Memory Augmented State Space Model for Time Series Forecasting

Yinbo Sun, **Lintao Ma**, **Yu Liu**, **Shijun Wang**, **James Zhang**, **YangFei Zheng**, **Hu Yun**, **Lei Lei**, **Yulin Kang** and **Linbao Ye**
Ant Group
{yinbo.syb, lintao.mlt,nuoman.ly,shijun.wang,james.z,yangfei.zfy,huyun.h,jason.ll,yulin.kyl,linbao.ylb}@antgroup.com

Abstract

State space model (SSM) provides a general and flexible forecasting framework for time series. Conventional SSM with fixed-order Markovian assumption often falls short in handling the long-range temporal dependencies and/or highly non-linear correlation in time-series data, which is crucial for accurate forecasting. To this extend, we present *External Memory Augmented State Space Model* (EMSSM) within the sequential Monte Carlo (SMC) framework. Unlike the common fixed-order Markovian SSM, our model features an external memory system, in which we store informative latent state experience, whereby to create "memoryful" latent dynamics modeling complex long-term dependencies. Moreover, conditional normalizing flows are incorporated in our emission model, enabling the adaptation to a broad class of underlying data distributions. We further propose a Monte Carlo Objective that employs an efficient variational proposal distribution, which fuses the filtering and the dynamic prior information, to approximate the posterior state with proper particles. Our results demonstrate the competitiveness of forecasting performance of our proposed model comparing with other state-of-the-art SSMs.

1 Introduction

Time series forecasting is vitally important in real world application, such as demand forecasting, optimizing business processes and financial risk management. One popular probabilistic time series forecasting framework is the state space models (SSM), whose key idea is to create a generative model of the data in terms of latent variables that captures the temporal evolution of the system (i.e., state transitions) , in comparison with many other non-generative models, such as Transformers [Wen *et al.*, 2022], that normally require a lot of training data. The prediction of the observations can be obtained by the state's posterior estimation of the previous time step and the dynamic model of the state. SSMs are flexible for complex data patterns and are scalable to large datasets, while not accumulating errors as autoregressive models for multi-step forecasting.

One main challenge of SSM is its inference method. Specifically, for Linear Guassian SSM, straight-forward algorithms, such as Kalman Filter (KF), can be used to accurately infer the posterior state, but for other non-linear SSM, many early methods suffer from scalability problem, which leads to the development of the following methods to approximate the posterior distribution more efficiently: i) *Variational Inference (VI)*; ii) *Sequential Monte Carlo (SMC)* or *particle filters* using weighted particles iii) *Variational Sequential Monte Carlo (VSMC)*, combining VI and SMC, which is more flexible, accurate and more efficient, allowing increased number of particles.

Another challenge of SSM is the transition dynamics. The canonical SSMs follow the assumption of fixed-order k Markovian state transition, such as KF and other non-linear extensions, which is insufficient to characterize many systems of practical relevance. The failure of fixed-order Markovian state dynamics to learn long correlation or to model complex state dynamics quantitatively [Bialek *et al.*, 2001] leads to the following VI-based SSM works. [Alaa and van der Schaar, 2019] applies attention mechanism into the SSM to model disease progression. [Chung *et al.*, 2015] (variational recurrent neural network, VRNN), [Fraccaro *et al.*, 2016] (stochastic recurrent neural network, SRNN), and [Krishnan *et al.*, 2015] (Deep KFs) solve the non-fixed-order Markovian problems with gating mechanism, such as Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU).

However, these non-fixed-order Markovian models for time series face a common challenge, i.e., the state transition dynamics is based on the historical context with finite length, suffering from the limited storage capacity of memorized history. We therefore propose the combination of memory-based non-fixed-order Markovian models and SMC to relieve this limitation, as SMC can achieve scalable state posterior estimation, fully utilizing the stochastic particles (samples), which to the best of our knowledge, has not been attempted.

To address these problems, we propose *External Memory Augmented State Space Model* (EMSSM), which provides multi-step ahead probabilistic forecast with non-Markovian state transition and does not accumulate prediction errors like autoregressive models [Rasul *et al.*, 2020; Rasul *et al.*, 2021; Salinas *et al.*, 2019]. The prior transition of EMSSM is augmented with external memory network for efficient learning. Our external memory is designed to preserve not only the

context window but all the historical states, while fusing the information with noisy particles. In addition, we incoporate VSMC [Naesseth *et al.*, 2018] as the inference algorithm, and the accurate posterior state with weighted particles sampled by VSMC can be stored in the external memory. Furthermore, resorting to Normalizing Flows (NF) as the emission model, EMSSM can adapt to data of complex distributions and high-dimension.

