log p(\mathbf{x}_t \mid \mathbf{x}_{<t}) = \log \int \frac{f(\mathbf{a},\mathbf{b})}{f(\mathbf{a},\mathbf{b})} p(\mathbf{z}_{<t} \mid \mathbf{x}_{<t}) \int p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{<t}) \mathrm{d}\mathbf{z}_t \mathrm{d}\mathbf{z}_{<t} \quad (5)$$

We move the integral over $\mathbf{z}_{<t}$ out of the log operation and apply the Jensen's inequality.

$$\begin{aligned} \log p(\mathbf{x}_t \mid \mathbf{x}_{<t}) \geq\ & \mathbb{E}_{f(\mathbf{a},\mathbf{b})p(\mathbf{z}_{<t}\mid\mathbf{x}_{<t})} \left[\log \int p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{<t}) \mathrm{d}\mathbf{z}_t \right] \quad (6) \\ & - \mathbb{E}_{f(\mathbf{a},\mathbf{b})p(\mathbf{z}_{<t}\mid\mathbf{x}_{<t})} \left[\log f(\mathbf{a},\mathbf{b}) \right] \end{aligned}$$

We introduce the variational posterior $q(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{\leq t})$, and apply Jensen's inequality to replace the intractable integral $\log \int p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{<t}) \mathrm{d}\mathbf{z}_t$ with its lower bound.

$$\begin{aligned} \log p(\mathbf{x}_t \mid \mathbf{x}_{<t}) \geq\ & \mathbb{E}_{f(\mathbf{a},\mathbf{b})p(\mathbf{z}_{<t}\mid\mathbf{x}_{<t})} \left[\mathbb{E}_{q(\mathbf{z}_t\mid\mathbf{z}_{<t},\mathbf{x}_{\leq t})} \left[\log \frac{p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{<t})}{q(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_{\leq t})} \right] \right] \\ & - \mathbb{E}_{f(\mathbf{a},\mathbf{b})p(\mathbf{z}_{<t}\mid\mathbf{x}_{<t})} \left[\log f(\mathbf{a},\mathbf{b}) \right]\ . \quad (7) \end{aligned}$$

We plug in our generative and inference model described in the main text and use the recurrent state $\mathbf{s}_{t-1}$ to summarize the previous latent variables $\mathbf{z}_{<t}$. The previous posterior $p(\mathbf{z}_{<t} \mid \mathbf{x}_{<t})$ is approximated by $q(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})$. We introduce the weighting function $\omega(\mathbf{s}_{t-1}, \mathbf{x}_t)$ as the function $f(\mathbf{a},\mathbf{b})$. The expectation with respect to $f(\mathbf{a},\mathbf{b})p(\mathbf{z}_{<t} \mid \mathbf{x}_{<t})$ can be approximated with samples coming from $q(\mathbf{s}_{t-1} \mid \mathbf{x}_{<t})$ and then being weighted by $\omega(\mathbf{s}_{t-1}, \mathbf{x}_t)$.

$$\begin{aligned} \log p(\mathbf{x}_t \mid \mathbf{x}_{<t}) \geq\ & \frac{1}{k} \sum_i^k \omega(\mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t) \mathbb{E}_{q(\mathbf{z}_t\mid\mathbf{s}_{t-1}^{(i)},\mathbf{x}_t)} \left[\log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{s}_{t-1}^{(i)}) + \log \frac{p(\mathbf{z}_t \mid \mathbf{s}_{t-1}^{(i)})}{q(\mathbf{z}_t \mid \mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t)} \right] \\ & - \frac{1}{k} \sum_i^k \omega(\mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t) \log \omega(\mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t) \quad (8) \end{aligned}$$

□

[1] The $\sim$ just helps to visually distinguish the two distributions that appear in the main text.

