entropy

MDPI

Article

History Marginalization Improves Forecasting in Variational Recurrent Neural Networks

Chen Qiu [1,2], Stephan Mandt [3] and Maja Rudolph [4,*]

[1] Bosch Center for AI, 71272 Renningen, Germany; Chen.Qiu@de.bosch.com
[2] Department of Computer Science, TU Kaiserslautern, 67653 Kaiserslautern, Germany
[3] Department of Computer Science, University of California, Irvine, CA 92697, USA; Mandt@uci.edu
[4] Bosch Center for AI, Pittsburgh, PA 15222, USA
* Correspondence: Maja.Rudolph@us.bosch.com

Citation: Qiu, C.; Mandt, S.; Rudolph, M. History Marginalization Improves Forecasting in Variational Recurrent Neural Networks. *Entropy* **2021**, *23*, 1563. https://doi.org/10.3390/e23121563

Academic Editors: Eric Nalisnick and Dustin Tran

Received: 30 September 2021
Accepted: 19 November 2021
Published: 24 November 2021

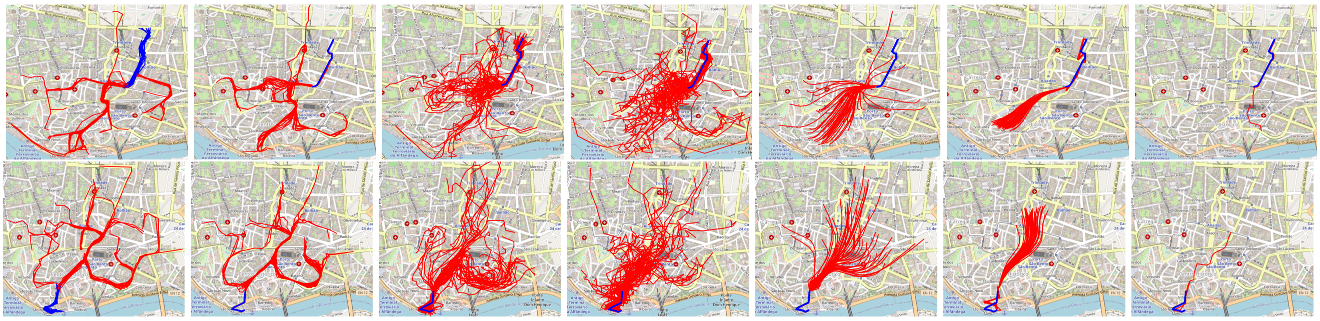
Abstract: Deep probabilistic time series forecasting models have become an integral part of machine learning. While several powerful generative models have been proposed, we provide evidence that their associated inference models are oftentimes too limited and cause the generative model to predict mode-averaged dynamics. Mode-averaging is problematic since many real-world sequences are highly multi-modal, and their averaged dynamics are unphysical (e.g., predicted taxi trajectories might run through buildings on the street map). To better capture multi-modality, we develop variational dynamic mixtures (VDM): a new variational family to infer sequential latent variables. The VDM approximate posterior at each time step is a mixture density network, whose parameters come from propagating multiple samples through a recurrent architecture. This results in an expressive multi-modal posterior approximation. In an empirical study, we show that VDM outperforms competing approaches on highly multi-modal datasets from different domains.

Keywords: sequential latent variable models; time series forecasting; variational inference

1. Introduction

Making sense of time series data can be challenging, especially in real world data-sets that are highly multi-modal. There may be multiple plausible future projections at any given part of the observed sequence, but the average projection is often highly unlikely or even physically impossible. As an example, consider a dataset of taxi trajectories (https://www.kaggle.com/crailtap/taxi-trajectory, accessed on 1 March 2020). In each row of Figure 1a, we have selected 50 routes from the dataset with similar starting behavior (blue). Even though these routes are quite similar to each other in the first ten waypoints, the continuations of the trajectories (red) can exhibit distinct behaviors and lead to points on any far edge of the map. We see that trajectories follow a few main traffic arteries, which are the data distribution's modes. Our goal is to learn a generative model of the data that can forecast plausible continuations for the trajectories based on some initial waypoints.

Most data-driven neural forecasting models are based on assumptions such as Gaussianity to make learning tractable and efficient. However, trying to capture the dynamics through unimodal distributions can lead either to "over-generalization" (i.e., placing probability mass in spurious regions) or focusing only on the dominant mode. Even expressive neural approaches based on deep sequential latent variable models fail to capture this multi-modality fully. In this paper, we stress that the shortcomings of these models can be traced back to restrictive modeling assumptions in their approximate *inference*. To address this, we develop variational dynamic mixtures (VDM): a new inference approach for deep sequential latent variable models. Our main contributions are as follows:



(**a**) Taxi Data (**b**) VDM (**c**) AESMC (**d**) DMM-IAF (**e**) CF-VAE (**f**) VRNN (**g**) RKN

Figure 1. Forecasting taxi trajectories is challenging due to the highly multi-modal nature of the data (**a**). VDM (**b**) succeeds in generating diverse plausible predictions (red), based the beginning of a trajectory (blue). The other methods, auto-encoding sequential Monte Carlo (AESMC) [1], deep Markov model [2] with variational posteriors based on inverse autoregressive flows [3] (DMM-IAF), conditional flow variational autoencoder (CF-VAE) [4], variational recurrent neural network (VRNN) [5], recurrent Kalman network (RKN) [6], suffer from mode averaging.

- A new inference model. We establish a new type of variational family for inference in sequential latent variable models. Instead of a structured variational approximation, VDM marginalizes over past states. This leads to an efficient mean-field factorization where each variational factor is multi-modal by construction.
- An evaluation metric for multi-modal forecasting. The negative log-likelihood measures predictive accuracy but neglects an important aspect of multi-modal forecasts—sample diversity. In Section 4, we propose a score inspired by the Wasserstein distance [7] which evaluates both prediction quality and diversity. This metric complements our evaluation based on log-likelihoods.
- An extensive empirical study. In Section 4, we use VDM to study various datasets, including synthetic data, a stochastic Lorenz attractor, taxi trajectories, basketball player trajectories, and a U.S. pollution dataset with the measurements of various pollutants over time. We illustrate VDM's ability in modeling multi-modal dynamics and provide quantitative comparisons to other methods showing that VDM compares favorably to previous work.

2. Related Work

Recurrent neural networks (RNNs) such as long short-term memorys (LSTMs) [8] and gated recurrent units (GRUs) [9] have proven successful on many time series modeling tasks. However, as deterministic models they cannot capture uncertainties in their dynamic predictions. Stochastic RNNs make these sequence models non-deterministic [5,10–12]. For example, the variational recurrent neural network (VRNN) [5] enables multiple stochastic forecasts due to its stochastic transition dynamics. An extension of VRNN [13] uses an auxiliary cost to alleviate the KL-vanishing problem. It improves on VRNN inference by forcing the latent variables to also be predictive of future observations. Another line of related methods rely on particle filtering [1,14,15] and in particular sequential Monte Carlo (SMC) to improve the evidence lower bound. In contrast, VDM adopts an explicitly multi-modal posterior approximation. Another SMC-based work [16] employs search-based techniques for multi-modality but is limited to models with finite discrete states. Recent works [17–19] use normalizing flows in the latent space to model the transition dynamics. A task orthogonal to multi-modal inference is learning disentangled representations. Here too, mixture models are used [20,21]. These papers use discrete variables and a mutual information based term to disentangle different aspects of the data. VAE-like models [4,22] and GAN-like models [23,24] only have global, time independent latent variables. Yet, they show good results on various tasks, including forecasting. With a deterministic decoder, these models focus on average dynamics and do not capture local details (including multi-modal transitions) very well.

Classical state-space models (SSMs) are popular due to their tractable inference and interpretable predictions. Similarly, *deep* SSMs with locally linear transition dynamics enjoy tractable inference [6,25–27]. However, these models are often not expressive enough to capture complex (or highly multi-modal) dynamics. Nonlinear deep SSMs [2,28–31] are more flexible. Their inference is often no longer tractable and requires variational approximations. Unfortunately, in order for the inference to be tractable, the variational approximations are often simplistic and do not approximate multi-modal posteriors well with negative effects on the trained models. Multi-modality can be incorporated via normalizing flows [3] or via additional discrete switching latent variables, such as switching linear dynamical systems [32–34].

3. Method–Variational Dynamic Mixtures

Variational methods for sequential latent variable models often use a structured posterior approximation [1,2,5,25–28], where the variational factors condition on past states. These factors are usually considered to be conditional Gaussians. The Gaussian assumption significantly limits the generative model's dynamics and often leads to mode-averaging behavior. With VDM we develop a variational method for deep sequential latent variable models that overcomes these shortcomings.