In summary, the main contributions of this paper are the following:

- A novel external memory augmented SSM architecture equipped with conditional NF to forecast multivariate time series with high-dimensional complex data distribution, yielding competitive performances compared with other SSMs in real-world datasets
- The demonstration that external memory can efficiently utilize the weighted particles sampled by VSMC and can handle the long-term nonlinear temporal dependencies
- A powerful dynamic function better expressing partially predictable observations, contributing to lower KL divergence in comparison to other SSMs

The paper is organized as follows. We first illustrate the problem definition in Sec.2, and then we review the related literature in Sec.3. The generative and inference model are introduced in Sec.4. Experiment and Conclusion are presented in Sec 5, Sec.6, respectively.

2 Problem Definition

The general probabilistic time series forecasting problem is defined as follows. Let $\mathbf{y}_t \in \mathbb{R}^v$ and $\mathbf{x}_t \in \mathbb{R}^k$ denote the time series target, and the time-varying covariate (input) vector, repectively, at time t. SSM models the temporal correlations and long-term dependencies among observation vectors $\mathbf{y}_{1:T} := [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T]$ via unobserved (latent) state representation $\mathbf{z}_{1:T} := [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_T]$, where $\mathbf{z} \in \mathbb{R}^d$. SSM is characterized by transition distribution $\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_t) = f_\theta(\cdot|\mathbf{z}_{<t}, \mathbf{x}_t)$ and emission distribution $\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{x}_t) = g_\theta(\cdot|\mathbf{z}_t, \mathbf{x}_t)$, where θ denotes the parameters of the generative model. The conditional transition density $f_\theta(\cdot|\mathbf{z}_{<t}, \mathbf{x}_t))$ models the temporal dynamics of the system, which defines how the latent $\mathbf{z}_t$ is updated given the input $\mathbf{x}_t$ and the previous states $\mathbf{z}_{<t}$. The noisy observation variable $\mathbf{y}_t$ is generated by the latent state $\mathbf{z}_t$ via the emission density $g_\theta(\cdot|\mathbf{z}_t, \mathbf{x}_t)$, given the current state $\mathbf{z}_t$ and input $\mathbf{x}_t$. Under the assumption of SSM and d-separation properties of the graphical model, the joint distribution $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}|\mathbf{x}_{1:T})$ can be factorized as:

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

3 Related Work

3.1 Inference of SSM

For linear Gaussian systems, Kalman Filter (KF) can be used to solve the inference problem analytically. In contrast, for non-linear SSM, the exact inference becomes intractable and many approximate algorithms have been proposed, which can be summarized as follows:

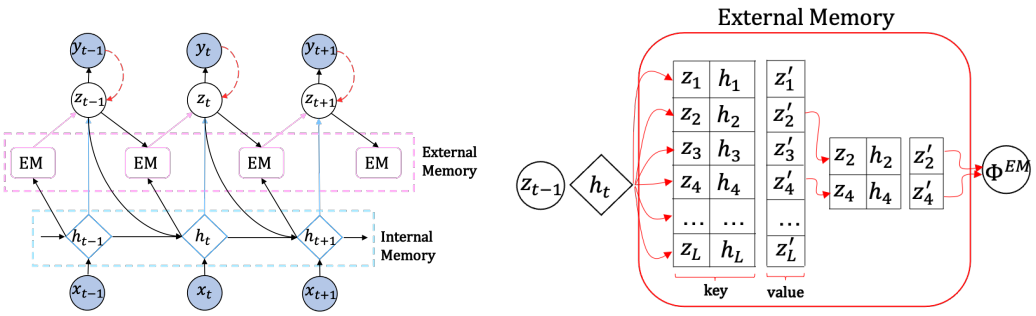
- The deterministic methods such as the extended KF and Variational Inference (VI) [Arasaratnam and Haykin, 2009; Blei *et al.*, 2017]. These methods can quickly approximate the posterior state, but lack accuracy.
- The stochastic methods, such as particle filters or smoothers that approximate the exact posterior distributions using a set of weighted samples (particles) via sequential Monte Carlo [Gordon *et al.*, 1993; Kitagawa, 1996], can get faithful posterior estimation, but suffer from inefficiency.
- The hybrid algorithms, such as stochastic VI [Krishnan *et al.*, 2017; Karl *et al.*, 2016] and variational SMC(VSMC) [Naesseth *et al.*, 2018; Maddison *et al.*, 2017; Le *et al.*, 2017], can balance between efficiency and reliability of the posterior estimation, and here in this work, we adopt VSMC to approximate the posterior distribution using augmented proposal distribution in SMC with variational distribution.