C Supplementary to Stochastic Cubature Approximation

Cubature approximation. The cubature approximation is widely used in the engineering community as a deterministic method to numerically integrate a nonlinear function $f(\cdot)$ of Gaussian random variable $z \sim \mathcal{N}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2 \mathbb{I})$, with $\mathbf{z} \in \mathbb{R}^d$. The method proceeds by constructing $2d+1$ sigma points $\mathbf{z}^{(i)} = \mu_{\mathbf{z}} + \sigma_{\mathbf{z}} \xi^{(i)}$. The cubature approximation is simply a weighted sum of the sigma points propagated through the nonlinear function $f(\cdot)$,

$$\int f(\mathbf{z}) \mathcal{N}(\mathbf{z} \mid \mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2 \mathbb{I}) \mathrm{d}\mathbf{z} \approx \sum_{i=1}^{2d+1} \gamma^{(i)} f(\mathbf{z}^{(i)}) \,. \tag{9}$$

Simple analytic formulas determine the computation of weights $\gamma^{(i)}$ and the locations of $\xi^{(i)}$.

$$\gamma^{(i)} = \begin{cases} \frac{1}{2(n+\kappa)} & , i = 1, ..., 2n \\ \frac{\kappa}{n+\kappa} & , i = 0 \end{cases} \qquad \xi^{(i)} = \begin{cases} \sqrt{n+\kappa}\mathbf{e}_i & , i = 1, ..., n \\ -\sqrt{n+\kappa}\mathbf{e}_{i-n} & , i = n+1, ..., 2n \\ 0 & , i = 0 \,, \end{cases} \tag{10}$$

where κ is a hyperparameter controlling the spread of the sigma points in the n-dimensional sphere. Further $\mathbf{e}_i$ represents a basis in the n-dimensional space, which is choosen to be a unit vector in cartesian space, e.g. $\mathbf{e}_1 = [1, 0, ..., 0]$.

Stochastic cubature approximation. In stochastic cubature approximation (SCA), we adopt the computation of $\xi^{(i)}$ in Eq. (10), and infuse the sigma points with standard Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbb{I})$ to obtain stochastic *sigma variables* $\mathbf{s}^{(i)} = \mu_{\mathbf{z}} + \sigma_{\mathbf{z}}(\xi^{(i)} + \epsilon)$. We choose $\kappa = 0.5$ to set the weights $\gamma^{(i)}$ equally.

D Supplementary to Ablation Study of Regularization Terms

We investigate the effect of the regularization terms using the synthetic data (main text, Fig. 3). We can see in Table 1, VDM($k = 9$) can be trained successfully with $\mathcal{L}_{\text{ELBO}}$ only, and both regularization terms improve the performance (negative log-likelihood of multi-steps ahead prediction), while VDM($k = 1$) doesn't work whatever the regularization terms. Additionally, we tried to train the model only with the regularization terms (each separate or together) but these options diverged during training.

Table 1: Ablation study of the regularization terms for synthetic data (main text, Fig. 3)

	$\mathcal{L}_{\text{ELBO}}$	$\mathcal{L}_{\text{ELBO}}$&$\mathcal{L}_{pred}$	$\mathcal{L}_{\text{ELBO}}$&$\mathcal{L}_{adv}$	$\mathcal{L}_{\text{VDM}}$
VDM($k = 9$)	2.439±0.005	2.379±0.008	2.381±0.006	**2.363**±0.004
VDM($k = 1$)	3.756±0.003	3.960±0.008	3.743±0.005	3.878±0.007

E Supplementary to Experiments Setup

E.1 Stochastic lorenz attractor setup

Lorenz attractor is a system of three ordinary differential equations:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \sigma(\mathbf{y} - \mathbf{x}), \quad \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{x}(\rho - \mathbf{z}) - \mathbf{y}, \quad \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{x}\mathbf{y} - \beta\mathbf{z} \,, \tag{11}$$

where σ, ρ, and β are system parameters. We set $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$ to make the system chaotic. We simulate the trajectories by RK4 with a step size of 0.01. To make it stochastic, we add process noise to the transition, which is a mixture of two Gaussians $0.5\mathcal{N}(\mathbf{m}_0, \mathbf{P}) + 0.5\mathcal{N}(\mathbf{m}_2, \mathbf{P})$, where

$$\mathbf{m}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{m}_1 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0.06 & 0.03 & 0.01 \\ 0.03 & 0.03 & 0.03 \\ 0.01 & 0.03 & 0.05 \end{bmatrix}. \tag{12}$$

