Generative Time Series Forecasting with Diffusion, Denoise and Disentanglement

Anonymous Author(s)
Affiliation
Address
email

Abstract

Time series forecasting has been a widely explored task that is of great importance in many applications. However, it is common that real-world time series data are recorded in a short time period, which results in a big gap between the deep model and the limited and noisy time series. In this work, we propose to address the time series forecasting problem with generative modeling. and propose a bidirectional variational auto-encoder (BVAE) equipped with diffusion, denoise and disentanglement, namely D^3VAE. Specifically, a coupled diffusion probabilistic model is proposed to augment the time series data without increasing the aleatoric uncertainty contributed to the data. To ensure the generated series move towards the true target, we further propose to adapt and integrate the multiscale denoising score matching into the diffusion process for time series forecasting. In addition, to enhance the interpretability and stability of the prediction, we treat the latent variable in a multivariate manner and disentangle them on top of minimizing total correlation. Extensive experiments on both synthetic data and real-world data show that D^3VAE outperforms competitive algorithms with remarkable margins.

1 Introduction

Time series forecasting is of great importance for risk averse and decision making. Traditional RNN-based methods capture temporal dependencies of the time series to predict the future. LSTMs and GRUs [52, 16, 15, 41] introduce the gate functions into cell structure to handle long-term dependencies effectively. CNN-based models capture the complex inner patterns of the time series through the convolutional operations [28, 4, 3]. Recently, the former-based models have shown great performance in time series forecasting [51, 53, 25, 29] with the help of multi-head self-attention. However, one big issue of the neural networks in time series forecasting is the uncertainty [14, 1] resulting from the properties of the deep structure. The models based on vector autoregression (VAR) [5, 10, 23] try to model the distribution of the time series from the hidden states which could provide more reliability to the prediction. Yet the performance of VAR is not satisfactory [27].

Interpretable representation learning is another merit for time series forecasting. Variational Auto-Encoders (VAEs) have shown not only the superiority in modeling latent distributions of the data and reducing the gradient noise [37, 24, 30, 45] but also the interpretability on time series forecasting [11, 12]. However, the interpretability of VAEs might be inferior due to the entangled latent variables. There have been efforts to learn representation disentangling [22, 2, 18], which show that the well disentangled representation can improve the performance and robustness of the algorithm.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Moreover, real-world time series are often noisy and recorded in a short time period, which may result in overfitting and poor generalization issues [13, 42, 54, 49][1]. To this end, we address the time series forecasting problem with generative modeling. Specifically, we propose a bidirectional variational auto-encoder (BVAE) equipped with diffusion, denoise and disentanglement, namely D^{3}VAE. More specifically, we first adapt the Nouveau VAE [45] to the time series forecasting task and develop a BVAE as the backbone. Then, we propose a coupled diffusion probabilistic model to remedy the limitation of time series data by augmenting the input time series, as well as the output time series, inspired by the diffusion models [19, 35, 36]. Though the merit of generalizability is helpful, the diffused samples might be corrupted, which results in a generative model moving towards the noisy target. Therefore, we further develop a scaled denoising score matching network for cleaning diffused target time series. In addition, we disentangle the latent variables of the time series by assuming that different disentangled dimensions of the latent variables correspond to different temporal patterns (such as trend, seasonal and so on). Our contributions can be summarized as:

- We propose a coupled diffusion probabilistic model aiming to reduce the aleatoric uncertainty of the time series and improve the generalization capability of the generative model.
- We integrate the multiscale denoising score matching into the coupled diffusion process to improve the accuracy of the generated results.
- We disentangle the latent variables of the generative model to improve the interpretability for time series forecasting.
- Extensive experiments on both synthetic datasets and real-world datasets demonstrate that D^{3}VAE outperforms competitive baselines with satisfactory margins.

2 Methodology

2.1 Generative Time Series Forecasting

Problem Formulation. Given an input multivariate time series $X = \{x_1, x_2, ..., x_n \,|\, x_i \in \mathbb{R}^d\}$ and the corresponding target time series $Y = \{y_{n+1}, y_{n+2}, ..., y_{n+m} \,|\, y_j \in \mathbb{R}^{d'}\}$ ($d' \leq d$). We assume that Y can be generated from latent variables $Z \in \Omega_Z$ that can be drawn from the Gaussian distribution $Z \sim p(Z|X)$. The latent distribution can be further formulated as $p_\phi(Z|X) = g_\phi(X)$ where g_ϕ denotes a nonlinear function. Then, the data-density of the target series is given by:

$$p_\theta(Y) = \int_{\Omega_Z} p_\phi(Z|X)(Y - f_\theta(Z))dZ\,, \tag{1}$$

where f_θ denotes a parameterized function. The target time series can be obtained directly by sampling from $p_\theta(Y)$.

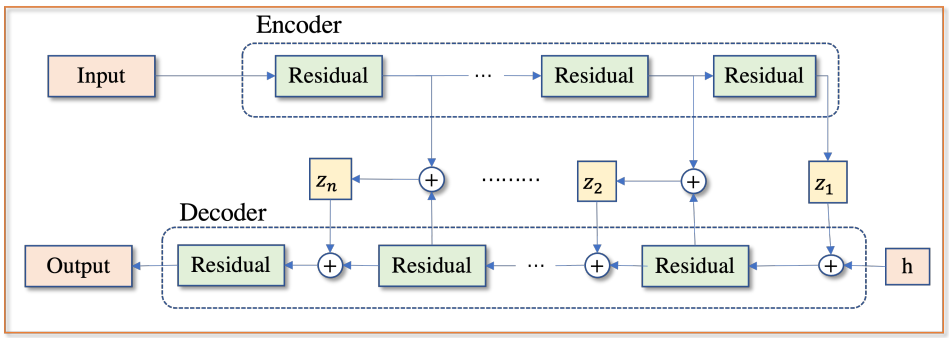


Figure 1: The architecture of Bidirectional Variational Auto-Encoder.

Bidirectional Variational Auto-Encoder. In this work, we employ the bidirectional variational auto-encoder (BVAE) [45] as the backbone generative model which is described in Fig. 1, where Z is

[1]Detailed literature review can be found in Appendix A in the supplementary materials.

treated in a multivariate fashion $Z = [z_1, \cdots, z_n]$ ($z_i \in \mathbb{R}^m$) and $z_{i+1} \sim p(z_{i+1}|z_i, X)$. Then, n is determined in accordance with the number of residual blocks in the encoder, as well as the decoder.