3.2 Memory-based Approach for Complex System

Internal Memory. Constructing the well-defined transition dynamics for the state is vitally important in SSM. Conventional SSMs follow the fixed-order Markovian dynamic assumption, ignoring the long-term complex temporal dependencies. A lot of effort based on the VI framework has been devoted to overcome the flaw of this naive assumption. [Chung *et al.*, 2015](variational recurrent neural network/VRNN), [Fraccaro *et al.*, 2016](stochastic recurrent neural network/SRNN), [Krishnan *et al.*, 2015](Deep KFs) utilize LSTM or GRU to model the state dynamics while preserving the information over long intervals. Such recurrent dynamics, integrating both the computation and the memory storage, can be regarded as *internal memory*, which has two major drawbacks, i.e., poor scalability and restricted capability to distill information over the context of limited length.

External Memory. External memory-augmented neural network (MANN) has been used to increase the network capacity and to better capture long-term structure within sequences. It was originally proposed in [Weston *et al.*, 2015] and expanded in later works [Sukhbaatar *et al.*, 2015]. The combination of memory buffer and differentiable addressing mechanism has been developed to alleviate the memory capacity constraint in several tasks such as natural language processing [Kumar *et al.*, 2016; Sukhbaatar *et al.*, 2015] and algorithm learning [Bošnjak *et al.*, 2017].

Motivated by the success of external memory in various areas, we incorporate the same mechanism into SSM to model the transition dynamics of time series, to address the limitation of internal memory. External memory here stores the accurate posterior state with weighted particles sampled by VSMC, which facilitates the main idea of our framework, i.e., similar target evolution over time corresponds to similar hidden states and the same target value will appear again with high probability. We will demonstrate later that the search and retrieval of similar posterior state preserved in external memory improves the forecasting performance.



(a) EMSSM (b) External Memory

Figure 1. (a) Graphical structure of the proposed EMSSM with the generative and inference processes. EM denotes the external memory and $\mathbf{h}_t$ denotes the hidden state of RNN. The pink solid line and sky blue solid line indicate the generative process of encoded vectors Φ^{EM} and Φ^{IM}, respectively, and the red dashed line indicates the inference process. (b) The external memory structure corresponds to EM in (a). "key" and "value" represent M_k and M_v in eq.2, respectively. Here we retrieve two nearest neighbours to $[\mathbf{z}_{t-1}, \mathbf{h}_t]$ and generate the context Φ^{EM}

.

4 Memory-Augmented State Space Model

In this section, we describe the proposed *External Memory Augmented State Space Model* (EMSSM), whose overall structure is illustrated in Figure 1. Unlike VRNN or SRNN, we add the additional dependency on the external memory when generating $\mathbf{z}_t$, and the whole memory architecture consists of *internal memory* and *external memory* listed below:

- **Internal Memory** corresponds to GRU network, distilling the information from local context window.
- **External Memory** memorizes the historical state transitions, which integrate information of the past states, covariates and targets, recording both the global temporal pattern (which cannot be distilled from the local context window) and the stochastic part of $p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_t)$ in the past context. Figure 1(b) shows the schematics of *External Memory*.

4.1 Internal Memory

Internal memory is constructed with GRU network, which recursively preserves information into a vector with fixed size for later retrieval. Given the previous latent state vector $\mathbf{z}_{t-1} \in \mathbb{R}^d$, the previous hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^l$, the covariate input $\mathbf{x}_t$ and the multi-layer recurrent GRU network f_{GRU} with parameter θ, at every time step, GRU updates the current hidden state $\mathbf{h}_t$ and outputs the encoded dense vector Φ^{IM}, which fuses the information from the local context:

$$\Phi^{\text{IM}}, \mathbf{h}_t = f_{\text{GRU}}([\mathbf{z}_{t-1}, \mathbf{x}_t], \mathbf{h}_{t-1}; \theta). \tag{1}$$

Above RNN encoded context information is the major component proposed in VRNN and SRNN, with a major drawback that no global information is shared among different time windows. Especially for the historical latent states further away from the current window. For example, in the demand-forecasting scenario, the latent state of the same holiday last year may be an important reference of the demand estimation for the current year. To overcome the limitation of internal memory, we propose to use the additional *external memory*.

4.2 External Memory

Unlike the internal memory, the external memory with size L is a refreshed buffer, storing the historical posterior state transitions. The external memory distills the stochastic information of $p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_t)$ depending on noisy particles of z_t which is conditioned on $\mathbf{z}_{t-1}$ and $\mathbf{h}_t$. The external memory consists of both key and value memories:

$$\text{M}_k = \begin{bmatrix} \mathbf{z}_1, \mathbf{h}_1 \\ \cdots \\ \mathbf{z}_L, \mathbf{h}_L \end{bmatrix} \in \mathbb{R}^{L\times(d+l)}, \quad \text{M}_v = \begin{bmatrix} \mathbf{z}'_1 \\ \cdots \\ \mathbf{z}'_L \end{bmatrix} \in \mathbb{R}^{L\times d}, \tag{2}$$

where $\mathbf{z}_i \in \mathbb{R}^d$ represents the previous state, $\mathbf{h}_i$ is the GRU state, and $\mathbf{z}'_i \in \mathbb{R}^d$ represents the transitioned state.