Unlike recent work on dynamics modeling, VDM relies on a mean-field assumption. Marginalization over past states mediates temporal dependencies. It has three effects. (1) The factorization of the posterior approximation is mean-field, leading to efficient evidence lower bound (ELBO) computations. (2) The marginalization introduces information about previously inferred dynamics into the variational factors. (3) Each variational factor is a mixture of Gaussians, resulting in an advantageous inference procedure for learning multi-modal dynamics.

We first present the generative model (Section 3.1) and the multi-modal inference model (Section 3.2) of VDM. In Section 3.3, we then present the variational objective including an optional regularization term that gives a nice performance boost. At last, we discuss alternative implementation choices that are optional but can enhance the expressiveness of the model in Section 3.4.

3.1. The Generative Model of VDM

Given sequential observations $\mathbf{x}_{1:T} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, we assume that the underlying dynamics are governed by the latent states $\mathbf{z}_{1:T} = (\mathbf{z}_1, \ldots, \mathbf{z}_T)$. Although the approach is more general, we consider a basic deep latent sequence modeling architecture inspired by [5]. The generative modelconsists of a transition and an emission model. The transition model $p(\mathbf{z}_t \mid \mathbf{z}_{<t})$ describes the temporal evolution of the latent states whose dynamics are governed by a recurrent neural network, such as a GRU [35], ϕ^{GRU} with the hidden state $\mathbf{h}_t$ [9,28,36] (for a better long term generation, we do not incorporate autoregressive feedback from the data $\mathbf{x}_t$). The emission model $p(\mathbf{x}_t \mid \mathbf{z}_{\leq t})$ maps the states to observations. We assume they are parameterized by two separate neural networks, the transition network ϕ^{tra} and the emission network ϕ^{dec}. With $\mathbf{h}_1$ initialized to a vector of zeros, the latent states $\mathbf{z}_t$ are sampled recursively as

$$p(\mathbf{z}_t \mid \mathbf{z}_{<t}) = \mathcal{N}(\mu_{0,t}, \sigma^2_{0,t}\mathbb{I}), \quad \text{where} \quad [\mu_{0,t}, \sigma^2_{0,t}] = \phi^{tra}(\mathbf{h}_t), \, \mathbf{h}_t = \phi^{\mathrm{GRU}}(\mathbf{z}_{t-1}, \mathbf{h}_{t-1}). \tag{1}$$

Conditioned on $\mathbf{z}_t$ and $\mathbf{h}_t$, the data are then generated according to the emission model

$$p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}) = \mathcal{N}(\mu_{x,t}, \sigma^2_{x,t}\mathbb{I}), \qquad \text{where} \quad [\mu_{x,t}, \sigma^2_{x,t}] = \phi^{dec}(\mathbf{z}_t, \mathbf{h}_t). \tag{2}$$

Similar generative models have been studied before. The main innovation of VDM is its inference procedure.

3.2. *The Variational Posterior of VDM*

While the VRNN [5] and other variational approaches for neural recurrent models use a structured posterior, we make the mean-field assumption that the variational family factors over time. Even though our generative model is similar to the VRNN, the competitive edge of VDM comes from marginalizing over past states in the inference. Like including an auxiliary variable in the variational factors [37], this makes the posterior approximation more flexible and relates to placing a prior on the variational parameters of the mean-field factors [38,39]. In VDM the past states $\mathbf{z}_{<t}$ are treated as auxiliary variables for the marginal posterior at time t. This allows the method to pass information about previously inferred dynamics into the variational factors.

$$q(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t \mid \mathbf{x}_{\leq t}) = \prod_{t=1}^{T} \int q_{\text{aug}}(\mathbf{z}_t, \mathbf{z}_{<t} \mid \mathbf{x}_{\leq t}) d\mathbf{z}_{<t}\,. \tag{3}$$

While this variational approximation has the added expressiveness of marginalizing out past states, it is mean-field, which leads to advantages when deriving the variational objective. We assume the augmented distribution factorizes into an inference distribution q_{inf} and a target distribution q_{tar},

$$q_{\text{aug}}(\mathbf{z}_t, \mathbf{z}_{<t} \mid \mathbf{x}_{\leq t}) = q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_t) q_{\text{tar}}(\mathbf{z}_{<t} \mid \mathbf{x}_{\leq t}). \tag{4}$$

The distributions q_{inf} and q_{tar} have different roles:

- q_{inf} reflects the generative model's transition dynamics and combines it with the current observation $\mathbf{x}_t$. It is a Gaussian distribution whose parameters are obtained by propagating $\mathbf{z}_{<t}$ through the RNN of the generative model and using an inference network to combine the output with $\mathbf{x}_t$.
- q_{tar} is a distribution we will use to sample past states for approximating the marginalization in Equation (3). Its name suggests that it is generally intractable and will be approximated via self-normalized importance sampling.

The variational posterior of VDM marginalizes over past states (Equation (3)). The target distribution specifies how past states are sampled and the inference distribution specifies how the new observation should correct the distribution over latent states. In the simplest version of VDM sampling from the target distribution corresponds to sampling from previous posteriors. Then we show that we can add modeling flexibility by using self-normalized weighted sampling for the target distribution.

3.2.1. Parametrization of the Variational Posterior

The VDM inference approach uses the same RNN as the generative model to track the history of the latent states. By using the RNN to summarize information from past states, sampling from the target distribution can be done efficiently. Using previously inferred posteriors as the target distribution, $q_{\text{tar}}(\mathbf{z}_{<t} \mid \mathbf{x}_{\leq t}) := q(\mathbf{z}_{<t} \mid \mathbf{x}_{<t})$, past states are sampled sequentially as follows. At each time step t, we sample K samples from the previous posterior $\mathbf{z}_{t-1}^{(i)} \sim q(\mathbf{z}_{t-1} \mid \mathbf{x}_{<t})$ indexed by i and these samples are aggregated by the RNN (with same parameters ϕ^{GRU} as in the generative model.)

$$\mathbf{z}_{t-1}^{(i)} \sim q(\mathbf{z}_{t-1} \mid \mathbf{x}_{<t}) \qquad \mathbf{h}_t^{(i)} = \phi^{\text{GRU}}(\mathbf{z}_{t-1}^{(i)}, \hat{\mathbf{h}}_{t-1}), \qquad \hat{\mathbf{h}}_t = \mathbb{E}_{q_{\text{tar}}(\mathbf{z}_{<t} \mid \mathbf{x}_{\leq t})}[\mathbf{h}_t]\,. \tag{5}$$

We initialize $\hat{\mathbf{h}_1}$ to a vector of zeros. To avoid an exponential blow-up of the number of samples as t increases, we compute an expected history $\hat{\mathbf{h}}_t$ for the recursion of the RNN.

To evaluate the inference distribution on each of the samples, an inference network ϕ^{inf} combines the output of the RNN with the new observation $\mathbf{x}_t$ to produce the mean and variance of q_{inf} that we assume to be Gaussian

$$q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t) = \mathcal{N}(\mu_{\text{inf},t}^{(i)}, \sigma_{\text{inf},t}^{(i)2}\mathbb{I}), \quad \text{where} \quad [\mu_{\text{inf},t}^{(i)}, \sigma_{\text{inf},t}^{(i)2}] = \phi^{inf}(\mathbf{h}_t^{(i)}, \mathbf{x}_t)\,. \tag{6}$$

We use the notation $\mathbf{z}_{<t}^{(i)}$ to indicate that the parameters of the distribution are computed as a function of $\mathbf{h}_t^{(i)}$ as defined in Equation (5). By using the transition dynamics of the generative model, the inference model can focus its capacity on learning how to account for the new observation when inferring $\mathbf{z}_t$. Given samples from the target distribution, we can approximate the marginalization in Equation (3) to obtain

$$\begin{aligned} q(\mathbf{z}_t \mid \mathbf{x}_{\leq t}) &= \mathbb{E}_{q_{\text{tar}}(\mathbf{z}_{<t}\mid\mathbf{x}_{\leq t})}[q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_t)] \\ &\approx \sum_{i=1}^{K} \omega_t^{(i)} q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t) = \sum_{i=1}^{K} \omega_t^{(i)} \mathcal{N}(\mu_{\text{inf},t}^{(i)}, \sigma_{\text{inf},t}^{(i)2}\mathbb{I}), \quad \text{where} \quad \omega_t^{(i)} = \frac{1}{K}. \end{aligned} \tag{7}$$

The marginal variational posterior becomes an equally weighted mixture density network [40], which is a good choice for modeling multi-modal dynamics (as our experiments show). The variational posterior of VDM can gain additional modeling flexibility by choosing different parametrizations for the mixture weights.