In our problem settings, time series forecasting is to learn the representation Z that captures useful signals of X, and map the low dimensional X to the high dimensional latent space which is suitable for factor disentangling. However, many real-world time series data are recorded in a short time period with inevitable noises, which may hinder the neural networks being applied directly. Hence, large amount of data are needed to ensure the generalization of neural network (see Appendix G). Before diving into the detailed techniques, we first introduce a preliminary proposition.

Proposition 1. *Given a time series X and its inherent noise ϵ_X, we have the decomposition: $X = \langle X_r, \epsilon_X \rangle$, where X_r is the ideal time series data without noise. X_r and ϵ_X are independent from each other. Let $p_\phi(Z|X) = p_\phi(Z|X_r, \epsilon_X)$, the estimated target series $\widehat{Y}$ can be generated with the distribution $p_\theta(\widehat{Y}|Z) = p_\theta(\widehat{Y}_r|Z) \cdot p_\theta(\epsilon_{\widehat{Y}}|Z)$ where $\widehat{Y}_r$ is the ideal part of $\widehat{Y}$ and $\epsilon_{\widehat{Y}}$ is the estimation noise. Without loss of generality, $\widehat{Y}_r$ can be fully captured by the model. That is, $\|Y_r - \widehat{Y}_r\| \longrightarrow 0$ where Y_r is the ideal part of the ground truth target series Y. In addition, Y can be decomposed as $Y = \langle \widehat{Y}_r, \epsilon_Y \rangle$ (ϵ_Y denotes the noise of Y). Therefore, the error between the ground truth and the prediction, i.e. $\|Y - \widehat{Y}\| = \|\epsilon_Y - \epsilon_{\widehat{Y}}\| > 0$, can be deemed as the combination of aleatoric uncertainty and epistemic uncertainty.*

2.2 Coupled Diffusion Probabilistic Model

The diffusion process proposed by [19] is fixed to a Markov chain that gradually adds Gaussian noise to the data (see Appendix E for illustrative examples of the diffusion process on time series data). Specifically, given the input $X = X^{(0)} \sim q(X^{(0)})$, the approximate posterior $q(X^{(1:T)}|X^{(0)})$ can be obtained as

$$q(X^{(1:T)}|X^{(0)}) = \prod_{t=1}^{T} q(X^{(t)}|X^{(t-1)}), \quad q(X^{(t)}|X^{(t-1)}) = \mathcal{N}(X^{(t)}; \sqrt{1-\beta_t}X^{(t)}, \beta_t I), \tag{2}$$

where a monotonically increasing (the value difference is the same among all adjacent pair of points) variance schedule $\beta = \{\beta_1, \cdots, \beta_T \,|\, \beta_t \in [0, 1)\}$ is employed to control the level of noise to be added. Then, let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have

$$q(X^{(t)}|X^{(0)}) = \mathcal{N}(X^{(t)}; \sqrt{\bar{\alpha}_t}X^{(0)}, (1-\bar{\alpha}_t)I). \tag{3}$$

Furthermore, according to Proposition 1 and Eq. (3), we have $X^{(0)} = \langle X_r, \epsilon_X \rangle$ and $X^{(t)} \sim q(X^{(t)}|X^{(0)})$. Consequently, $X^{(t)}$ can be decomposed as follows:

$$X^{(t)} = \sqrt{\bar{\alpha}_t}X^{(0)} + (1-\bar{\alpha}_t)\delta_X := \langle \underbrace{\sqrt{\bar{\alpha}_t}X_r}_{\text{ideal part}}, \underbrace{\sqrt{\bar{\alpha}_t}\epsilon_X + (1-\bar{\alpha}_t)\delta_X}_{\text{noisy part}} \rangle, \tag{4}$$

where δ_X denotes the standard Gaussian noise of X. As α can be determined when the variance schedule β is initialized, the ideal part is also determined in the diffusion process. Let $\widehat{X}_r^{(t)} = \sqrt{\bar{\alpha}_t}X_r$ and $\delta_{\widehat{X}}^{(t)} = \sqrt{\bar{\alpha}_t}\epsilon_X + (1-\bar{\alpha}_t)\delta_X$, then, according to Proposition 1, we have

$$p_\phi(Z^{(t)}|X^{(t)}) = p_\phi(Z^{(t)}|\widehat{X}_r^{(t)}, \delta_{\widehat{X}}^{(t)}), \quad p_\theta(\widehat{Y}^{(t)}|Z^{(t)}) = p_\theta(\widehat{Y}_r^{(t)}|Z^{(t)})p_\theta(\delta_{\widehat{Y}}^{(t)}|Z^{(t)}), \tag{5}$$

where $\delta_{\widehat{Y}}^{(t)}$ denotes the generated noise contributed to $\widehat{Y}^{(t)}$. To relieve the effect of the aleatoric uncertainty resulted from time series data, we further apply the diffusion process to the target series $Y = Y^{(0)} \sim q(Y^{(0)})$. In particular, a scale parameter ω is adopted, such that $\beta'_t = \omega\beta_t, \alpha'_t = 1 - \beta'_t$ and $\bar{\alpha}'_t = \prod_{s=1}^{t} \alpha'_s$. Then, according to Proposition 1, we have the following decomposition:

$$Y^{(t)} = \sqrt{\bar{\alpha}'_t}Y^{(0)} + (1-\bar{\alpha}'_t)\delta_Y := \langle \underbrace{\sqrt{\bar{\alpha}'_t}Y_r}_{\text{ideal part}}, \underbrace{\sqrt{\bar{\alpha}'_t}\epsilon_Y + (1-\bar{\alpha}'_t)\delta_Y}_{\text{noisy part}} \rangle = \langle \widetilde{Y}_r^{(t)}, \delta_{\widetilde{Y}}^{(t)} \rangle. \tag{6}$$

Consequently, we have $q(Y^{(t)}) = q(\widetilde{Y}_r^{(t)})q(\delta_{\widetilde{Y}}^{(t)})$. Afterward, with Proposition 1 and Eqs. (5) and (6) we can draw following conclusions.