In SSM, KL divergence $\text{KL}(q(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t)|p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x}_t))$ quantifies the additional information besides the prior knowledge over the latent state when predicting the current observation. KL divergence being zero means that the current observation can be well predicted by the prior information defined by the transition distribution alone. But the prior knowledge or the prior construction function generally does not have all the information, which contributes to the discrepancy between the real target and the prediction, e.g., forecasting the sales amount during promotions. To bridge the gap between the prior and the posterior, we employ external memory to record the historical state transitions, i.e., the non-parameterized distribution mapping $\mathcal{M} : \overline{\mathbf{z}}_{t-1} \longmapsto \overline{\mathbf{z}}_t$ between the previous ($\overline{\mathbf{z}}_{t-1}$) and the post latent states ($\overline{\mathbf{z}}_t$). $\overline{\mathbf{z}}_t$ is obtained as the weighted average of all P particles from the sequential Monte Carlo (SMC) algorithm:

$$\overline{\mathbf{z}}_t = \sum_{p=1}^{P} \omega_p \mathbf{z}_t^p \tag{3}$$

As a result of equation 3, we do not keep all the P particles $\{\mathbf{z}_t^p\}_{p=1}^P$ for computation efficiency and space saving purposes.

External Memory Controller. When training random sequences in every batch, the weighted latent state $\overline{\mathbf{z}}_t$ (equation 3) and hidden GRU state $\overline{\mathbf{h}}_t = \sum_{p=1}^P \omega_p \mathbf{h}_t^p$ form the

state transition pair $\{[\overline{\mathbf{z}}_{t-1}, \overline{\mathbf{h}}_t], \overline{\mathbf{z}}_t\}$ that is written into the memory. To eliminate the information retrieval redundancy, we perform the state vector neighborhood subset generation

$$\mathrm{M}_v^{\text{sub}} := \{[\mathbf{z}_j, \mathbf{h}_j]\}_{1 \le j \le L} \subseteq \{[\mathbf{z}_i, \mathbf{h}_i] \in \mathbb{R}^{d+l}\}_{i=1}^{L}$$

via the function $f_n(\cdot)$ such as the kd-tree, cosine similarity, Euclidean distance, etc. Then to retrieve the memory vector, we use the function $f_w(\cdot)$ to yield soft attention weights or the average weights w_t.

Lastly, the retrieved external memory context Φ^{EM} is a weighted sum of the row vectors of $\mathrm{M}_v^{\text{sub}}$ with weights w_t:

$$\begin{aligned} \mathrm{M}_v^{\text{sub}} &= f_n(\mathbf{z}_{t-1}, \mathbf{h}_t, \mathrm{M}_v), \\ w_t &= f_w(\mathbf{z}_{t-1}, \mathbf{h}_t, \mathrm{M}_v^{\text{sub}}), \\ \Phi^{\text{EM}} &= w_t \mathrm{M}_v^{\text{sub}}. \end{aligned} \tag{4}$$

The final memory output context Φ_t is the concatenation of the GRU output Φ^{IM} obtained in equation 1 and the external memory output Φ^{EM} obtained in equation 4:

$$\Phi_t = [\Phi^{\text{IM}}, \Phi^{\text{EM}}], \tag{5}$$

which is then fed into the generative model in equation 6 below.

4.3 Generative Model

In the state space model, firstly we focus on the joint distribution of the latent states and observations which consists of the transition and emission distribution. The overall joint distribution in our EMSSM can be modeled via the following factorization:

$$\begin{aligned} p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T} | \mathbf{x}_{1:T}) &= p(\mathbf{y}_{1:T} | \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T} | \Phi_{1:T}(\mathbf{x}_{1:T})) \\ &= \prod_{t=1}^{T} \underbrace{p(\mathbf{y}_t | \mathbf{z}_t)}_{\text{Emission}} \underbrace{p(\mathbf{z}_t | \Phi_t(\mathbf{z}_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_t))}_{\text{Transition}}, \end{aligned} \tag{6}$$

where $\Phi_{1:T}$ is the output of the external memory system defined in equation 5.