3.2.2. Generalized Mixture Weights

Assume that we chose a target distribution that is different from the past approximate posterior $q_{\text{tar}}(\mathbf{z}_{<t} \mid \mathbf{x}_{\leq t}) \neq q(\mathbf{z}_{<t} \mid \mathbf{x}_{<t})$. If we still use samples from the past posterior to approximate the marginalization in Equation (7), the *importance weights* $\omega_t^{(i)}$ have to correct for the discrepancy between the base distribution (approximate posterior) and the target distribution q_{tar} ([41], Ch. 9). In a more general variational family for VDM than described above, the target distribution does not equal the base distribution. In this generalized setting, instead of choosing a parametrization for q_{tar} and then deriving the importance weights, we directly choose how to parameterize the weights which we ensure are self-normalized ([41], Ch. 9.2). We choose the generalized weights to be,

$$\omega_t^{(i)} := \omega(\mathbf{x}_t, \mathbf{z}_{<t}^{(i)}) / \left(\sum_{j=1}^{K} \omega(\mathbf{x}_t, \mathbf{z}_{<t}^{(j)}) \right), \tag{8}$$

$$\text{where} \qquad \omega(\mathbf{x}_t, \mathbf{z}_{<t}) := p(\mathbf{x}_t \mid \mathbf{z}_{<t}) = \mathbb{E}_{\mathbf{z}_t}[p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}) p(\mathbf{z}_t \mid \mathbf{z}_{<t})]. \tag{9}$$

With this definition the weights are normalized by construction, $\sum_{i=1}^{K} \omega_t^{(i)} = 1$. We could choose any finite and non-negative expression for $\omega(\mathbf{x}_t, \mathbf{z}_{<t})$. As in importance sampling for bootstrap particle filters [42], our choice of weights takes into account each sample's relevance for predicting the new observation $\mathbf{x}_t$. Another advantage is that we do not introduce additional variational parameters. The only variational parameters of the VDM inference model are the neural network parameters of ϕ^{inf}. The predictive likelihood $p(\mathbf{x}_t \mid \mathbf{z}_{\leq t})$, can be computed by plugging in samples $\mathbf{z}_{<t}^{(i)}$, that are sampled and aggregated according to Equation (5), into the generative model (Equations (1) and (2)). Pseudo code for the generative and the inference model are in Algorithms 1 and 2.

Algorithm 1: Generative model.

```
Inputs: $\mathbf{z}_\tau, \mathbf{h}_\tau$
Outputs: $\mathbf{x}_{\tau+1:T}$
for $t = \tau + 1 : T$ do
    $\mathbf{h}_t = \phi^{\text{GRU}}(\mathbf{z}_{t-1}, \mathbf{h}_{t-1})$
    $[\mu_{0,t}, \sigma_{0,t}^2] = \phi^{tra}(\mathbf{h}_t)$ {Equation (1)}
    $\mathbf{z}_t \sim \mathcal{N}(\mu_{0,t}, \sigma_{0,t}^2\mathbb{I})$
    $[\mu_{x,t}, \sigma_{x,t}^2] = \phi^{dec}(\mathbf{z}_t, \mathbf{h}_t)$ {Equation (2)}
    $\mathbf{x}_t \sim \mathcal{N}(\mu_{x,t}, \sigma_{x,t}^2\mathbb{I})$
end for
```

Algorithm 2: Inference model.

```
**Inputs:** $\mathbf{x}_{1:\tau}, \hat{\mathbf{h}}_1$
**Outputs:** $\mathbf{z}_{1:\tau}, \hat{\mathbf{h}}_\tau$
$[\mu_{\text{inf},1}, \sigma^2_{\text{inf},1}] = \phi^{inf}(\mathbf{x}_1, \hat{\mathbf{h}}_1)$
$\mathbf{z}_1^{(i)} \sim \mathcal{N}(\mu_{z,1}, \sigma^2_{z,1}\mathbb{I})$
**for** $t = 2 : \tau$ **do**
    $\mathbf{h}_t^{(i)} = \phi^{\text{GRU}}(\mathbf{z}_{t-1}^{(i)}, \hat{\mathbf{h}}_{t-1})$ {Equation (5)}
    $[\mu^{(i)}_{\text{inf},t}, \sigma^{(i)2}_{\text{inf},t}] = \phi^{inf}(\mathbf{x}_t, \mathbf{h}_t^{(i)})$ {Equation (6)}
    $\omega_t^{(i)} := \omega(\mathbf{x}_t, \mathbf{h}_t^{(i)}) / \sum_{j=1}^{K} \omega(\mathbf{x}_t, \mathbf{h}_t^{(j)})$ {Equation (8)}
    $\mathbf{z}_t^{(i)} \sim \sum_i^k \omega_t^{(i)} \mathcal{N}(\mu^{(i)}_{\text{inf},t}, \sigma^{(i)2}_{\text{inf},t}\mathbb{I})$
    $\hat{\mathbf{h}}_t = \sum_i^k \omega_t^{(i)} \mathbf{h}_t^{(i)}$
**end for**
```

It is interesting to wonder about the connections to structured variational inference. If we do not marginalize over $\mathbf{z}_{<t}$ but rather condition on it (use the same inference distribution q_{inf}), we obtain the structured variational approximation used in the conventional VRNN approach. The advantage of instead carrying out the marginalization is that we explore multiple modes of the transition dynamics. Approximating the marginalization in Equation (3) with a single sample ($K = 1$), recovers the inference model of VRNN [5].

3.3. *The Variational Objective of VDM*

VDM is fit with a variational objective. It consists of the ELBO terms and an optional regularization term that is helpful to improve the performance. In our empirical study, we investigate the effect of the regularization term both for VDM and for other existing methods. We found that when the method worked well without the regularization term, the regularization term gave an additional performance boost, especially on the qualitative results.

We will first describe the ELBO for VDM and then motivate and explain the regularization term. As in [37] the ELBO is derived based on the augmented model in Equation (4). The main challenge is to lower-bound the entropy of the augmented variational distribution, which contains an implicit component. In Appendix A, we show that this quantity can be lower-bounded and that the lower bound can be estimated using the reparameterization trick. The resulting instantaneous ELBO is:

$$\begin{aligned} &\log p(\mathbf{x}_{1:T}) \geq \mathcal{L}_{\text{ELBO}} \qquad (10) \\ &= \sum_{t=1}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t | \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)} \left[\log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{<t}^{(i)}) - \log \left(\sum_{i=1}^{K} \omega_t^{(i)} q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t) \right) \right] \\ &+ \mathbb{E}_{q(\mathbf{z}_1 | \mathbf{x}_1)}[\log p(\mathbf{z}_1)] + \sum_{t=2}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t | \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)} \left[\log p(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}) \right] \end{aligned}$$

Given a dataset $\mathcal{D}$, VDM's parameters of the generative and inference model $\phi = [\phi^{tra}, \phi^{dec}, \phi^{\text{GRU}}, \phi^{inf}]$ are obtained by minimizing the loss

$$\mathcal{L}_{\text{VDM}}(\phi) = \mathbb{E}_{\mathcal{D}} \left[-\mathcal{L}_{\text{ELBO}}(\phi) - \sum_{t=1}^{T} \lambda \mathcal{L}_{pred,t}(\phi) \right], \qquad (11)$$

with a hyperparameter λ determining the strength of the regularization. We propose to augment the ELBO with a *prediction term*. We empirically compare the effect of including and excluding the regularization term in the objective. VDM is competitive without the prediction term, but we got the strongest when including the regularization term $\mathcal{L}_{pred,t}$. We set the hyper-parameter $\lambda = 1$, though an additional performance boost could be obtained by tuning it on the validation set.

The prediction term $\mathcal{L}_{pred}$, encourages the variational posterior (from the previous time step) to produce samples that maximize the predictive likelihood,

$$\mathcal{L}_{pred,t}(\phi) = \log \mathbb{E}_{q(\mathbf{z}_{t-1}|\mathbf{x}_{<t})}[p(\mathbf{x}_t \mid \mathbf{z}_{<t})] \approx \log \frac{1}{K}\sum_{i}^{K} p(\mathbf{x}_t \mid \mathbf{z}_{t-1}^{(i)}, \mathbf{h}_{t-1}), \tag{12}$$

the likelihood under each sample $p(\mathbf{x}_t \mid \mathbf{z}_{t-1}^{(i)}, \mathbf{h}_{t-1})$ is assumed to be Gaussian. The mean and variance of this distribution are computed by propagating the sample through the transition model (Equation (1)) and the result through the emission model (Equation (2)) (see Algorithm 1.) This regularization term is helpful to improve the prediction performance since it depends on the predictive likelihood of samples, which is not involved in the ELBO.