Lemma 1. *$\forall \varepsilon > 0$, there exists a probabilistic model $f_{\phi,\theta} := (p_\phi, p_\theta)$ to guarantee that $\mathcal{D}_{\mathrm{KL}}(q(\widetilde{Y}_r^{(t)})||p_\theta(\widehat{Y}_r^{(t)})) < \varepsilon$, where $\widehat{Y}_r^{(t)} = f_{\phi,\theta}(X^{(t)})$.*

Lemma 2. *With the coupled diffusion probabilistic model, $\lim_{t\to\infty} \mathcal{D}_{\mathrm{KL}}(q(\delta_{\widetilde{Y}}^{(t)})||p_\theta(\delta_{\widehat{Y}}^{(t)})|z^{(t)}) < \mathcal{D}_{\mathrm{KL}}(q(\epsilon_Y)||p_\theta(\epsilon_{\widehat{Y}}))$.*

Proof. According to Proposition 1, let $\epsilon_Y = \langle \epsilon_{\widehat{Y}}, \delta_Y \rangle$, then $q(\epsilon_Y) = q(\epsilon_{\widehat{Y}})q(\delta_Y)$. Let $\Delta = \mathcal{D}_{\mathrm{KL}}(q(\epsilon_Y)||p_\theta(\epsilon_{\widehat{Y}})) - \mathcal{D}_{\mathrm{KL}}(q(\epsilon_{\widehat{Y}})||p_\theta(\epsilon_{\widehat{Y}}))$, we have

$$\begin{aligned}\Delta &= \mathcal{D}_{\mathrm{KL}}(q(\epsilon_{\widehat{Y}})q(\delta_Y)||p_\theta(\epsilon_{\widehat{Y}})) - \mathcal{D}_{\mathrm{KL}}(q(\epsilon_{\widehat{Y}})||p_\theta(\epsilon_{\widehat{Y}})) \\ &= \mathcal{D}_{\mathrm{KL}}(q(\epsilon_{\widehat{Y}})||p_\theta(\epsilon_{\widehat{Y}})) + \mathcal{D}_{\mathrm{KL}}(q(\delta_Y)||p_\theta(\epsilon_{\widehat{Y}})) - \mathcal{D}_{\mathrm{KL}}(q(\epsilon_{\widehat{Y}})||p_\theta(\epsilon_{\widehat{Y}})) \\ &= \mathcal{D}_{\mathrm{KL}}(q(\delta_Y)||p_\theta(\epsilon_{\widehat{Y}})) > 0\,,\end{aligned}$$

which leads to $\mathcal{D}_{\mathrm{KL}}(q(\epsilon_Y)||p_\theta(\epsilon_{\widehat{Y}})) > \mathcal{D}_{\mathrm{KL}}(q(\epsilon_{\widehat{Y}})||p_\theta(\epsilon_{\widehat{Y}}) > 0$. Moreover, both $\delta_{\widetilde{Y}}^{(t)}$ and $\delta_{\widehat{Y}}^{(t)}$ are Gaussian noises, when $t \to \infty$, $\exists\ \varepsilon' > 0$, we have $\mathcal{D}_{\mathrm{KL}}(q(\delta_{\widetilde{Y}}^{(t)})||p_\theta(\delta_{\widehat{Y}}^{(t)}|Z^{(t)})) \leq \varepsilon' < \mathcal{D}_{\mathrm{KL}}(q(\epsilon_Y)||p_\theta(\epsilon_{\widehat{Y}}))$. □

Therefore, the uncertainty contributed to the generative model and the inherent data noise can be reduced through the coupled diffusion probabilistic model. In addition, the diffusion process augments the input series, as well as the target series, simultaneously, which can improve the generalization capability for (esp. short) time series forecasting.

2.3 Scaled Denoising Score Matching for Diffused Time Series Cleaning

Although the time series data can be augmented with the aforementioned coupled diffusion probabilistic model, the generative distribution $p_\theta(\widehat{Y}^{(t)})$ tends to move towards the diffused target series $Y^{(t)}$ which have been corrupted [32, 43]. To further "clean" the diffused target series, we employ the Denoising Score Matching (DSM) to accelerate the de-uncertainty process without sacrificing the model flexibility. DSM [46, 32] was proposed to link Denoising Auto-Encoder (DAE) [47] to Score Matching (SM) [20]. Let $\widetilde{Y}$ denote the diffused target series, then we have the objective

$$L_{\mathrm{DSM}}(\zeta) = \mathbb{E}_{p_{\sigma_0}(\widetilde{Y},Y)}\|\nabla_{\widetilde{Y}} \log(q_{\sigma_0}(\widetilde{Y}|Y)) + \nabla_{\widetilde{Y}} E(\widetilde{Y};\zeta)\|^2\,, \tag{7}$$

where $p_{\sigma_0}(\widetilde{Y}, Y)$ is the joint density of pairs of clean and corrupted samples $(Y, \widetilde{Y})$, $\nabla_{\widetilde{Y}} \log(q_{\sigma_0}(\widetilde{Y}|Y))$ is the derivative of log density of the single noise kernel which is dedicated to replace the Parzen density estimator: $p_{\sigma_0}(\widetilde{Y}) = \int q_{\sigma_0}(\widetilde{Y}|Y)p(Y)dY$ in score matching, and $E(\widetilde{Y};\zeta)$ is the energy function. In the particular case of Gaussian noise, $\log(q_{\sigma_0}(\widetilde{Y}|Y)) = -(\widetilde{Y}-Y)^2/2{\sigma_0}^2 + C$. Thus, we have

$$L_{\mathrm{DSM}}(\zeta) = \mathbb{E}_{p_{\sigma_0}(\widetilde{Y},Y)}\|Y - \widetilde{Y} + \sigma_0^2 \nabla_{\widetilde{Y}} E(\widetilde{Y};\zeta)\|^2\,. \tag{8}$$

Then, for the diffused target series at step t, we can obtain

$$L_{\mathrm{DSM}}(\zeta, t) = \mathbb{E}_{p_{\sigma_0}(\widetilde{Y}^{(t)},Y)}\|Y - \widetilde{Y}^{(t)} + \sigma_0^2 \nabla_{\widetilde{Y}^{(t)}} E(\widetilde{Y}^{(t)};\zeta)\|^2 \tag{9}$$

To scale the noise of different levels [32], a monotonically decreasing series of fixed σ values $\{\sigma_1, \cdots, \sigma_T \,|\, \sigma_t = 1 - \bar{\alpha}_t\}$ (refer to the aforementioned variance schedule β in Section 2.2) is adopted. Therefore, the objective for multi-scaled DSM is

$$L(\zeta, t) = \mathbb{E}_{q_\sigma(\widetilde{Y}^{(t)}|Y)p(Y)} l(\sigma_t)\|Y - \widetilde{Y}^{(t)} + \sigma_0^2 \nabla_{\widetilde{Y}^{(t)}} E(\widetilde{Y}^{(t)};\zeta)\|^2\,, \tag{10}$$

where $\sigma \in \{\sigma_1, \cdots, \sigma_T\}$ and $l(\sigma_t) = \sigma_t$. With Eq. (10), we can ensure that the gradient has the right magnitude by setting σ_0.