Transition Distribution. The transition distribution describes how the dynamic states $\mathbf{z}_t$ evolve with the prior knowledge $\mathbf{x}_t$, and$\mathbf{z}_{<t}$. In the classic KF, the state $\mathbf{z}_t$ has the linear map with covariate $\mathbf{x}_t$ and the previous state $\mathbf{z}_{t-1}$. We adopt the external memory to model the state evolution to better capture the long-term dependency:

$$\mathbf{z}_t \sim f_\theta(\cdot | \Phi_t(\mathbf{z}_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_t)), \tag{7}$$

where $\mathbf{z}_{t-1}$ is the stochastic latent state, $\mathbf{h}_{t-1}$ is the deterministic hidden state of GRU and $\mathbf{x}_t$ is the covariate. The transition density function $f_\theta(\cdot | \Phi_t(\mathbf{z}_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_t))$ is specified as a diagonal Gaussian distribution with mean and variance parameterized by the output of a multilayer perceptron (MLP) network by the memory context Φ_t using the reparameterization trick.

Emission Distribution. The emission model describes the time series data generation process. In order to adapt to high dimensional complex data distribution, we combine the multilayer perceptron transformation and conditional normalizing flow (CNF). The emission distribution is modeled by the following formula:

$$\mathbf{y}_t \sim g_\theta(\cdot | \mathcal{N}(0, \mathrm{I}), h_\theta(\mathbf{z}_t)), \tag{8}$$

where the emission density function $g_\theta(\cdot|\cdot)$ is a CNF transforming the base distribution $\mathcal{N}(0, \mathrm{I})$ to a complex target distribution. For NF, the base distribution $p(\mathcal{X})$ can be mapped to the data distribution $p(\mathcal{Y})$ via multiple coupling layers: $\mathcal{X} \mapsto \mathcal{C}^{(0)} \mapsto \mathcal{C}^{(1)} \mapsto \cdots \mapsto \mathcal{C}^{(n)} \mapsto \mathcal{Y}$. The relation between the probability density (PDF) $p(\mathcal{X})$ and $p(\mathcal{Y})$ can be expressed by the change of variable formula: $p(\mathcal{X}) = p(\mathcal{Y}) \left| \det \left(\frac{\partial \mathcal{Y}}{\partial \mathcal{X}} \right) \right|$. We build our model block by adopting the non-volume preserving flow, such as RealNVP [Dinh *et al.*, 2016] whose Jacobian determinant is a lower triangular matrix that can be efficiently computed. $p(\mathcal{X})$ is chosen as the standard normal distribution $\mathcal{X} \sim \mathcal{N}(0, \mathrm{I})$ and the coupling layer output $\mathcal{C}^{(i)}$ is conditioned on the latent state $\mathbf{z}_t$, which means the mapping varies with time via the CNF mechanism. The nonlinear transformation function $h_\theta(\mathbf{z}_t)$ is implemented by multilayer perceptron (MLP) network with input $\mathbf{z}_t$.

4.4 Inference and Parameter Learning

Sequential Monte Carlo. Given the observations $\mathbf{y}_{1:T}$ and covariates $\mathbf{x}_{1:T}$, we want to infer the posterior states $\mathbf{z}_{1:T}$ and learn the parameter θ of the generative model for our dynamic SSM. To learn θ, one can maximize the marginal probabilistic density $p(\mathbf{y}_{1:T})$ by integrating out $\mathbf{z}_{1:T}$:

$$p(\mathbf{y}_{1:T}) = \int p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) d\mathbf{z}_{1:T}.$$

However, the above integral cannot be carried out analytically because the generative and emission models are implemented with a neural network, for which we employ the technique of VI [Blei *et al.*, 2017] to convert maximizing the marginal probabilistic density into maximizing the evidence lower bound (ELBO), using the concavity of log function and Jensen's inequality.

$$\begin{aligned} \mathcal{F}(\theta, \phi) &= \mathbb{E}_{q_\phi(\mathbf{z}_{1:T})}[\log p_\theta(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) - \log q_\phi(\mathbf{z}_{1:T})] \\ &\le \log \mathbb{E}_{q_\phi(\mathbf{z}_{1:T})} \frac{p_\theta(\mathbf{y}_{1:T}, \mathbf{z}_{1:T})}{q_\phi(\mathbf{z}_{1:T})} = \log p(\mathbf{y}_{1:T}), \end{aligned}$$

where $q_\phi(\mathbf{z}_{1:T})$ is taken from a family of distributions with parameter ϕ. Following the recent development of the Bayesian inference, we perform the variational sequential Monte Carlo (VSMC) algorithm with P particles [Naesseth *et al.*, 2018], which uses the variational distribution as the proposal distribution. The algorithm gives an unbiased estimate of the marginal likelihood $\hat{p}_\theta(\mathbf{y}_{1:T}) = \prod_{t=1}^{T} [1/P \sum_{p=1}^{P} \omega_t^{(p)}]$, where $\omega_t^{(p)}$ is the unnormalized importance weight of particle p at time t and maximize the surrogate ELBO

$$\tilde{\mathcal{F}}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}_{1:T})} \log \hat{p}_\theta(\mathbf{y}_{1:T}).$$

After training the objective, the proposal distribution for the inference is learned simultaneously. The biased gradient of surrogate ELBO $\nabla \tilde{\mathcal{F}}(\theta, \phi) \approx \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,1)}[\nabla \log \hat{p}_{\theta,\phi}(\mathbf{y}_{1:T})]$, omitting the score gradient term, is applied to maximize $\tilde{\mathcal{F}}(\theta, \phi)$, which leads to faster convergence as the prior and the posterior distributions are reparameterized with the parameters θ and ϕ.