3.4. Alternative Modeling Choices

Next, we discuss alternative implementations of VDM that are optional, but can enhance the expressiveness of the model.

Our method involves sampling from Gaussian distributions at multiple steps. While Monte-Carlo (MC) methods work, it turns out that we can achieve better results with fewer samples by drawing on so-called cubature approximations [43–45], which choose samples more carefully. In our stochastic cubature approximation (SCA), the usually deterministically-chosen curbature points are further randomized for better performance, allowing us to use fewer samples than in naive MC. See Appendix B for more details.

An alternative choice of the expression for the weights is

$$\omega(\mathbf{x}_t, \mathbf{z}_{<t}^{(i)}) := \mathbb{1}(i = \arg\max_{j\in[1,\cdots,K]} p(\mathbf{x}_t \mid \mathbf{z}_{<t}^{(j)})). \tag{13}$$

which corresponds to a hard choice between the samples. Only the component associated with the sample that achieves the highest predictive likelihood is nonzero. We stress that this choice for the weights still corresponds to a multi-modal posterior approximation: all K mixture components that result from propagating different latent states $\mathbf{z}_{<t}^{(j)}$ through the GRU are considered as candidate modes, and the most likely mixture component is selected after new data is observed. Even though each single observation is assigned only to a single mode, the combination of the modes (namely a mixture) is used to model the entire data. Similarly as in "best-of-many" sampling [22], the zeroed-out components in the mixture density network have the capacity to focus on other modes. We found the hard choice works well in our empirical study and use it as the default choice for VDM.

4. Evaluation and Experiments

In this section, we evaluate VDM's ability to model multi-modal dynamics and show its competitive forecasting performance in various domains. We first introduce the evaluation metrics, baselines and summarize all ablations. Experiments on synthetic data demonstrate that VDM is truly multi-modal thereby supporting the modeling choices of Section 3, especially for the inference model. Experiments on real-world datasets with challenging multi-modal dynamics show the benefit of VDM over state-of-the-art (deep) probabilistic time-series models.

4.1. Evaluation Metrics

In the experiments, we create a training set, a validation set, and a test set. During validation and test, each trajectory is split into two parts; initial observations (given to the models for inference) and continuations of the trajectory (to be predicted and not accessible to the models). The inference models are used to process the initial observations and to infer latent states. These are then processed by the generative models to produce forecasts.

We use 3 criteria to evaluate these forecasts (i) multi-step prediction $p(\mathbf{x}_{t+1:t+\tau} \mid \mathbf{x}_{1:t})$, (ii) one-step-ahead prediction $p(\mathbf{x}_{t+1} \mid \mathbf{x}_{1:t})$, and (iii) a new metric inspired by the Wasserstein distance. As in other work [4,22,46], (i) and (ii) are reported in terms of negative log-likelihood. When the models' predictive distribution for one-step-ahead prediction is assumed to be Gaussian, its negative log-likelihood can be computed in closed form. However, the long-term forecasts have to be evaluated using samples. For each ground truth $\mathbf{x}$ we generate $n = 1000$ forecasts $\hat{\mathbf{x}}_i$ given initial observations from the beginning. For a fair comparison with methods that do not output a predictive variance, we choose a constant variance.

$$NLL = -\log\left(\frac{1}{n}\sum_{i}^{n}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(\hat{\mathbf{x}}_i - \mathbf{x})^2}{2}\right)\right). \tag{14}$$

This evaluates the predictive accuracy but neglects a key aspect of multi-modal forecasts – diversity. We propose a new evaluation metric, which takes both diversity and accuracy of predictions into account. Inspired by the Wasserstein distance [7], we compute the distance between the ground truth distribution $\mathbf{X}$ and the model distribution $\hat{\mathbf{X}}$ as

$$W(\mathbf{X}, \hat{\mathbf{X}}) = \inf_{\pi}\left(\frac{1}{n}\sum_{i}^{n}\|(\mathbf{x}_i - \hat{\mathbf{x}}_{\pi(i)}\|_2\right), \tag{15}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the ground truth sequences and model forecasts, and π denotes all permutations. We select n samples from the test set with similar initial observations. The model is expected to generate samples matching all ground truth continuations given the initial observations. The model generates $10 \times n$ forecasts. We compute the distance between n ground truth sequences and the top n well-matched predictions with Equation (15). Since the forecasts do not match with ground truth sequences one to one well due to the randomness, we generate more forecasts to mitigate the variance of the results. We report the average of W-distances over different initial observations.

4.2. Baselines

We choose baselines from three classes of models. Two stochastic recurrent models are variational recurrent neural network (VRNN) [5] and auto-encoding sequential Monte Carlo (AESMC) [1]. VRNN has a similar but more powerful generative model than VDM, and AESMC uses SMC to achieve a tighter lower bound. However, compared to VDM, both use the structured variational approximation rather than marginalizing over past states. Two deep SSMs are recurrent Kalman network (RKN) [6] and deep Markov model [2] with variational posteriors based on inverse autoregressive flows [3] (DMM-IAF). RKN models the latent space with locally linear SSMs. DMM-IAF is a nonlinear deep SSM leveraging a structured variational inference with flexible variational distributions based on flows. A final baseline is conditional flow variational autoencoder (CF-VAE) [4], a global latent variable model based on normalizing flows.

For fair comparisons, we add recurrent states to DMM-IAF, and fix the dimension of the latent variables $\mathbf{z}_t$ and $\mathbf{h}_t$ to be the same for VDM, AESMC, DMM-IAF and VRNN which have the same resulting model size (except for the additional autoregressive feedback in VRNN, and additional flows in DMM-IAF). AESMC and VDM use the same number of samples. RKN does not have recurrent states, so we choose a higher latent dimension to make model size comparable. CF-VAE has only one global latent variable which needs more capacity and we make it higher-dimensional than $\mathbf{z}_t$. Implementation details are in Appendix D. Since $\mathcal{L}_{pred}$ can be easily applied to all baselines except for CF-VAE, we trained them with or without $\mathcal{L}_{pred}$, and report the best results.

4.3. Ablations

VDM has many ingredients; the type of sampling method, different approximation schemes for the expectations w.r.t. q_{tar}, and the optional regularization term, which can also be beneficial to existing methods. To disentangle the contributions of the varying ingredients we include an extensive ablation study. The definition of all VDM variants is in Table 1. VDM is the default model using improved Gaussian sampling, *hard* weights for the mixtures (Equation (13)), and trained with $\mathcal{L}_{\text{VDM}}$. In VDM ($\mathcal{L}_{\text{ELBO}}$), we study the contribution of the prediction term and only use $\mathcal{L}_{\text{ELBO}}$ as the training objective. In VDM-SCA-S, we use improved Gaussian sampling and *soft* weights (Equation (9)) instead, in VDM-MC-S, we use Monte-Carlo sampling and soft weights, while in VDM-MC-U, we use Monte-Carlo sampling and uniform weights ($\omega_t^{(i)} = \frac{1}{K}$). The comparison of them allows us to understand the effect of different modeling choices: various VDM variants typically outperform the sequential latent variable baselines, and the fine-tuned modeling choices provide a performance boost (shown in Tables 2 and 3).

Table 1. Definition of VDM variants. By tuning the modeling choices of sampling (MC sampling or SCA in Section 3.4), weights (uniform weights, soft weights in Equation (9), or hard weights in Equation (13)), and the loss function (with or without $\mathcal{L}_{pred}$), we propose 5 variants of VDM.

	VDM	**VDM ($\mathcal{L}_{\text{ELBO}}$)**	**VDM-SCA-S**	**VDM-MC-S**	**VDM-MC-U**
Sampling	SCA	SCA	SCA	Monte-Carlo	Monte-Carlo
Weights	hard	hard	soft	soft	uniform
Loss	$\mathcal{L}_{\text{VDM}}$	$-\mathcal{L}_{\text{ELBO}}$	$\mathcal{L}_{\text{VDM}}$	$\mathcal{L}_{\text{VDM}}$	$\mathcal{L}_{\text{VDM}}$

4.4. Results

We evaluate VDM on synthetic data and three real-world datasets: taxi trajectories, NBA SportVu data, and U.S. pollution data. The experiment on synthetic data demonstrates that VDM is truly multi-modal. By comparing with existing methods on real-world datasets, we show the benefit of VDM over state-of-the-art (deep) probabilistic time-series models.