In the generative time series forecasting setting, the generated samples will be tested without applying the diffusion process. To further denoise the generated target series $\widehat{Y}$, we apply a single step gradient denoising jump [40]:

$$\widehat{Y}_{\text{clean}} = \widehat{Y} - \sigma_0^2 \nabla_{\widehat{Y}} E(\widehat{Y};\zeta)\,. \tag{11}$$

The generated results tend to possess bigger distribution space than the true target, and the noisy term in Eq. (11) approximates the noise between the generated target series and the "cleaned" target series. Therefore, $\sigma_0^2 \nabla_{\widehat{Y}} E(\widehat{Y};\zeta)$ can be treated as the estimated uncertainty of the prediction.

2.4 Disentangling Latent Variables for Interpretation

The interpretability of the time series forecasting model is of great importance for many downstream tasks [44, 17, 21]. With disentangling the latent variables of the generative model, not only the interpretability but also the reliance of the prediction can be further enhanced [31].

To disentangle the latent variable $Z = [z_1, \cdots, z_n]$, we attempt to minimize the Total Correlation (TC) [50, 22], which is a popular metric to measure the dependence among multiple random variables,

$$\mathrm{TC}(z_i) = \mathcal{D}_{\mathrm{KL}}(p_\phi(z_i)||\bar{p}_\phi(z_i)), \qquad \bar{p}_\phi(z_i) = \prod_{j=1}^{m} p_\phi(z_{i,j}) \tag{12}$$

where m denotes the number of factors of z_i that need to be disentangled. In general, lower TC means better disentanglement if the latent variables preserve useful information. However, a very low TC can still be obtained when the latent variables carry no meaningful signals. Through the bidirectional structure of BVAE, such issues can be tackled without too many efforts. As shown in Fig. 1, the signals are disseminated in both encoder and decoder, such that rich semantics are aggregated into the latent variables. Furthermore, to alleviate the effect of potential irregular values, we average the total correlations of $z_{1:n}$, then the loss w.r.t. the TC score of BVAE can be obtained:

$$L_{\mathrm{TC}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{TC}(z_i)\,. \tag{13}$$

Optimization. To reduce the effect of uncertainty, the coupled diffusion equipped with the denoising network is proposed without sacrificing too much generalizability. Then we disentangle the latent variables of the generative model by minimizing the TC of the latent variables. Finally, we reconstruct the loss with several trade off parameters, and with Eqs. (10), (11) and (13) we have

$$\mathcal{L} = \psi \cdot \mathcal{D}_{\mathrm{KL}}(q(Y^{(t)})||p_\theta(\widehat{Y}^{(t)})) + \lambda \cdot L(\zeta, t) + \gamma \cdot L_{\mathrm{TC}} + L_{\mathrm{mse}}(\widehat{Y}^{(t)}, Y^{(t)})\,, \tag{14}$$

where L_{mse} is the MSE between $\widehat{Y}^{(t)}$ and $Y^{(t)}$. We minimize the above loss function to learn the generative model.

3 Experiments

3.1 Experiment Settings

Datasets. We generate two synthetic datasets suggested by [9],

$$w_t = a \cdot w_{t-1} + \tanh(b \cdot w_{t-2}) + \sin(w_{t-3}) + \mathcal{N}(0, 0.5I)$$
$$X = [w_1, w_2, ..., w_N] \cdot F + \mathcal{N}(0, 0.5I)\,,$$

where $w_t \in \mathbb{R}^2$ and $0 \leq w_{t,1}, w_{t,2} \leq 1$ $(t = 1, 2, 3)$, $F \in \mathbb{R}^{2\times k} \sim \mathcal{U}[-1, 1]$, k denotes the dimensionality and N is the number of time points, a, b are two constants. We set $a = 0.9, b = 0.2, k = 20$ to generate D_1, and $a = 0.5, b = 0.5, k = 40$ for D_2, and $N = 800$ for both D_1 and D_2.

Six real-world datasets with diverse spatio-temporal dynamics are selected, including Traffic [27], Electricity[2], Weather[3], WP (Wind Power)[4] and ETTs [53] (ETTm1 and ETTh1). To highlight the uncertainty in short time series scenarios, for each dataset, we slice a subset from the starting point to make sure that each sliced dataset containing at most 1000 time points. Subsequently, we obtain 5%-Traffic, 3%-Electricity, 2%-Weather, 2%-WP, 1%-ETTm1 and 5%-ETTh1. The statistical descriptions of the real-world datasets can be found in Appendix F.1 All datasets are split chronologically and adopt the same train/validation/test ratios, i.e. 7:1:2.

[2]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[3]https://www.bgc-jena.mpg.de/wetter/

[4]We collected this dataset and will publish it upon acceptance.

Table 1: Performance comparisons on synthetic data in terms of MSE and CRPS. The best results are in boldface.

Model		D^{3}VAE	NVAE	β-TCVAE	f-VAE	DeepAR	TimeGrad	GP-copula	VAE
D1	8	**0.512**$_{\pm.033}$	$1.201_{\pm.027}$	$0.631_{\pm.003}$	$0.854_{\pm.099}$	$1.153_{\pm.125}$	$0.966_{\pm.102}$	$1.202_{\pm.108}$	$0.912_{\pm.132}$
		0.585$_{\pm.021}$	$0.905_{\pm.011}$	$0.658_{\pm.002}$	$0.745_{\pm.036}$	$0.758_{\pm.038}$	$0.698_{\pm.024}$	$0.773_{\pm.033}$	$0.786_{\pm.053}$
	16	**0.571**$_{\pm.025}$	$1.184_{\pm.025}$	$0.758_{\pm.047}$	$1.046_{\pm.270}$	$0.911_{\pm.046}$	$0.945_{\pm.315}$	$0.915_{\pm.059}$	$0.908_{\pm.177}$
		0.625$_{\pm.013}$	$0.897_{\pm.012}$	$0.747_{\pm.027}$	$0.835_{\pm.108}$	$0.699_{\pm.014}$	$0.709_{\pm.100}$	$0.704_{\pm.020}$	$0.765_{\pm.067}$
D2	8	**0.599**$_{\pm.049}$	$1.966_{\pm.047}$	$3.096_{\pm.197}$	$3.353_{\pm.430}$	$0.977_{\pm.137}$	$0.963_{\pm.385}$	$1.037_{\pm.082}$	$3.079_{\pm.345}$
		0.628$_{\pm.027}$	$1.255_{\pm.021}$	$1.680_{\pm.062}$	$1.640_{\pm.154}$	$0.727_{\pm.058}$	$0.706_{\pm.123}$	$0.753_{\pm.026}$	$1.504_{\pm.098}$
	16	**0.786**$_{\pm.041}$	$1.955_{\pm.051}$	$3.067_{\pm.443}$	$3.109_{\pm.428}$	$0.972_{\pm.144}$	$0.850_{\pm.061}$	$1.082_{\pm.071}$	$3.132_{\pm.160}$
		$0.728_{\pm.026}$	$1.251_{\pm.020}$	$1.643_{\pm.183}$	$1.558_{\pm.157}$	$0.720_{\pm.050}$	**0.649**$_{\pm.017}$	$0.762_{\pm.008}$	$1.560_{\pm.060}$

Baselines. We compare D^{3}VAE with one GP (Gaussian Process) based method (GP-copula [38]), two auto-regressive methods (DeepAR [39] and TimeGrad [36]), and four VAE-based methods, i.e. vanilla VAE, NVAE [45], factor-VAE (f-VAE for short) [22] and β-TCVAE [6].