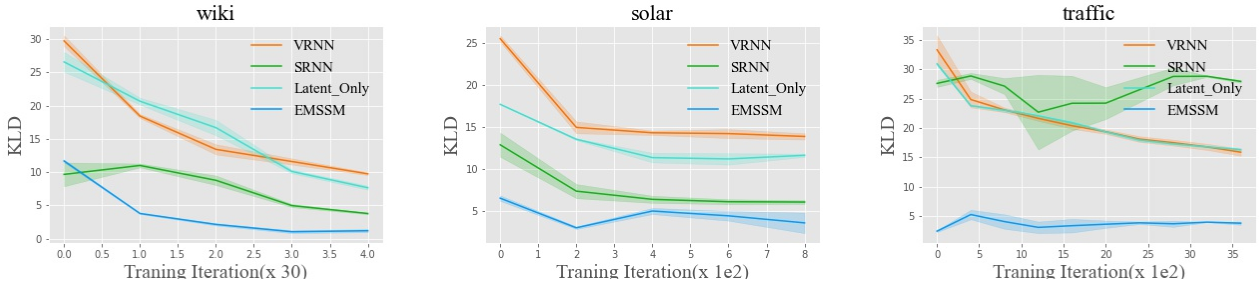


Figure 2. The average Kullback-Leibler divergence (KLD) (lower is better) for our model and other SSMs (VRNN, SRNN and Latent_only) on three real world datasets.

Proposal Distribution. A good estimate with low variance using the SMC algorithm relies on the design of the proposal distribution. Our proposal distribution utilizes both the prior knowledge Φ of of the predictive prior dynamics in the generative model and the observation $\mathbf{y}_t$. The proposal distribution uses GRU as the base feature extraction network to learn the filter or smoothing posterior collecting the information in all time steps. The encoded feature $\mathbf{a}_t \in \mathbb{R}^p$ is generated by passing the target value $\mathbf{y}_t$ and covariate $\mathbf{x}_t$ to the GRU network. The final proposal context Ψ is the concatenation of memory output context Φ and encoded feature $\mathbf{a}_t$:

$$\tilde{\mathbf{z}}_t \sim f_\phi(\cdot|\Psi), \quad \Psi = [\Phi, \mathbf{a}_t],$$
$$\mathbf{z}_t = \tilde{\mathbf{z}}_t + \overline{\mathbf{z}}_{\text{prior,t}}.$$

The inference density function $f_\phi(\cdot|\Psi)$ is also specified as a diagonal Gaussian distribution whose mean and variance are parameterized by the output of a multilayer perceptron (MLP) and inference memory context Ψ using the reparameterization trick. Here, motivated by the work of [Fraccaro *et al.*, 2016], rather than directly learning $\mathbf{z}_t$, we learn the residual term $\tilde{\mathbf{z}}_t$ between the true posterior $\mathbf{z}_t$ and prior mean $\overline{\mathbf{z}}_{\text{prior,t}}$, i.e., $\tilde{\mathbf{z}}_t = \mathbf{z}_t - \overline{\mathbf{z}}_{\text{prior,t}}$. Equivalently, the final posterior $\mathbf{z}_t = \tilde{\mathbf{z}}_t + \overline{\mathbf{z}}_{\text{prior,t}}$.

5 Experiment Results

5.1 Evaluation Metric and Data Set

We evaluate our model for multivariate time series forecasting on five popular public datasets including electricity, solar, traffic, exchange, and wikipedia (see [Rasul *et al.*, 2020] for the properties and the summary of the datasets). These public datasets exhibit different seasonal patterns with hourly and daily frequencies. For the evaluation of different models, each dataset prior to the fixed forecast date is used as training data and the remaining data as the prediction data. The rolling window prediction starting from the fixed forecasting date is adopted as model evaluation method.