4.4.1. Synthetic Data with Multi-Modal Dynamics

We generate synthetic data with two dimensions and four modes and compare the performance of VDM with 9 samples (Figure 2, left), DMM-IAF (Figure 2, middle), and AESMC using 9 particles (Figure 2, right). Since variational inference is known to try to match the aggregated posterior with the predictive prior [47], it is instructive to fit all three models and to look at their predictive prior $p(\mathbf{z}_2|\mathbf{x}_{\leq 1})$ and the aggregated posterior $p(\mathbf{z}_2|\mathcal{D})$. Because of the multi-modal nature of the problem, all 3 aggregated posteriors are multi-modal, but only VDM ($K = 9$) learns a multi-modal predictive prior (thanks to its choice of the variational family). Although AESMC and DMM-IAF with flexible structured variational distributions achieve a good match between the prior and the aggregated posterior, the predictive prior does not clearly separate into different modes. In contrast, the inference model of VDM successfully uses multiple samples and explores multiple modes of the transition dynamics to separate latent states into separate modes.

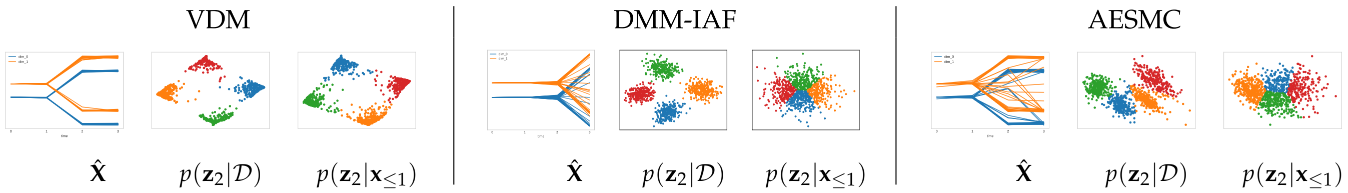


Figure 2. Experiments on 2d synthetic data with 4 modes highlight the multi-modality of VDM. We train VDM (left), DMM-IAF (middle), and AESMC (right) on a training set of trajectories $\mathcal{D}$ of length 4, and plot generated trajectories $\hat{\mathbf{X}}$ (2 colors for 2 dimensions). VDM and AESMC both use 9 samples. We also plot the aggregated posterior $p(\mathbf{z}_2|\mathcal{D})$, and the predictive prior $p(\mathbf{z}_2|\mathbf{x}_{\leq 1})$ (4 colors for 4 clusters, and not related to the colors in the trajectories plot) at the second time step. Only VDM learns a multi-modal predictive prior, which explains its success in modeling multi-modal dynamics.

4.4.2. Stochastic Lorenz Attractor

The Lorenz attractor is a deterministic system governed by ordinary differential equations. Under certain parameter settings, it is chaotic—even small errors can cause considerable differences in the future. We add noise to the transition and emission function to make it stochastic (details in Appendix C). All models are trained and then tasked to predict 90 future observations given 10 initial observations. Figure 3 illustrates qualitatively that VDM (Figure 3b), AESMC (Figure 3c), and DMM-IAF (Figure 3d) succeed in modeling the chaotic dynamics of stochastic Lorenz attractor, while CF-VAE (Figure 3e) and VRNN (Figure 3f) miss local details, and RKN (Figure 3g) which lacks the capacity for stochastic transitions does not work at all. In terms of quantitative results, VDM achieves the best scores on multi-step prediction and W-distance, while VDM-MC-S works best on one-step prediction (Table 2). VDM ($\mathcal{L}_{\text{ELBO}}$) does not include $\mathcal{L}_{pred}$ in the training and is therefore outperformed by other VDM variants. The baselines AESMC and DMM-IAF also give comparable results. Since the dynamics of Lorenz attractor are governed by ordinary differential equations, the transition dynamics at each time step are not obviously multi-modal, which explains why all models with stochastic transitions do reasonably well. Next, we will show the advantages of VDM on real-world data with multi-modal dynamics.

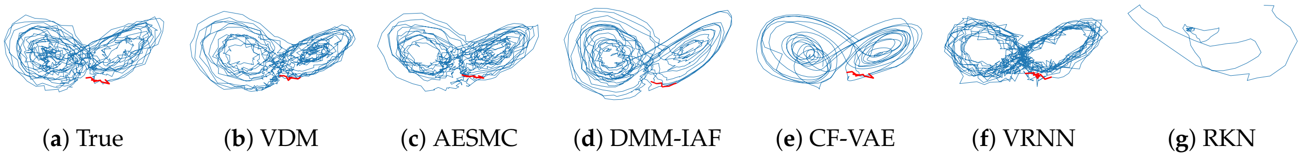


Figure 3. Generated samples from VDM and baselines for stochastic Lorenz attractor. The models generate the future 990 steps (blue) based on the first 10 observations (red). Due to the chaotic property, the reconstruction is impossible even the model learns the right dynamics. VDM, AESMC, and DMM-IAF capture the stochastic dynamics well, while RKN fails.

4.4.3. Taxi Trajectories

The taxi trajectory dataset involves taxi trajectories in Porto, Portugal. Each trajectory is a sequence of two-dimensional locations over time. Here, we cut the trajectories to a fixed length of 30 to simplify the comparison (details in Appendix C). The task is to predict the next 20 observations given 10 initial observations. Ideally, the forecasts should follow the street map (though the map is not accessible to the models). The results in Table 2 show that VDM variants typically outperform the other *sequential* latent variable models quantitatively. By tuning the modeling choices of sampling, weights, and the objective, VDM achieves the best results on the one-step prediction and W-distance that measures both diversity and accuracy of predictions. CF-VAE which is a global latent variable model, achieves the lowest negative log-likelihood in multi-step prediction. However, this value does not match the qualitative results in Figure 1. Since CF-VAE has to encode the entire structure of the trajectory forecast into a single latent variable, its predictions seem to average over plausible continuations but are locally neither plausible nor accurate. In comparison, VDM and the other models involve a sequence of latent variables. As the forecasting progresses, the impact of the initial observations becomes weaker and weaker. As a result, local structure can be captured more accurately. While the forecasts are plausible and can be highly diverse, they potentially evolve into other directions than the ground truth. For this reason, their multi-step prediction results are worse in terms of log-likelihood. That is why the empirical W-distance is useful to complement the evaluation of multi-modal tasks. It reflects that the forecasts of VDM are diverse and plausible. Additionally, we illustrate the predictive prior $p(\mathbf{z}_t|\mathbf{x}_{<t})$ at different time steps in Figure 4. VDM learns a multi-modal predictive prior, while AESMC and DMM-IAF result in an uni-modal predictive prior, even though they employ flexible variational distributions.

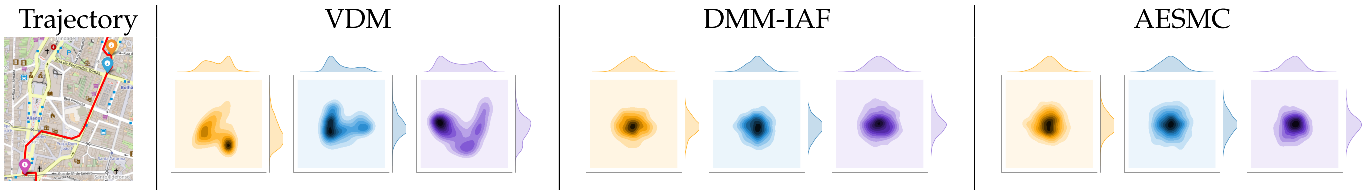


Figure 4. An illustration of predictive priors $p(\mathbf{z}_t|\mathbf{x}_{<t})$ of taxi trajectories from VDM, DMM-IAF, and AESMC at 3 forks in the road marked on the map. VDM and AESMC both use 13 samples. VDM succeeds in capturing the multi-modal distributions, while DMM-IAF and AESMC approximate them with uni-modal distributions. For visualization, the distributions are projected to 2d with KDE.

Table 2. Prediction error on stochastic Lorenz attractor and taxi trajectories for three evaluation metrics (details in main text). On the stochastic Lorenz attractor, VDM achieves the best performance. AESMC and DMM-IAF also give comparable results. On the taxi trajectories, CF-VAE achieves the best result in multi-step ahead prediction, since it uses a global variable, that guides the trajectories into generally the right direction. Meanwhile VDM variants outperform all sequential models, and outperform CF-VAE on the other metrics. To test different modeling choices we include the VDM variants of Table 1.