Implementation Details. An input-l_x-predict-l_y window is applied to roll the train, validation and test sets with stride one time step, respectively, and this setting is adopted for all datasets. Hereinafter, the last dimension of the multivariate time series is selected as the default target variable.
We use Adam optimizer with initial learning rate $5e-4$. The batch size is 16 and the training is set to 20 epochs at most equipped with early stopping. The number of disentanglement factors is chosen from $\{4, 8\}$, and $\beta_T \in \beta$ is set to range from 0.01 to 0.1 with different diffusion steps $T \in [100, 1000]$, then ω is set to 0.1. The trade-off hyperparameters are set as $\psi = 0.05, \lambda = 0.1, \gamma = 0.001$ for ETTs, and $\psi = 0.5, \lambda = 1.0, \gamma = 0.01$ for others. All the experiments were carried out on a Linux machine with a single NVIDIA P40 GPU. The experiments are repeated 5 times and the average and variance of the predictions are reported. We use the Continuous Ranked Probability Score (CRPS) [34] and Mean Squared Error (MSE) as the evaluation metrics, for both metrics, the lower the better. In particular, CRPS is used to evaluate the similarity of two distributions and is equivalent to Mean Absolute Error (MAE) when two distributions are discrete.

3.2 Main Results

Two different prediction lengths, i.e. $l_y \in \{8, 16\}$ and $l_x = l_y$, are evaluated. The results of longer prediction lengths are available in supplementary materials (see Appendix D).

Toy Datasets. In Table 1, we can observe that D^{3}VAE achieves SOTA performance most of the time, and achieves competitive CRPS in D_2 for prediction length 16. Besides, VAEs outperform VARs and GP on D_1, but VARs achieve better performance on D_2, which demonstrate the advantage of VARs in learning complex temporal dependencies.

Real-World Datasets. As for the experiments on real-world data, D^{3}VAE achieves consistent SOTA performance except for the prediction length 16 on Wind dataset (Table 2). Particularly, under the input-8-predict-8 setting, D^{3}VAE can provide remarkable improvements on Traffic, Electricity, Wind, ETTm1, ETTh1 and Weather w.r.t. MSE reduction (90%, 71%, 48%, 43%, 40% and 28%). Regarding the CRPS reduction, D^{3}VAE achieves 73% reduction on Traffic, 31% on Wind and 27% on Electricity under the input-8-predict-8 setting, and 70% reduction on Traffic, 18% on electricity and 7% on Weather under the input-16-predict-16 setting. Overall, D^{3}VAE gains 43% MSE reduction and 23% CRPS reduction among above settings. More results under the longer prediction setting and full dataset setting can be found in supplementary materials.

Uncertainty Estimation. The uncertainty can be assessed by estimating the noise of the outcome series when doing the prediction (see Section 2.3). Through scale parameter ω, the generated distribution space can be adjusted accordingly (results on the effect of ω can be found in Appendix D.3). The showcases in Fig. 2 demonstrate the uncertainty estimation of the yielded series in Traffic dataset,

Table 2: The performance comparisons on real-world datasets in terms of MSE and CRPS, and the best results are in boldface.