Besides, we add a Gaussian noise with zero mean and diagonal standard deviation $[0.6, 0.4, 0.8]$ as the observation noise. Totally, we simulate 5000 sequences as training set, 200 sequences as validation set, and 800 sequences as test set. For evaluation of Wasserstein distance, we simulate 10 groups of sequences additionally. Each group has 100 sequences with similar initial observations.

E.2 Taxi trajectories setup

The full dataset is very large and the length of trajectories varies. We select the trajectories inside the Porto city area with length in the range of 30 and 45, and only extract the first 30 coordinates of each trajectory. Thus we obtain a dataset with a fixed sequence length of 30. We split it into the training set of size 86386, the validation set of size 200, and the test set of size 10000.

E.3 U.S. pollution data setup

The U.S. pollution dataset consists of four pollutants (NO2, O3, SO2 and O3). Each of them has 3 major values (mean, max value, and air quality index). It is collected from counties in different states for every day from 2000 to 2016. Since the daily measurements are too noisy, we firstly compute the monthly average values of each measurement, and then extract non-overlapped segments with the length of 24 from the dataset. Totally we extract 1639 sequences as training set, 25 sequences as validation set, and 300 sequences as test set.

F Implementation Details

Here, we provide implementation details of variational dynamic mixtures (VDM) models used across the three datasets in the main paper. VDM consists of

- encoder: embed the first observation $\mathbf{x}_0$ to the latent space as the initial latent state $\mathbf{z}_0$.
- transition network: propagate the latent states $\mathbf{z}_t$.
- decoder: map the latent states $\mathbf{z}_t$ and the recurrent states $\mathbf{h}_t$ to observations $\mathbf{x}_t$.
- inference network: update the latent states $\mathbf{z}_t$ given observations $\mathbf{x}_t$.
- latent GRU: summarize the historic latent states $\mathbf{z}_{\leq t}$ in the recurrent states $\mathbf{h}_t$.
- discriminator: be used for adversarial training.

The optimizer is Adam with the learning rate of $1e-3$. In all experiments, the networks have the same architectures but different sizes. The model size depends on observation dimension $\mathbf{d}_\mathbf{x}$, latent state dimension $\mathbf{d}_\mathbf{z}$, and recurrent state dimension $\mathbf{d}_\mathbf{h}$. The number of samples used at each time step in the training is $2\mathbf{d}_\mathbf{z} + 1$. If the model output is variance, we use the exponential of it to ensure its non-negative.

- Encoder: input size is $\mathbf{d}_\mathbf{x}$; 3 linear layers of size 32, 32 and $2\mathbf{d}_\mathbf{z}$, with 2 ReLUs.
- Transition network: input size is $\mathbf{d}_\mathbf{h}$; 3 linear layers of size 64, 64, and $2\mathbf{d}_\mathbf{z}$, with 3 ReLUs.
- Decoder: input size is $\mathbf{d}_\mathbf{h} + \mathbf{d}_\mathbf{z}$; 3 linear layers of size 32, 32 and $2\mathbf{d}_\mathbf{x}$, with 2 ReLUs.
- Inference network: input size is $\mathbf{d}_\mathbf{h} + \mathbf{d}_\mathbf{x}$; 3 linear layers of size 64, 64, and $2\mathbf{d}_\mathbf{z}$, with 3 ReLUs.
- Latent GRU: one layer GRU of input size $\mathbf{d}_\mathbf{z}$ and hidden size $\mathbf{d}_\mathbf{h}$
- Discriminator: one layer GRU of input size $\mathbf{d}_\mathbf{x}$ and hidden size $\mathbf{d}_\mathbf{h}$ to summarize the previous observations as the condition, and a stack of 3 linear layers of size 32, 32 and 1, with 2 ReLUs and one sigmoid as the output activation, whose input size is $\mathbf{d}_\mathbf{h} + \mathbf{d}_\mathbf{x}$.