	Stochastic Lorenz Attractor			**Taxi Trajectories**		
	Multi-Step	**One-Step**	**W-Distance**	**Multi-Step**	**One-Step**	**W-Distance**
RKN	104.41	1.88	16.16	4.25	−2.90	2.07
VRNN	65.89 ± 0.21	−1.63	16.14 ± 0.006	5.51 ± 0.002	−2.77	2.43 ± 0.0002
CF-VAE	32.41 ± 0.13	n.a	8.44 ± 0.005	**2.77 ± 0.001**	n.a	0.76 ± 0.0003
DMM-IAF	25.26 ± 0.24	−1.29	7.47 ± 0.014	3.29 ± 0.001	−2.45	0.70 ± 0.0003
AESMC	25.01 ± 0.22	−1.69	**7.29 ± 0.005**	3.31 ± 0.001	−2.87	0.66 ± 0.0004
VDM	**24.49 ± 0.16**	−1.81	**7.29 ± 0.003**	2.88 ± 0.002	−**3.68**	**0.56 ± 0.0008**
VDM($\mathcal{L}_{\text{ELBO}}$)	25.01 ± 0.27	−1.74	7.30 ± 0.004	3.10 ± 0.005	−3.05	0.61 ± 0.0003
VDM−SCA−S	24.69 ± 0.16	−1.83	7.30 ± 0.009	3.09 ± 0.001	−3.24	0.64 ± 0.0005
VDM−MC−S	24.67 ± 0.16	−**1.84**	7.30 ± 0.005	3.17 ± 0.001	−3.21	0.68 ± 0.0008
VDM−MC−U	25.04 ± 0.28	−1.81	7.31 ± 0.002	3.30 ± 0.002	−2.42	0.69 ± 0.0002

Table 3. Prediction error on basketball players' trajectories and U.S. pollution data for two evaluation metrics (details in main text). VDM makes the most accurate multi-step and one-step ahead predictions. The tested variants of VDM are defined in Table 1.

	NBA SportVu		**US Pollution**	
	Multi-Steps	**One-Step**	**Multi-Steps**	**One-Step**
RKN	4.88	1.55	53.13	6.98
VRNN	5.42 ± 0.009	−2.78	49.32 ± 0.13	8.69
CF-VAE	3.24 ± 0.003	n.a	45.86 ± 0.04	n.a
DMM-IAF	3.63 ± 0.002	−3.74	44.82 ± 0.11	9.41
AESMC	3.74 ± 0.003	−3.91	41.14 ± 0.13	6.93
VDM	**3.23 ± 0.003**	**−5.44**	**37.64 ± 0.07**	**6.91**
VDM($\mathcal{L}_{\text{ELBO}}$)	3.29 ± 0.003	−5.04	39.87 ± 0.04	7.60
VDM−SCA−S	3.31 ± 0.001	−5.08	39.58 ± 0.09	7.82
VDM−MC−S	3.35 ± 0.007	−5.00	40.33 ± 0.03	8.12
VDM−MC−U	3.39 ± 0.006	−4.82	41.81 ± 0.10	7.71

4.4.4. NBA SportVu Data

This dataset (A version of the dataset is available at https://www.stats.com/data-science/, accessed on 1 September 2020) consists of sequences of 2D coordinates describes the movements of basketball players and the ball. We extract the trajectories and cut them to a fixed length of 30 to simplify the comparisons (details in Appendix C). The task is to predict the next 20 observations given 10 initial observations. Players can move anywhere on the court and hence their movement is less structured than taxi trajectories that are constrained by the underlying street map. Due to this, the initial movement patterns are not similar enough to evaluate W-distance. VDM outperforms all baselines and other VDM variants in the multi-step prediction and one-step prediction (Table 3). Other VDM variants perform also reasonably well and better than the other *sequential* latent variable models. Figure 5 illustrates qualitatively that VDM (Figure 5b) and CF-VAE (Figure 5e) succeed in capturing the multi-modal dynamics. The forecasts of AESMC (Figure 5c) and DMM-IAF (Figure 5d) are less plausible (not as smooth as data). VRNN (Figure 5f) and RKN (Figure 5g) fail in capturing the multi-modality.

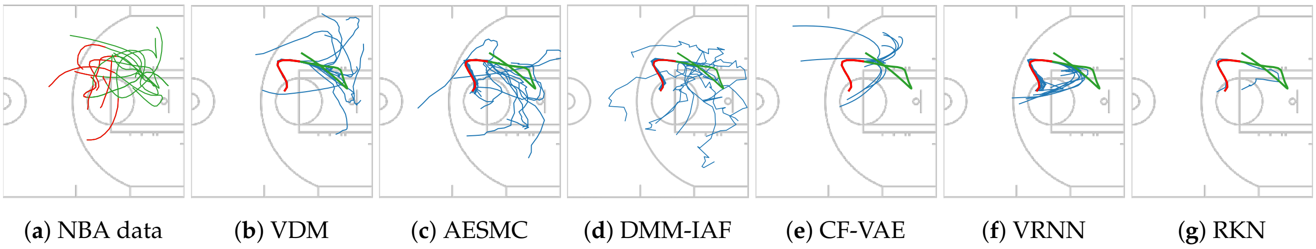

(**a**) NBA data (**b**) VDM (**c**) AESMC (**d**) DMM-IAF (**e**) CF-VAE (**f**) VRNN (**g**) RKN

Figure 5. VDM and CF-VAE generate plausible multi-modal trajectories of basketball plays. Each model's forecasts (blue) are based on the first 10 observations (red). Ground truth data is green.

4.4.5. U.S. Pollution Data

(https://www.kaggle.com/sogun3/uspollution, accessed on 1 March 2020).In this experiment, we study VDM on the U.S. pollution dataset (details in Appendix C). The data is collected from counties in different states from 2000 to 2016. Each observation has 12 dimensions (mean, max value, and air quality index of NO_2, O_3, SO_2, and CO). The goal is to predict monthly pollution values for the coming 18 months, given observations of the previous 6 months. We ignore the geographical location and time information to treat the development tendency of pollution in different counties and different times as i.i.d. The unknown context information makes the dynamics multi-modal and challenging to predict accurately. Due to the small size and high dimensionality of the dataset, there are not enough samples with very similar initial observations. Thus, we cannot evaluate empirical W-distance in this experiment. VDM outperforms all baselines in both evaluations (Table 3).

5. Conclusions

We presented variational dynamic mixtures (VDM), a new approach to inference in sequential latent variable models that improves the model's ability to forecast multi-modal dynamics. The main ideas of VDM is a mean-field factorization with history marginalization, which introduces more complete information about previously inferred dynamics into the variational factors. We also promoted the Wasserstein-distance like metric to evaluate multi-modal forecasting tasks. VDM succeeds in learning challenging multi-modal dynamics and outperforms existing methods on a variety of data sets.

Author Contributions: Conceptualization and methodology, M.R., C.Q. and S.M.; software and validation, C.Q.; writing, M.R., S.M. and C.Q.; supervision, M.R. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: Stephan Mandt acknowledges support by DARPA under contract No. HR001120C0021, the Department of Energy under grant DE-SC0022331, the National Science Foundation under the NSF CAREER award 2047418 and Grants 1928718, 2003237 and 2007719, as well as gifts from Intel, Disney, and Qualcomm. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or NSF.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The Bosch Group is carbon neutral. Administration, manufacturing and research activities do no longer leave a carbon footprint. This also includes GPU clusters on which the experiments for this study have been performed.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. ELBO Derivations

The generative model is

$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{z}_1) \prod_{t=2}^{T} p(\mathbf{z}_t | \mathbf{z}_{<t}) \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{z}_{\leq t}) \,. \tag{A1}$$

The inference model is

$$q(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{x}_{\leq t}) = \prod_{t=1}^{T} \int q_{\text{inf}}(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{x}_t) q_{\text{tar}}(\mathbf{z}_{<t} | \mathbf{x}_{\leq t}) d\mathbf{z}_{<t}. \tag{A2}$$

The KL divergence between the approximate posterior and the true posterior of $\mathbf{z}_{1:T}$ is

$$\text{KL}(q(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) \mid\mid p(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})) = \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} \left[\log q(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) - \log \frac{p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T})}{p(\mathbf{x}_{1:T})} \right] \tag{A3}$$

and since the KL divergence is non-negative we get the following evidence lower bound

$$\begin{aligned} &\log p(\mathbf{x}_{1:T}) \geq \\ &\mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t | \mathbf{z}_{\leq t}) + \log p(\mathbf{z}_1) + \sum_{t=2}^{T} \log p(\mathbf{z}_t \mid \mathbf{z}_{<t}) - \log q(\mathbf{z}_{1:T} | \mathbf{x}_{1:T}) \right] . \end{aligned} \tag{A4}$$