Model		D^{3}VAE	NVAE	β-TCVAE	f-VAE	DeepAR	TimeGrad	GP-copula	VAE
Traffic	8	**0.081**$_{\pm.003}$	$1.300_{\pm.024}$	$1.003_{\pm.006}$	$0.982_{\pm.059}$	$3.895_{\pm.306}$	$3.695_{\pm.246}$	$4.299_{\pm.372}$	$0.794_{\pm.130}$
		0.207$_{\pm.003}$	$0.593_{\pm.004}$	$0.894_{\pm.003}$	$0.666_{\pm.032}$	$1.391_{\pm.071}$	$1.410_{\pm.027}$	$1.408_{\pm.046}$	$0.759_{\pm.07}$
	16	**0.081**$_{\pm.009}$	$1.271_{\pm.019}$	$0.997_{\pm.004}$	$0.998_{\pm.042}$	$4.141_{\pm.320}$	$3.495_{\pm.362}$	$4.575_{\pm.141}$	$0.632_{\pm.057}$
		0.200$_{\pm.014}$	$0.589_{\pm.001}$	$0.893_{\pm.002}$	$0.692_{\pm.026}$	$1.338_{\pm.043}$	$1.329_{\pm.057}$	$1.506_{\pm.025}$	$0.671_{\pm.038}$
Electricity	8	**0.251**$_{\pm.015}$	$1.134_{\pm.029}$	$0.901_{\pm.052}$	$0.893_{\pm.069}$	$2.934_{\pm.173}$	$2.703_{\pm.087}$	$2.924_{\pm.218}$	$0.853_{\pm.040}$
		0.398$_{\pm.011}$	$0.542_{\pm.003}$	$0.831_{\pm.004}$	$0.809_{\pm.024}$	$1.244_{\pm.037}$	$1.208_{\pm.024}$	$1.249_{\pm.048}$	$0.795_{\pm.016}$
	16	**0.308**$_{\pm.030}$	$1.150_{\pm.032}$	$0.850_{\pm.003}$	$0.807_{\pm.034}$	$2.803_{\pm.199}$	$2.770_{\pm.237}$	$3.065_{\pm.186}$	$0.846_{\pm.062}$
		0.437$_{\pm.020}$	$0.531_{\pm.003}$	$0.814_{\pm.002}$	$0.782_{\pm.024}$	$1.220_{\pm.048}$	$1.240_{\pm.048}$	$1.307_{\pm.042}$	$0.793_{\pm.029}$
Weather	8	**0.169**$_{\pm.022}$	$0.801_{\pm.024}$	$0.234_{\pm.042}$	$0.591_{\pm.198}$	$2.317_{\pm.357}$	$2.715_{\pm.189}$	$2.412_{\pm.761}$	$0.560_{\pm.192}$
		0.357$_{\pm.024}$	$0.757_{\pm.013}$	$0.404_{\pm.040}$	$0.565_{\pm.080}$	$0.858_{\pm.078}$	$0.920_{\pm.013}$	$0.897_{\pm.115}$	$0.572_{\pm.077}$
	16	**0.187**$_{\pm.047}$	$0.811_{\pm.016}$	$0.212_{\pm.012}$	$0.530_{\pm.167}$	$1.269_{\pm.187}$	$1.110_{\pm.083}$	$1.357_{\pm.145}$	$0.424_{\pm.141}$
		0.361$_{\pm.046}$	$0.759_{\pm.009}$	$0.388_{\pm.014}$	$0.547_{\pm.067}$	$0.783_{\pm.059}$	$0.733_{\pm.016}$	$0.811_{\pm.032}$	$0.503_{\pm.068}$
ETTm1	8	**0.527**$_{\pm.073}$	$0.921_{\pm.026}$	$1.538_{\pm.254}$	$2.326_{\pm.445}$	$2.204_{\pm.420}$	$1.877_{\pm.245}$	$2.024_{\pm.143}$	$2.375_{\pm.405}$
		0.557$_{0.048}$	$0.760_{\pm.026}$	$1.015_{\pm.112}$	$1.260_{\pm.167}$	$0.984_{\pm.074}$	$0.908_{\pm.038}$	$0.961_{\pm.027}$	$1.258_{\pm.104}$
	16	**0.968**$_{\pm.104}$	$1.100_{\pm.032}$	$1.744_{\pm.100}$	$2.339_{\pm.270}$	$2.350_{\pm.170}$	$2.032_{\pm.234}$	$2.486_{\pm.207}$	$2.321_{\pm.469}$
		0.821$_{\pm.072}$	$0.822_{\pm.026}$	$1.104_{\pm.041}$	$1.249_{\pm.088}$	$0.974_{\pm.016}$	$0.919_{\pm.031}$	$0.984_{\pm.016}$	$1.259_{\pm.132}$
ETTh1	8	**0.292**$_{\pm.036}$	$0.483_{\pm.017}$	$0.703_{\pm.054}$	$0.870_{\pm.134}$	$3.451_{\pm.335}$	$4.259_{\pm1.13}$	$4.278_{\pm1.12}$	$1.006_{\pm.281}$
		0.424$_{\pm.033}$	$0.461_{\pm.011}$	$0.644_{\pm.038}$	$0.730_{\pm.060}$	$1.194_{\pm.034}$	$1.092_{\pm.028}$	$1.169_{\pm.055}$	$0.762_{\pm.115}$
	16	**0.374**$_{\pm.061}$	$0.488_{\pm.010}$	$0.681_{\pm.018}$	$0.983_{\pm.139}$	$1.929_{\pm.105}$	$1.332_{\pm.125}$	$1.701_{\pm.088}$	$0.681_{\pm.104}$
		$0.488_{\pm.039}$	**0.463**$_{\pm.018}$	$0.640_{\pm.008}$	$0.760_{\pm.062}$	$1.029_{\pm.030}$	$0.879_{\pm.037}$	$0.999_{\pm.023}$	$0.641_{\pm.055}$
Wind	8	**0.681**$_{\pm.075}$	$1.854_{\pm.032}$	$1.321_{\pm.379}$	$1.942_{\pm.101}$	$12.53_{\pm2.25}$	$12.67_{\pm1.75}$	$11.35_{\pm6.61}$	$2.006_{\pm.145}$
		0.596$_{\pm.052}$	$1.223_{\pm.014}$	$0.863_{\pm.143}$	$1.067_{\pm.086}$	$1.370_{\pm.107}$	$1.440_{\pm.059}$	$1.305_{\pm.369}$	$1.103_{\pm.100}$
	16	$1.033_{\pm.062}$	$1.955_{\pm.015}$	**0.894**$_{\pm.038}$	$1.262_{\pm.178}$	$13.96_{\pm.1.53}$	$12.86_{\pm2.60}$	$13.79_{\pm5.37}$	$1.138_{\pm.205}$
		0.757$_{\pm.053}$	$1.247_{\pm.011}$	$0.785_{\pm.037}$	$0.843_{\pm.066}$	$1.347_{\pm.060}$	$1.240_{\pm.070}$	$1.261_{\pm.171}$	$0.862_{\pm.092}$

where the last six dimensions are treated as the target variables. We can find that the noise estimation can quantify the uncertainty effectively. For example, the uncertainty grows rapidly when extreme values are encountered.

Disentanglement Evaluation. It is difficult to label disentangled factors by hand for time series forecasting, thus we take different dimensions of Z as the factors to be disentangled: $Z = [z_1, \cdots, z_n]$, then an instance consisting of factor and label $(z_{i,j}, j)$ is constructed. We shuffle these n examples and attempt to classify them with a discriminator, then the disentanglement can be evaluated by measuring the loss of discriminator. Due to the space limit, the evaluation of disentanglement with different factors can be seen in supplementary materials (Appendix C).

3.3 Model Analysis

Ablation Study of the Coupled Diffusion and Denoising Network. To evaluate the effectiveness of the coupled diffusion model (CDM), we compare the full versioned D^{3}VAE with its three variants: i) D^{3}VAE$_{-\widetilde{Y}}$, i.e. D^{3}VAE without diffused Y, ii) D^{3}VAE$_{-\widehat{X}}$, i.e. D^{3}VAE without diffused X, and iii) D^{3}VAE$_{-\mathrm{CDM}}$, i.e. D^{3}VAE without any diffusion. Besides, the performance of D^{3}VAE without the denoising network (i.e. DSM) is also reported when the target series is not diffused, which are denoted as D^{3}VAE$_{-\widetilde{Y}-\mathrm{DSM}}$ and D^{3}VAE$_{-\mathrm{CDM}-\mathrm{DSM}}$. The ablation study is carried out on Traffic and Electricity datasets under the settings of input-16-predict-16 and input-32-predict-32. In Table 3, we can find that the diffusion process is effective in augmenting the input or the target. Moreover,