Stochastic Lorenz attractor. Observation dimension $\mathbf{d_x}$ is 3, latent state dimension $\mathbf{d_z}$ is 6, and recurrent state dimension $\mathbf{d_h}$ is 32.

Taxi trajectories. Observation dimension $\mathbf{d_x}$ is 2, latent state dimension $\mathbf{d_z}$ is 6, and recurrent state dimension $\mathbf{d_h}$ is 32.

U.S. pollution data[2] Observation dimension $\mathbf{d_x}$ is 12, latent state dimension $\mathbf{d_z}$ is 8, and recurrent state dimension $\mathbf{d_h}$ is 48.

Here, we give the number of parameters for each model in different experiments in Table 2.

Table 2: Number of parameters for each model in three experiments. VDM, auto-encoding sequential Monte Carlo (AESMC), variational recurrent neural network (VRNN), and recurrent Kalman network (RKN) have comparable number of parameters. conditional flow variational autoencoder (CF-VAE) has much more parameters.

	RKN	VRNN	CF-VAE	AESMC	VDM
Lorenz	23170	22506	7497468	22218	22218
Taxi	23118	22248	7491123	22056	22056
Pollution	35774	33192	8162850	31464	31464

G Additional Evaluation Results

We evaluate more variants of VDM in the chosen experiments to investigate the different choices of sampling methods (Monte Carlo method, and SCA) and weighting functions (Eqs. (13) and (14)). In addition to Eq. (13) described in the main text, we define one other choice in Eq. (14).

$$\omega_t^{(i)} = \omega(\mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t)/k := \mathbb{1}(i = \arg\max_j p(\mathbf{x}_t \mid \mathbf{s}_{t-1}^{(j)})) \tag{13}$$

$$\omega_t^{(i)} = \omega(\mathbf{s}_{t-1}^{(i)}, \mathbf{x}_t)/k := \mathbb{1}(i = j \sim \text{Cat}(\cdot \mid \omega^1, \ldots, \omega^k)), \quad \omega^j \propto p(\mathbf{x}_t \mid \mathbf{s}_{t-1}^{(j)}), \tag{14}$$

We define the weighting function as an indicator function, in Eq. (13) we set the non-zero component by selecting the sample that achieves the highest likelihood, and in Eq. (14) the non-zero index is sampled from a categorical distribution with probabilities proportional to the likelihood. The first choice (Eq. (13)) is named with δ-function, and the second choice (Eq. (14)) is named with categorical distribution. Besides, in VDM-Net, we evaluate the performance of replacing the closed-form inference of the weighting function with an additional inference network. In Table 3, we show the choices in different variants. All models are trained with $\mathcal{L}_{\text{ELBO}}\&\mathcal{L}_{pred}$.

Table 3: Definition of VDM variants

	VDM($k=1$)	VDM-MC+δ	VDM-SCA+Cat	VDM-SCA+δ
Sampling method	Monte-Carlo	Monte-Carlo	SCA	SCA
Weighting function	n.a.	δ-function	Categorical distribution	δ-function

G.1 Stochastic Lorenz attractor

G.2 Taxi trajectories

G.3 U.S. pollution data

[2] https://www.kaggle.com/sogun3/uspollution

Table 4: Ablation study of VDM's variants on stochastic Lorenz attractor for three distance metrics (see main text). The variants are defined in Table 3. All variants give comparable quantitative results.

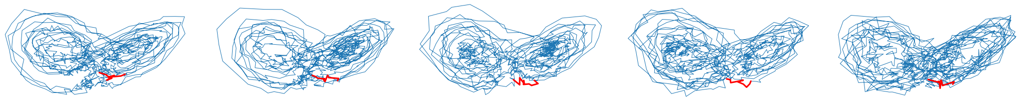
	VDM($k=1$)	VDM-Net	VDM-MC+δ	VDM-SCA+Cat	VDM-SCA+δ
Multi-steps	25.03±0.28	26.65±0.15	24.67±0.16	24.69±0.16	24.49±0.16
One-step	-1.81	-1.71	-1.84	-1.83	-1.81
W-distance	7.31±0.002	7.68±0.002	7.31±0.005	7.30±0.009	7.29±0.003

Table 5: Ablation study of VDM's variants on taxi trajectories for three distance metrics (see main text). The variants are defined in Table 3. VDM-SCA+δ outperforms other variants and approaches our default VDM (trained with $\mathcal{L}_{adv}$ additionally).