The continuous ranked probability score (CRPS) is a well-known probabilistic model metric that measures the compatibility of model's cumulative distribution function (CDF) F of the observation y, defined as

$$\text{CRPS}(F, x) = \int_{\mathbb{R}} (F(u) - \mathbb{I}_{y \le u})^2 du, \tag{9}$$

where $\mathbb{I}_{y \le u}$ is the indicator function which is one if $y \le u$ and zero otherwise. Lower CRPS scoring means better performance of the model's forecasting ability. The model's CDF can be approximated by the empirical CDF with n samples $\hat{y}_i$: $\hat{F}(u) = 1/n \sum_{i=1}^{n} \mathbb{I}\{\hat{y}_i \le u\}$. However, the CRPS metric is defined for univariate time series, and it cannot capture the dependencies across different time series. Therefore, we use CRPS_{sum} as the metric for multivariate time series forecasting, which can be obtained by first summing across different time series yielding $\text{CRPS}_{\text{sum}} = \mathbb{E}_t[\text{CRPS}(\hat{F}_{\text{sum}}(t), \sum_i x_t^i)]$. The CRPS_{sum} metric has been theoretically and experimentally justified in the previous work.

5.2 Results

We compare our model with other SSMs (KAVE, VRNN, SRNN and Latent only model) and other multivariate time series forecasting model (GARCH, DeepVAR and GRU RealNVP). We also incorporated the same NF (RealNVP) to the original VRNN and SRNN models for a fair comparison and ablation study. The results are shown in Table 1 where the mean and standard errors are obtained by three independent runs.

- **GARCH** [Bauwens *et al.*, 2006; Lütkepohl, 2005] is a multivariate generalized autoregressive conditional heteroskedasticity model.
- **DeepVAR lowrank-copula** [Salinas *et al.*, 2019] models the dynamics of the time series via the RNN and constructs the real data distribution via low-rank Gaussian copula.
- **KVAE** [Fraccaro *et al.*, 2017] is a variational autoencoder structure combined with the linear Gaussian state space model to model the dynamics of the multivariate time series data.
- **GRU RealNVP** [Rasul *et al.*, 2020] unrolls the GRU network describing the dynamics and model the complex data distribution via CNF.
- **VRNN RealNVP** uses the VRNN [Chung *et al.*, 2015] to model the data dynamics and augment the emission distribution with CNF.
- **SRNN RealNVP** utilizes the SRNN [Fraccaro *et al.*, 2016] to model the data dynamics and augment the emission distribution with CNF.

Method	exchange	solar	electricity	traffic	wiki
GARCH	0.023 ± 0.000	0.88 ± 0.002	0.19 ± 0.001	0.37 ± 0.0016	$-$
KAVE	0.014 ± 0.002	0.34 ± 0.025	0.051 ± 0.019	0.1 ± 0.005	0.095 ± 0.012
DeepVAR lowrank-Copula	0.007 ± 0.000	$\mathbf{0.319 \pm 0.011}$	0.064 ± 0.008	0.103 ± 0.006	0.241 ± 0.033
GRU RealNVP	0.0064 ± 0.001	0.331 ± 0.02	0.024 ± 0.001	0.078 ± 0.001	0.078 ± 0.001
VRNN RealNVP	0.0072 ± 0.0004	0.416 ± 0.011	0.027 ± 0.003	0.062 ± 0.002	0.070 ± 0.002
SRNN RealNVP	0.0071 ± 0.0002	0.406 ± 0.006	0.037 ± 0.002	0.082 ± 0.006	0.064 ± 0.001
Latent_only RealNVP	0.0074 ± 0.0005	0.412 ± 0.008	0.034 ± 0.003	0.105 ± 0.002	0.065 ± 0.001
EMSSM RealNVP(ours)	$\mathbf{0.0059 \pm 0.0003}$	0.360 ± 0.013	$\mathbf{0.023 \pm 0.001}$	$\mathbf{0.060 \pm 0.002}$	$\mathbf{0.061 \pm 0.001}$

Table 1. Test set CRPS_{sum} (lower is better) of eight models on five real world data sets. The best result for each dataset is in bold. Mean and standard errors are obtained by re-running each model three times.

- **Latent_only RealNVP** utilizes the latent_only structure [Gemici *et al.*, 2017] to model the data dynamics and augment the emission distribution by CNF. This structure can be seen as the variant of VRNN where the emission model only uses the latent variable without utilizing the RNN output. This model is evaluated for ablation purpose.

In Table 1, we observe that EMSSM with RealNVP achieves the state-of-the-art CRPS_{sum} metric on 4 of the 5 datasets. In comparison against VRNN and SRNN, EMSSM demonstrates competitive forecasting capability, indicating the effectiveness of the external memory mechanism, which can be further verified by the ablation comparison study. Specifically, two model instances are compared, i.e., EMSSM with RealNVP against Latent_only RealNVP that can be seen as EMSSM without the memory block. A lower CRPS_{sum} from EMSSM demonstrates the advantage of using the memory. In addition, the competitive forecasting performance in the datasets of different strengths of seasonality further implies that our model can capture data pattern beyond seasonality. We omit further discussion due to space limitation.