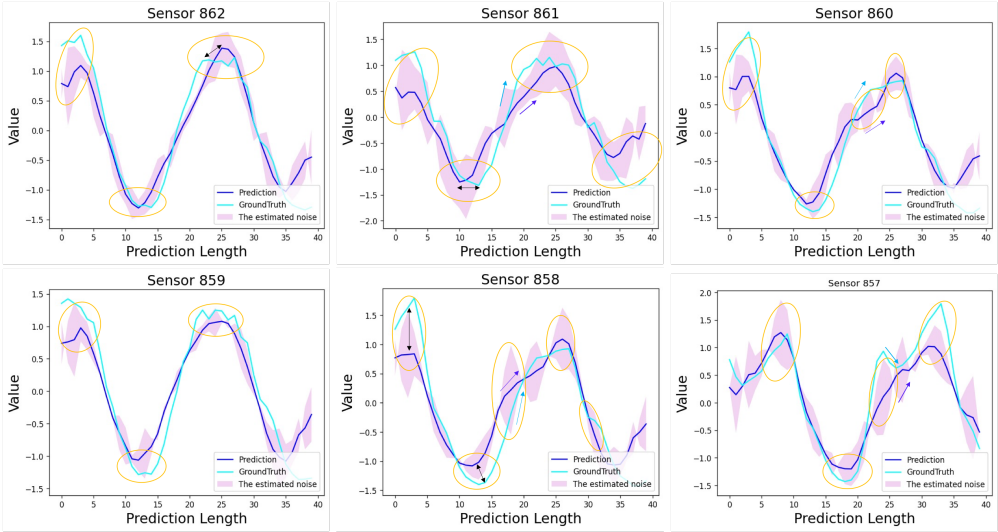


Figure 2: The uncertainty estimation of the prediction of the last six dimensions of the Traffic dataset, and the colored envelop denotes the estimated uncertainty.

Table 3: Ablation study of the coupled diffusion probabilistic model w.r.t. MSE and CSPR.

Dataset	Traffic		Electricity	
	16	32	16	32
$\text{D}^3\text{VAE}_{-\widetilde{Y}}$	$0.122_{\pm.006}$ $0.250_{\pm.008}$	$0.126_{\pm.013}$ $0.261_{\pm.017}$	$0.350_{\pm.043}$ $0.480_{\pm.032}$	$0.422_{\pm.012}$ $0.551_{\pm.012}$
$\text{D}^3\text{VAE}_{-\widetilde{Y}-\text{DSM}}$	$0.096_{\pm.006}$ $0.217_{\pm.010}$	$0.092_{\pm.008}$ $0.220_{\pm.013}$	$0.331_{\pm.023}$ $0.450_{\pm.021}$	$0.502_{\pm.079}$ $0.584_{\pm.053}$
$\text{D}^3\text{VAE}_{-\widehat{X}}$	$0.123_{\pm.003}$ $0.256_{\pm.006}$	$0.117_{\pm.007}$ $0.253_{\pm.013}$	$0.351_{\pm.047}$ $0.481_{\pm.036}$	$0.420_{\pm.056}$ $0.540_{\pm.046}$
$\text{D}^3\text{VAE}_{-\text{CDM}}$	$0.123_{\pm.004}$ $0.255_{\pm.007}$	$0.118_{\pm.008}$ $0.252_{\pm.015}$	$0.365_{\pm.025}$ $0.498_{\pm.018}$	$0.439_{\pm.014}$ $0.561_{\pm.016}$
$\text{D}^3\text{VAE}_{-\text{CDM}-\text{DSM}}$	$0.123_{\pm.003}$ $0.255_{\pm.003}$	$0.119_{\pm.003}$ $0.253_{\pm.005}$	$0.338_{\pm.041}$ $0.467_{\pm.029}$	$0.448_{\pm.062}$ $0.555_{\pm.041}$
D^3VAE	$\mathbf{0.081}_{\pm.009}$ $\mathbf{0.200}_{\pm.014}$	$\mathbf{0.091}_{\pm.007}$ $\mathbf{0.216}_{\pm.012}$	$\mathbf{0.308}_{\pm.030}$ $\mathbf{0.437}_{\pm.020}$	$\mathbf{0.410}_{\pm.075}$ $\mathbf{0.534}_{\pm.058}$

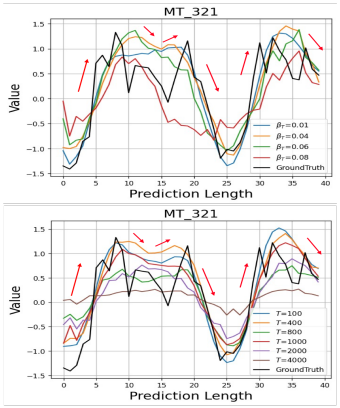


Figure 3: Comparisons of the predictions with different β_T and varying T on the Electricity dataset.

when the target is not diffused, the denoising network would be deficient for the noise level of the target cannot be estimated by then.

Variance Schedule β and The Number of Diffusion Steps T. In order to reduce the effect of the uncertainty while preserving the informative temporal patterns, the extent of the diffusion should be configured properly. Too small variance schedule or inadequate diffusion steps will lead to a meaningless diffusion process, on the contrary, the diffusion will be out of control. Here we analyze the effect of the variance schedule β and the number of diffusion steps T. We set $\beta_1 = 0$ and change the value of β_T in the range of $[0.01, 0.1]$, and T ranges from 100 to 4000. As shown in Fig. 3, we can see that the prediction performance can be improved if proper β_T and T are adopted.

4 Discussion

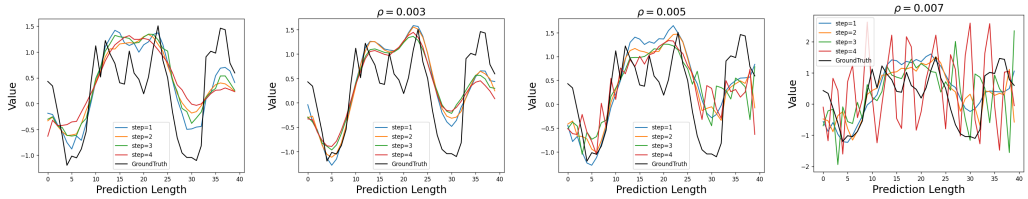
Sampling for Generative Time Series Forecasting.
The Langevin dynamics has been widely applied to the sampling of the energy based mod-

els (EBMs) [8],

$$Y_k = Y_{k-1} - \frac{\rho}{2}\nabla_Y E_\phi(Y_{k-1}) + \rho^{\frac{1}{2}}\mathcal{N}(0, I_d)\,, \tag{15}$$

where $k \in \{0, \cdots, K\}$, K denotes the number of sampling steps, and ρ is a constant. With K and ρ being properly configured, the samples with high quality can be generated. The Langevin dynamics has been successfully applied to image reconstruction and language processing [26, 7].