	VDM($k=1$)	VDM-Net	VDM-MC+δ	VDM-SCA+Cat	VDM-SCA+δ
Multi-steps	3.26±0.001	3.68±0.002	3.17±0.001	3.09±0.001	2.88±0.002
One-step	-2.99	-2.74	-3.21	-3.24	-3.68
W-distance	0.69±0.0005	0.79±0.0003	0.70±0.0008	0.64±0.0005	0.59±0.0008

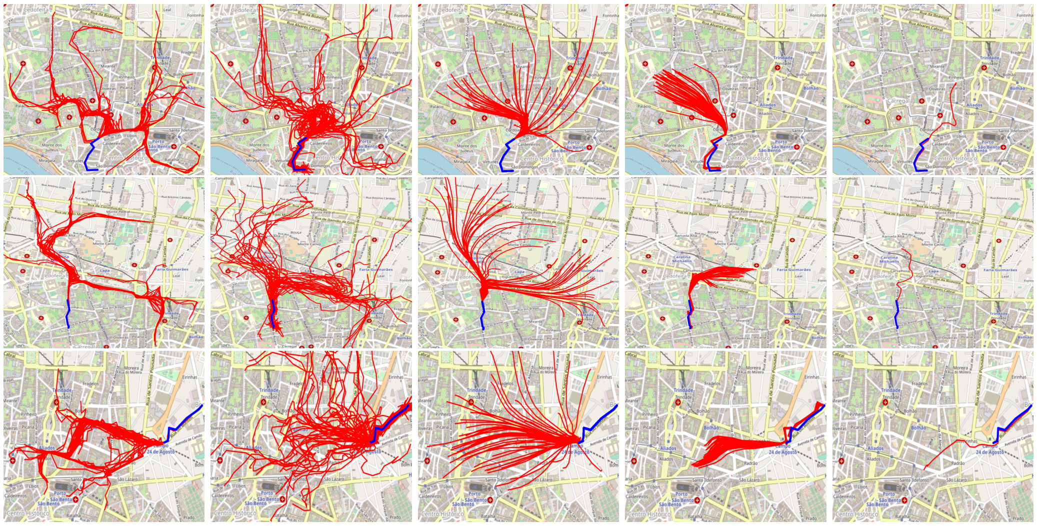
Table 6: Ablation study of VDM's variants on U.S. pollution data for two distance metrics (see main text). The variants are defined in Table 3. VDM-SCA+δ outperforms other variants.

	VDM($k=1$)	VDM-Net	VDM-MC+δ	VDM-SCA+Cat	VDM-SCA+δ
Multi-steps	42.33±0.11	52.44±0.04	40.33±0.03	39.58±0.09	37.64±0.07
One-step	7.97	10.70	8.12	7.82	6.91