We derive each of the three terms (the reconstruction, the cross-entropy, and the entropy) separately. This is the derivation of the reconstruction term:

$$\mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t | \mathbf{z}_{\leq t}) \right]$$

$$= \sum_{t=1}^{T} \int q_{\text{aug}}(\mathbf{z}_{\leq t} \mid \mathbf{x}_{\leq t}) [\log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{<t})] d\mathbf{z}_{\leq t} \tag{A5}$$

$$= \sum_{t=1}^{T} \iint q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_t) q_{\text{tar}}(\mathbf{z}_{<t} \mid \mathbf{x}_{\leq t}) [\log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{<t})] d\mathbf{z}_{<t} d\mathbf{z}_t \tag{A6}$$

$$\approx \sum_{t=1}^{T} \int \sum_{i=1}^{K} \omega_t^{(i)} q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t) \log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{<t}^{(i)}) d\mathbf{z}_t \tag{A7}$$

$$= \sum_{t=1}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t | \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)} \left[\log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{<t}^{(i)}) \right] , \tag{A8}$$

In Equations (A5) and (A6) we have used the definition of the approximate posterior from Equations (3) and (4), and in Equation (A7) we approximate the integration over the target distribution with samples as defined in Section 3.2.

This is the derivation of the negative cross-entropy term:

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})}\left[\log p(\mathbf{z}_1) + \sum_{t=2}^{T} \log p(\mathbf{z}_t \mid \mathbf{z}_{<t})\right] \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}[\log p(\mathbf{z}_1)] + \sum_{t=2}^{T} \int q_{\text{aug}}(\mathbf{z}_{\leq t} \mid \mathbf{x}_{\leq t})[\log p(\mathbf{z}_t \mid \mathbf{z}_{<t})]\mathrm{d}\mathbf{z}_{\leq t} && \text{(A9)} \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}[\log p(\mathbf{z}_1)] + \sum_{t=2}^{T} \iint q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}, \mathbf{x}_t) q_{\text{tar}}(\mathbf{z}_{<t} \mid \mathbf{x}_{\leq t})[\log p(\mathbf{z}_t \mid \mathbf{z}_{<t})]\mathrm{d}\mathbf{z}_{<t}\mathrm{d}\mathbf{z}_t && \text{(A10)} \\
&\approx \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}[\log p(\mathbf{z}_1)] + \sum_{t=2}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t|\mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)}\left[\log p(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)})\right]. && \text{(A11)}
\end{aligned}$$

Again, we plug Equations (3), (4) and (7) into approximating the integral over $\mathbf{z}_{<t}$.

This is the derivation of the entropy term:

$$\begin{aligned}
&-\mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})}[\log q(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})] \\
&= -\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_{\leq t})}[\log q(\mathbf{z}_t \mid \mathbf{x}_{\leq t})] && \text{(A12)} \\
&\approx -\sum_{t=1}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t|\mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)}\left[\log\left(\sum_{i=1}^{K} \omega_t^{(i)} q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)\right)\right]. && \text{(A13)}
\end{aligned}$$

Plugging these all together into Equation (A4), we get the ELBO.

$$\begin{aligned}
\log p(\mathbf{x}_{1:T}) \geq \mathcal{L}_{\text{ELBO}} = &\sum_{t=1}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t|\mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)}\left[\log p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{<t}^{(i)}) - \log\left(\sum_{i=1}^{K} \omega_t^{(i)} q_{\text{inf}}(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)\right)\right] && \text{(14)} \\
&+ \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}[\log p(\mathbf{z}_1)] + \sum_{t=2}^{T} \sum_{i=1}^{K} \omega_t^{(i)} \mathbb{E}_{q_{\text{inf}}(\mathbf{z}_t|\mathbf{z}_{<t}^{(i)}, \mathbf{x}_t)}\left[\log p(\mathbf{z}_t \mid \mathbf{z}_{<t}^{(i)})\right].
\end{aligned}$$

Since q_{inf} is Gaussian, the expectations are computed with samples. With the weights defined as one hot vectors as in Section 3.4, the computation simplifies further.

Appendix B. Supplementary to Stochastic 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 $\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2\mathbb{I})$, 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 $f(\cdot)$,

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

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

$$\gamma^{(i)} = \begin{cases} \frac{1}{2(n+\kappa)} & , i = 1, \ldots, 2n \\ \frac{\kappa}{n+\kappa} & , i = 0 \end{cases} \qquad \xi^{(i)} = \begin{cases} \sqrt{n+\kappa}\mathbf{e}_i & , i = 1, \ldots, n \\ -\sqrt{n+\kappa}\mathbf{e}_{i-n} & , i = n+1, \ldots, 2n \\ 0 & , i = 0, \end{cases} \tag{A15}$$

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., $\mathbf{e}_1 = [1, 0, \ldots, 0]$.

In stochastic cubature approximation (SCA), we adopt the computation of $\xi^{(i)}$ in Equation (A15), 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.

Appendix C. Supplementary to Experiments Setup

Appendix C.1. Stochastic Lorenz Attractor Setup

The Lorenz attractor is govern by ordinary differential equations:

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

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 noise to the transition, which is a mixture of Gaussians $0.5\mathcal{N}(\mathbf{m}_0, \mathbf{P}) + 0.5\mathcal{N}(\mathbf{m}_1, \mathbf{P})$, where

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

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 a training set, 200 sequences as a validation set, and 800 sequences as a test set. For evaluation of Wasserstein distance, we simulate 10 groups of sequences additionally. Each group has 100 sequences with similar initial observations.

Appendix C.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 length of 30. We split it into the training set of size 86,386, the validation set of size 200, and the test set of size 10,000.

Appendix C.3. U.S. Pollution Data Setup

The U.S. pollution dataset consists of four pollutants (NO_2, O_3, SO_2, and CO). Each of them has 3 major values (mean, max value, and air quality index). It is collected from counties in different states every day from 2000 to 2016. Since the daily measurements are very noisy and volatile, we compute the monthly average values of each measurement, and then extract non-overlapping segments of length 24 from the dataset. In total, we extract 1639 sequences as training set, 25 sequences as validation set, and 300 sequences as test set.

Appendix C.4. NBA SportVu Data Setup

The dataset consists of sequences of 2D coordinates that describes the movements of basketball players and the ball. We use a sliding window of the width 30, and the stride 30 to cut the long sequences to short sequences of a fixed length 30. We split them into the training set of size 8324, the validation set of size 489, and the test set of size 980.

Appendix D. Implementation Details

Here, we provide implementation details of VDM models used across the four datasets in the main text. VDM includes:

- Latent RNN: summarize the historic latent states $\mathbf{z}_{<t}$ in the hidden states $\mathbf{h}_t$.
- Transition network: transit the latent states $\mathbf{z}_t$ temporally.
- Emission network: map the latent states $\mathbf{z}_t$ and hidden states $\mathbf{h}_t$ to observations $\mathbf{x}_t$.

- Inference network: update states $\mathbf{z}_t$ given observations $\mathbf{x}_t$ and hidden states $\mathbf{h}_t$.

The optimizer is Adam with the learning rate of 1×10^{-3}. In all experiments, the networks have the same architectures but different sizes. The model size depends on observation dimension $\mathbf{d_x}$, latent state dimension $\mathbf{d_z}$, and hidden state dimension $\mathbf{d_h}$. The number of samples used at each time step in the training is $2\mathbf{d_z} + 1$. If the model output is variance, we use a softplus to ensure its non-negative.

- Latent RNN: one layer GRU of input size $\mathbf{d_z}$ and hidden size $\mathbf{d_h}$
- Transition network: input size is $\mathbf{d_h}$; 3 linear layers of size 64, 64, and $2\mathbf{d_z}$, with ReLUs.
- Emission network: input size is $\mathbf{d_h} + \mathbf{d_z}$; 3 linear layers of size 32, 32 and $2\mathbf{d_x}$, with ReLUs.
- Inference network: input size is $\mathbf{d_h} + \mathbf{d_x}$; 3 linear layers of size 64, 64, and $2\mathbf{d_z}$, with ReLUs.

Here, we give the exact dimension of observations $\mathbf{x}_t$, latent states $\mathbf{z}_t$, and hidden states $\mathbf{h}_t$ of VDM in four experiments in the main text in Table A1. We give the number of parameters for each model in experiments in the main text in Table A2.