KL divergences per time-step over $\mathbf{z}_t$ measure the additional information between the prior distribution and posterior distribution which is used to express the current observation. Lower KL divergences indicate that the prior function can explain most part of the current observation and the target is thus more predictable. Figure 2 shows the KL divergence of training sequence in the real-world datasets mentioned above, where we can see that using external memory yields almost the fastest drop and the lowest level of the KL divergence compared with other deep SSMs. This implies that the historical state information preserved in the external memory actually reduces the discrepancy between the prior and posterior distributions.

The time and space complexities of our EMSSM model are $\mathcal{O}(d^2 + dl + dL)PT)$ and $\mathcal{O}((d+l)(PT+L))$, respectively, whereas the time and space complexities of VRNN/SRNN model are $\mathcal{O}((d^2 + dl)PT)$ and $\mathcal{O}((d+l)PT)$, respectively, where we denote the time step as T, the number of particles as P, the dimension of hidden state $\mathbf{h}_t$ of GRU as l, the dimension of latent state as d, and the buffer size of the external memory as L. Due to the additional external memory, our model has higher time and space complexities than VRNN/SRNN, but our model demonstrates faster convergence and smaller prediction error as shown in Figure 2.

6 Conclusion

In this paper, we proposed a memory-augmented state space model with CNF for multivariate time series forecasting, providing multi-step ahead probabilistic forecast with non-Markovian state transition. The memory consists of both internal memory and external memory buffers, designed to memorize the global temporal pattern and the posterior state transitions, respectively. Our mechanism demonstrated lower KL divergence and better forecasting performance compared with other SSM models. The combination of CNF and the memory mechanism under the SSM framework enhanced the forecasting ability for multivariate time series in our experiments and we believe that our proposed method can better solve difficult real-world time series forecasting problems, such as holiday sales forecasting and anomaly detection, etc. For the future work, a more efficient storage and sampling mechanism for the memory buffer (e.g., using the information entropy) might further improve the training efficiency and prediction performance, which we will leave for further investigation.

References

[Alaa and van der Schaar, 2019] Ahmed M Alaa and Mihaela van der Schaar. Attentive state-space modeling of

disease progression. *Advances in Neural Information Processing Systems*, 32:11338–11348, 2019.

[Arasaratnam and Haykin, 2009] Ienkaran Arasaratnam and Simon Haykin. Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6):1254–1269, 2009.

[Bauwens *et al.*, 2006] Luc Bauwens, Sébastien Laurent, and Jeroen VK Rombouts. Multivariate garch models: a survey. *Journal of applied econometrics*, 21(1):79–109, 2006.

[Bialek *et al.*, 2001] William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural computation*, 13(11):2409–2463, 2001.

[Blei *et al.*, 2017] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[Bošnjak *et al.*, 2017] Matko Bošnjak, Tim Rocktäschel, Jason Naradowsky, and Sebastian Riedel. Programming with a differentiable forth interpreter. In *International conference on machine learning*, pages 547–556. PMLR, 2017.

[Chung *et al.*, 2015] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28:2980–2988, 2015.

[Dinh *et al.*, 2016] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[Fraccaro *et al.*, 2016] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. *arXiv preprint arXiv:1605.07571*, 2016.

[Fraccaro *et al.*, 2017] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in Neural Information Processing Systems*, pages 3604—-3613, 2017.

[Gemici *et al.*, 2017] Mevlana Gemici, Chia-Chun Hung, Adam Santoro, Greg Wayne, Shakir Mohamed, Danilo J Rezende, David Amos, and Timothy Lillicrap. Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*, 2017.

[Gordon *et al.*, 1993] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F-radar and signal processing*, volume 140, pages 107–113. IET, 1993.

[Karl *et al.*, 2016] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick Van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

[Kitagawa, 1996] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[Krishnan *et al.*, 2015] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[Krishnan *et al.*, 2017] Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[Kumar *et al.*, 2016] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR, 2016.

[Le *et al.*, 2017] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. *arXiv preprint arXiv:1705.10306*, 2017.

[Lütkepohl, 2005] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[Maddison *et al.*, 2017] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. *Advances in Neural Information Processing Systems*, 30, 2017.

[Naesseth *et al.*, 2018] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International conference on artificial intelligence and statistics*, pages 968–977. PMLR, 2018.

[Rasul *et al.*, 2020] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. *arXiv preprint arXiv:2002.06103*, 2020.

[Rasul *et al.*, 2021] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. *arXiv preprint arXiv:2101.12072*, 2021.

[Salinas *et al.*, 2019] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *arXiv preprint arXiv:1910.03002*, 2019.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015.

[Wen *et al.*, 2022] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

[Weston *et al.*, 2015] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *International Conference on Learning Representations (ICLR)*, 2015.