In this work, we employ a single-step gradient denoising jump to generate the target series. The experiments carried out demonstrate the effectiveness of such single step sampling. To investigate whether it is worthy to take more sampling steps for further performance improvement of time series forecasting, we conduct additional empirical study. We showcase the prediction results under different sampling strategies in Fig. 4. By omitting the additive noise in Langevin dynamics, we employ the multi-step denoising for D^{3}VAE to generate the target series and plot the generated results in Fig. 4a. Then, with the standard Langevin dynamics we can implement a generative procedure instead of denoising and compare the generated target series with different ρ (see Figs. 4b to 4d). We can observe that, more sampling steps might not be helpful in improving prediction performance for generative time series forecasting (Fig. 4a). Besides, larger sampling steps would lead to high computational complexity. On the other hand, different configurations of Langevin dynamics (with varying ρ) cannot bring indispensable benefits for time series forecasting (Figs. 4b to 4d).



(a) Multi-step denoising. (b) $\rho = 0.003$. (c) $\rho = 0.005$. (d) $\rho = 0.007$.

Figure 4: The prediction showcases in the Electricity dataset with different sampling strategies.

Limitations.

With the coupled diffusion probabilistic model, the aleatoric uncertainty of time series can be reduced though, new bias is brought into the series to mimic the distribution of the input and target. However, as a common issue in VAEs that any introduced bias in the input will result in bias in the generated output [48], the diffusion steps and variance schedule need to be chosen cautiously, such that this model can be applied to different time series tasks smoothly. And the proposed model is devised for general time series forecasting, it should be used properly to avoid the potential negative societal impacts, such as any kind of illegal applications.

In time series predictive analysis, disentanglement of the latent variables has been of great importance for interpreting the prediction to provide more reliance. Due to the lack of prior knowledge of the entangled factors in generative time series forecasting, only the unsupervised disentanglement learning can be done, which has been proven theoretically feasible for time series [31]. Despite this, for boarder applications of disentanglement and better performance, it is still worth of exploring the ways of labeling the factors of time series in the future. Moreover, because of the uniqueness of time series data, it is also an promising direction to explore more generative and sampling methods for time series generation task.

5 Conclusion

In this work, we propose a generative model with the bidirectional VAE as the backbone. To further improve the generalizability, we devise a coupled diffusion probabilistic model for time series forecasting. Then a scaled denoising network is developed to guarantee the prediction accuracy. Afterward, the latent variables are further disentangled for better model interpretability. Extensive experiments on synthetic data and real-world data validate that our proposed generative model achieves SOTA performance comparing to existing competitive generative models.

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[3] Mikolaj Binkowski, Gautier Marti, and Philippe Donnat. Autoregressive convolutional neural networks for asynchronous time series. In *International Conference on Machine Learning*, pages 580–589. PMLR, 2018.

[4] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

[5] Li-Juan Cao and Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6):1506–1518, 2003.

[6] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[7] Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation. *arXiv preprint arXiv:2004.11714*, 2020.

[8] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

[9] Amirreza Farnoosh, Bahar Azari, and Sarah Ostadabbas. Deep switching auto-regressive factorization: Application to time series forecasting. *arXiv preprint arXiv:2009.05135*, 2020.

[10] Konstantinos Fokianos, Anders Rahbek, and Dag Tjøstheim. Poisson autoregression. *Journal of the American Statistical Association*, 104(488):1430–1439, 2009.

[11] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pages 1651–1661. PMLR, 2020.

[12] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. Som-vae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199*, 2018.

[13] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[14] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

[15] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[16] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[17] Michaela Hardt, Alvin Rajkomar, Gerardo Flores, Andrew Dai, Michael Howell, Greg Corrado, Claire Cui, and Moritz Hardt. Explaining an increase in predicted risk for clinical alerts. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 80–89, 2020.

[18] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[20] Aapo Hyvärinen, Jarmo Hurri, and Patrik O Hoyer. Estimation of non-normalized statistical models. In *Natural Image Statistics*, pages 419–426. Springer, 2009.

[21] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

[22] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[23] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[24] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[25] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[26] Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

[27] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104, 2018.

[28] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pages 47–54. Springer, 2016.

[29] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32, 2019.

[30] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. *Advances in neural information processing systems*, 29, 2016.

[31] Yuening Li, Zhengzhang Chen, Daochen Zha, Mengnan Du, Denghui Zhang, Haifeng Chen, and Xia Hu. Learning disentangled representations for time series. *arXiv preprint arXiv:2105.08179*, 2021.

[32] Zengyi Li, Yubei Chen, and Friedrich T Sommer. Learning energy-based models in high-dimensional spaces with multi-scale denoising score matching. *arXiv preprint arXiv:1910.07762*, 2019.

[33] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[34] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

[35] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[36] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

[37] Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. *Advances in Neural Information Processing Systems*, 30, 2017.

[38] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32, 2019.

[39] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[40] Saeed Saremi and Aapo Hyvarinen. Neural empirical bayes. *arXiv preprint arXiv:1903.02334*, 2019.

[41] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[42] Qiquan Shi, Jiaming Yin, Jiajun Cai, Andrzej Cichocki, Tatsuya Yokota, Lei Chen, Mingxuan Yuan, and Jia Zeng. Block hankel tensor arima for multiple short time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5758–5766, 2020.

[43] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[44] Sana Tonekaboni, Shalmali Joshi, Kieran Campbell, David K Duvenaud, and Anna Goldenberg. What went wrong and when? instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems*, 33:799–809, 2020.

[45] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

[46] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[47] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[48] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in Neural Information Processing Systems*, 34, 2021.

[49] Juntao Wang, Wun Kwan Yam, Kin Long Fong, Siew Ann Cheong, and KY Wong. Gaussian process kernels for noisy time series: Application to housing price prediction. In *International Conference on Neural Information Processing*, pages 78–89. Springer, 2018.

[50] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

[51] Jiehui Xu, Jianmin Wang, Mingsheng Long, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34, 2021.

[52] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[53] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

[54] Yong Zou, Reik V Donner, Norbert Marwan, Jonathan F Donges, and Jürgen Kurths. Complex network approaches to nonlinear time series analysis. *Physics Reports*, 787:1–97, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4 for details.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discussÅ the potential negative societal impacts in Section 4
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Our problem formulation and theoretical proof are described in Section 2, and all the theoretical results are with reasonable and necessary assumptions.
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We use six real-world datasets, for each of them is provided by URL or citation in Section 3.1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3 and Appendix F
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All the experimental results are repeated five times, the mean and standard deviation of the results are provided in Section 3
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.1
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] The datasets and baseline models that used in this paper are all cited in Section 3.1.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The newly introduced dataset Wind is provided in the supplementary materials and will be released later.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The datasets used in this paper are all public datasets, and we didn't use/curate data from other people.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data we use doesn't contain any identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]