Table A1. Dimension details of VDM in four experiments.

	$\mathbf{d_x}$	$\mathbf{d_z}$	$\mathbf{d_h}$
Lorenz	3	6	32
Taxi	2	6	32
Pollution	12	8	48
SportVu	2	6	32

Table A2. Number of parameters for each model in four experiments. VDM, AESMC, DMM-IAF, VRNN, and RKN have comparable number of parameters. CF-VAE has much more parameters.

	RKN	**VRNN**	**CF-VAE**	**DMM-IAF**	**AESMC**	**VDM**
Lorenz	23,170	22,506	7,497,468	24,698	22,218	22,218
Taxi	23,118	22,248	7,491,123	24,536	22,056	22,056
Pollution	35,774	33,192	8,162,850	36,328	31,464	31,464
SportVu	23,118	22,248	7,491,123	24,536	22,056	22,056

All models are trained in our GPU cluster, which consists of NVIDIA GeForce GTX TITAN X GPUs, and NVIDIA TITAN X Pascal GPUs. Since VDM has a small model size, the performance does not rely on the hardware and the training has no high hardware requirements.

References

1. Le, T.A.; Igl, M.; Rainforth, T.; Jin, T.; Wood, F. Auto-Encoding Sequential Monte Carlo. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
2. Krishnan, R.G.; Shalit, U.; Sontag, D. Structured inference networks for nonlinear state space models. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
3. Kingma, D.P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; Welling, M. Improved Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2016; Volume 29.
4. Bhattacharyya, A.; Hanselmann, M.; Fritz, M.; Schiele, B.; Straehle, C.N. Conditional Flow Variational Autoencoders for Structured Sequence Prediction. *arXiv* **2019**, arXiv:1908.09008.
5. Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.C.; Bengio, Y. A recurrent latent variable model for sequential data. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2980–2988.
6. Becker, P.; Pandya, H.; Gebhardt, G.; Zhao, C.; Taylor, J.; Neumann, G. Recurrent Kalman Networks: Factorized Inference in High-Dimensional Deep Feature Spaces. In Proceedings of the Thirty-sixth International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
7. Villani, C. *Optimal Transport: Old and New*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008; Volume 338.

8. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
9. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
10. Fraccaro, M.; Sønderby, S.K.; Paquet, U.; Winther, O. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 2199–2207.
11. Gemici, M.; Hung, C.C.; Santoro, A.; Wayne, G.; Mohamed, S.; Rezende, D.J.; Amos, D.; Lillicrap, T. Generative temporal models with memory. *arXiv* **2017**, arXiv:1702.04649.
12. Li, Y.; Mandt, S. Disentangled sequential autoencoder. In Proceedings of the Thirty-fifth International Conference on Machine Learning, Stockholm Sweden, 10–15 July 2018.
13. Goyal, A.G.A.P.; Sordoni, A.; Côté, M.A.; Ke, N.R.; Bengio, Y. Z-forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6713–6723.
14. Naesseth, C.; Linderman, S.; Ranganath, R.; Blei, D. Variational sequential monte carlo. In Proceedings of the International Conference on Artificial Intelligence and Statistics (PMLR), Banff, AB, Canada, 4–8 July 2004; pp. 968–977.
15. Hirt, M.; Dellaportas, P. Scalable bayesian learning for state space models using variational inference with smc samplers. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Japan, 16–18 April 2019; pp. 76–86.
16. Saeedi, A.; Kulkarni, T.D.; Mansinghka, V.K.; Gershman, S.J. Variational particle approximations. *J. Mach. Learn. Res.* **2017**, *18*, 2328–2356.
17. Schmidt, F.; Hofmann, T. Deep state space models for unconditional word generation. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 6158–6168.
18. Schmidt, F.; Mandt, S.; Hofmann, T. Autoregressive Text Generation Beyond Feedback Loops. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3391–3397.
19. Ziegler, Z.M.; Rush, A.M. Latent normalizing flows for discrete sequences. In Proceedings of the Thirty-Sixth International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
20. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 2172–2180.
21. Li, Y.; Song, J.; Ermon, S. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3812–3822.
22. Bhattacharyya, A.; Schiele, B.; Fritz, M. Accurate and diverse sampling of sequences based on a "best of many" sample objective. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8485–8493.
23. Sadeghian, A.; Kosaraju, V.; Sadeghian, A.; Hirose, N.; Rezatofighi, H.; Savarese, S. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1349–1358.
24. Kosaraju, V.; Sadeghian, A.; Martín-Martín, R.; Reid, I.; Rezatofighi, H.; Savarese, S. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 137–146.
25. Karl, M.; Soelch, M.; Bayer, J.; Van der Smagt, P. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
26. Fraccaro, M.; Kamronn, S.; Paquet, U.; Winther, O. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3601–3610.
27. Rangapuram, S.S.; Seeger, M.W.; Gasthaus, J.; Stella, L.; Wang, Y.; Januschowski, T. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 7785–7794.
28. Zheng, X.; Zaheer, M.; Ahmed, A.; Wang, Y.; Xing, E.P.; Smola, A.J. State space LSTM models with particle MCMC inference. *arXiv* **2017**, arXiv:1711.11179.
29. Doerr, A.; Daniel, C.; Schiegg, M.; Nguyen-Tuong, D.; Schaal, S.; Toussaint, M.; Trimpe, S. Probabilistic recurrent state-space models. In Proceedings of the Thirty-fifth International Conference on Machine Learning, Stockholm Sweden, 10–15 July 2018.
30. De Brouwer, E.; Simm, J.; Arany, A.; Moreau, Y. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 7377–7388.
31. Gedon, D.; Wahlström, N.; Schön, T.B.; Ljung, L. Deep State Space Models for Nonlinear System Identification. *arXiv* **2020**, arXiv:2003.14162.
32. Linderman, S.; Johnson, M.; Miller, A.; Adams, R.; Blei, D.; Paninski, L. Bayesian learning and inference in recurrent switching linear dynamical systems. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 914–922.
33. Nassar, J.; Linderman, S.; Bugallo, M.; Park, I.M. Tree-Structured Recurrent Switching Linear Dynamical Systems for Multi-Scale Modeling. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

34. Becker-Ehmck, P.; Peters, J.; Van Der Smagt, P. Switching Linear Dynamics for Variational Bayes Filtering. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 553–562.
35. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
36. Auger-Méthé, M.; Field, C.; Albertsen, C.M.; Derocher, A.E.; Lewis, M.A.; Jonsen, I.D.; Flemming, J.M. State-space models' dirty little secrets: even simple linear Gaussian models can have estimation problems. *Sci. Rep.* **2016**, *6*, 26677. [CrossRef] [PubMed]
37. Maaløe, L.; Sønderby, C.K.; Sønderby, S.K.; Winther, O. Auxiliary deep generative models. In Proceedings of the International conference on Machine Learning (PMLR), New York, NY, USA, 20–22 June 2016; pp. 1445–1453.
38. Ranganath, R.; Tran, D.; Blei, D. Hierarchical variational models. In Proceedings of the International Conference on Machine Learning (PMLR), New York, NY, USA, 20–22 June 2016; pp. 324–333.
39. Sobolev, A.; Vetrov, D. Importance weighted hierarchical variational inference. *arXiv* **2019**, arXiv:1905.03290.
40. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
41. Owen, A.B. *Monte Carlo Theory, Methods and Examples*; 2013. Available online: https://artowen.su.domains/mc/ (accessed on 30 September 2021).
42. Schön, T.B.; Lindsten, F.; Dahlin, J.; Wågberg, J.; Naesseth, C.A.; Svensson, A.; Dai, L. Sequential Monte Carlo methods for system identification. *IFAC-Pap.* **2015**, *48*, 775–786.
43. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
44. Wu, Y.; Hu, D.; Wu, M.; Hu, X. A numerical-integration perspective on Gaussian filters. *IEEE Trans. Signal Process.* **2006**, *54*, 2910–2921. [CrossRef]
45. Arasaratnam, I.; Haykin, S. Cubature kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [CrossRef]
46. Lee, N.; Choi, W.; Vernaza, P.; Choy, C.B.; Torr, P.H.; Chandraker, M. Desire: Distant future prediction in dynamic scenes with interacting agents. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 336–345.
47. Tomczak, J.M.; Welling, M. VAE with a VampPrior. In Proceedings of the 21st International Conference on Artificial Intelligence and Statistics, Playa Blanca, Spain, 9–11 April 2018.