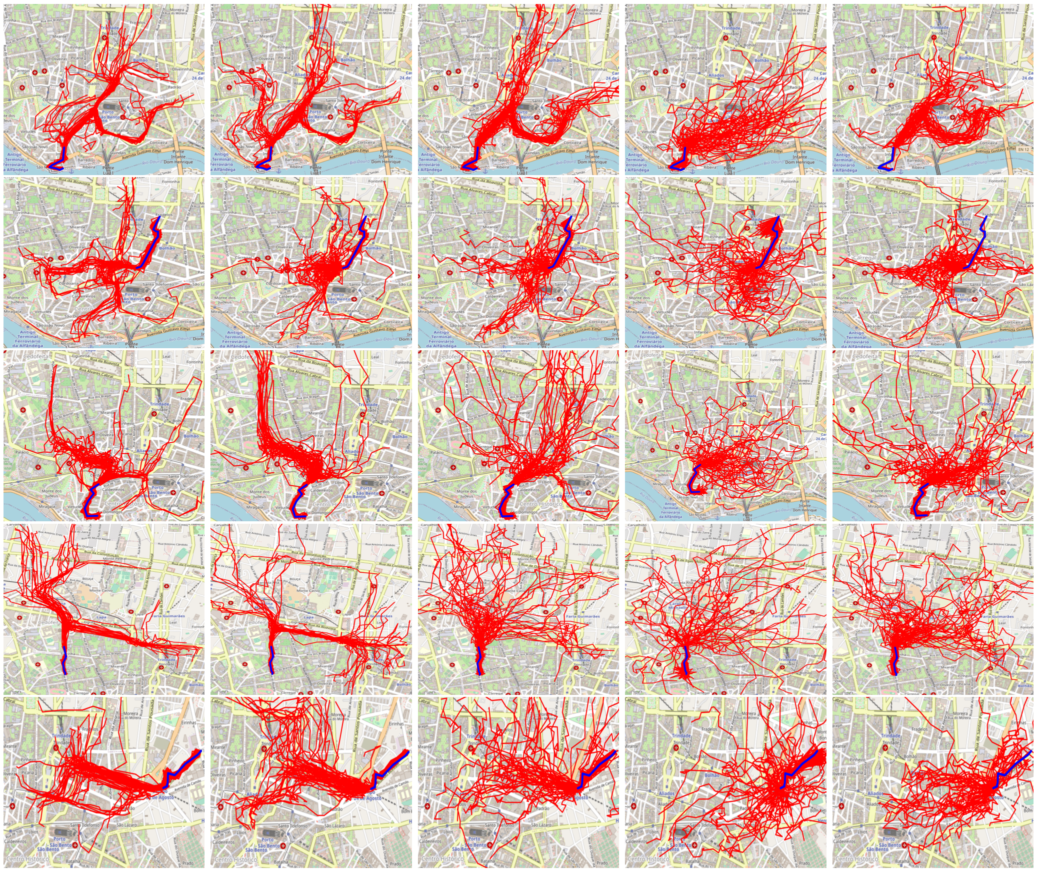
(a) VDM-SCA+δ (b) VDM-SCA+Cat (c) VDM-MC+δ (d) VDM-Net (e) VDM($k=1$)

Figure 1: Generated trajectories of stochastic Lorenz attractor from VDM variants. The first ten observations (red) are obtained from models given the first 10 true observations. The rest 990 observations (blue) are predicted. We can see, all variants give very good qualitative results. Since the fundamental dynamics is govern by ordinary differential equations, the transition at each time step is not highly multi-modal. Once the model is equipped with a stochastic transition, it is able to model this dynamics.



(a) VDM (ours) (b) AESMC (c) CF-VAE (d) VRNN (e) RKN

Figure 2: Generated 50 taxi trajectories in 3 different areas from VDM and the baselines. All models are required to predict the future continuations (red), based the beginning of a trajectory (blue). VDM generates more plausible trajectories compared with the baselines. While the generated trajectories from VDM follow the street map, the generated trajectories from all baselines are physically impossible. AESMC and CF-VAE can capture the general evolving direction, but suffer from capturing the multi-modality at each time step.



(a) VDM-SCA+δ (b) VDM-SCA+Cat (c) VDM-MC+δ (d) VDM-Net (e) VDM$(k = 1)$

Figure 3: Generated 50 taxi trajectories from VDM variants. All models are required to predict the future continuations (red), based the beginning of a trajectory (blue). VDM-SCA+δ achieves the best qualitative results among all variants. VDM-SCA+δ can generate plausible trajectories, even it is trained without the adversarial term $\mathcal{L}_{adv}$. We can see, for the weighting function, Eq. (13) is better than Eq. (14), and for the sampling method, SCA is better than Monte-Carlo method.