A Related Work

A.1 Time Series Forecasting

We first briefly review the related literature of time series forecasting (TSF) methods as below. Complex temporal patterns can be manifested over short- and long-term as the time series evolves across time. To leverage the time evolution nature, existing statistical models, such as ARIMA [6] and Gaussian process regression [7] have been well established and applied to many downstream tasks [28, 29, 2]. Recurrent neural network (RNN) models are also introduced to model temporal dependencies for TSF in a sequence-to-sequence paradigm [24, 9, 61, 40, 46, 50, 53]. Besides, temporal attention [49, 59, 56] and causal convolution [3, 5, 54] are further explored to model the intrinsic temporal dependencies. Recent Transformer-based models have strengthened the capability of exploring hidden intricate temporal patterns for long-term TSF [67, 42, 65, 71]. On the other hand, the multivariate nature of TSF is another topic many works have been focusing on. These works treat a collection of time series as a unified entity and mine the inter-series correlations with different techniques, such as probabilistic models [53, 50], matrix/tensor factorization [54, 55], convolution neural networks (CNNs) [3, 40], and graph neural networks (GNNs) [26, 43, 70, 66, 8].

To improve the reliability and performance of TSF, instead of modeling the raw data, there exist works inferring the underlying distribution of the time series data with generative models [69, 14]. Many studies have employed a variational auto-encoder (VAE) to model the probabilistic distribution of sequential data [21, 23, 12, 10, 47]. For example, VRNN [12] employs the VAE to each hidden state of RNN such that the variability of highly structured sequential data can be captured. To yield predictive distribution for multivariate TSF, TLAE [47] implements nonlinear transformation by replacing matrix factorization with encoder-decoder architecture and temporal deep temporal latent model. Another line of generative methods for TSF focus on energy-based models (EBMs), such as TimeGrad [51] and ScoreGrad [68]. EBMs do not restrict the tractability of the normalizing constants [68]. Though flexible, the unknown normalizing constant makes the training of EBMs particularly difficult.

This paper focuses on TSF with VAE-based models. Besides, as many real-world time series data are relatively short and small [58], a coupled probabilistic diffusion model is proposed to augment the input series, as well as the output series, simultaneously, such that the distribution space can be enlarged without increasing the aleatoric uncertainty [34]. Moreover, to guarantee the generated target series moving toward the true target, a multi-scaled score-matching denoising network is plugged in for accurate future series prediction. To our knowledge, this is the first work focusing on generative TSF with the diffusion model and denoising techniques.

A.2 Time Series Augmentation

Both the traditional methods and deep learning methods can deteriorate when limited time series data are encountered. Generating synthetic time series is commonly adopted for augmenting short time series [13, 18, 69]. Transforming the original time series by cropping, flipping, and warping [32, 15] is another approach dedicated to TSF when the training data is limited. Whereas the synthetic time series may not respect the original feature relationship across time, and the transformation methods do not change the distribution space. Thus, the overfitting issues cannot be avoided. Incorporating the probabilistic diffusion model for TSF differentiates our work from existing time series augmentation methods.

A.3 Uncertainty Estimation and Denoising for Time Series Forecasting

There exist works aiming to estimate the uncertainty [34] for time series forecasting [48, 62, 25] by epistemic uncertainty. Nevertheless, the inevitable aleatoric uncertainty of time series is often ignored, which may stem from error-prone data measurement, collection, and so forth [63]. Another line of studies focuses on detecting noise in time series data [45] or devising suitable models for noise alleviation [22]. However, none of the existing works attempts to quantify the aleatoric uncertainty, which further differentiates our work from priors.

It is necessary to relieve the effect of noise in real-world time series data [16]. [4, 39] propose to preprocess the time series with smoothing and filtering techniques. However, such preprocessing methods can only be applied to the noise raised by the irregular data of time series. Neural networks

are also introduced to denoise the time series [20, 57, 22, 33], while these deep networks can only deal with specific types of time series as well.

A.4 Interpretability of Time Series Forecasting

A number of works put effort into explaining the deep neural networks [64, 35, 1] to make the prediction more interpretable, but these methods often lack reliability when the explanation is sensitive to factors that do not contribute to the prediction [37]. Several works have been proposed to increase the reliability of TSF tasks [30, 31]. For multivariate time series, the interpretability of the representations can be improved by mapping the time series into latent space [19]. Besides, recent works have been proposed to disentangle the latent variables to identify the independent factors of the data, which can further lead to improved interpretability of the representation and higher performance [27, 41, 36]. The disentangled VAE has been applied to time series to benefit the generated results [44]. However, the choice of the latent variables is crucial for the disentanglement of time series data. We devise a bidirectional VAE (BVAE) and take the dimensions of each latent variable as the factors to be disentangled.

B Proofs of Lemma 1 and Lemma 2

With the coupled diffusion process and Eqs. (5) and (6), as well as Proposition 1, introduced in the main text, the diffused target series and generated target series can be decomposed as $\widetilde{Y}^{(t)} = \langle \widetilde{Y}^{(t)}, \delta_{\widetilde{Y}}^{(t)} \rangle$ and $\widehat{Y}^{(t)} = \langle \widehat{Y}^{(t)}, \delta_{\widehat{Y}}^{(t)} \rangle$. Then, we can draw the following two 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 $\widehat{Y}_r^{(t)} = f_{\phi,\theta}(X^{(t)})$.*

Proof. According to Proposition 1, $\widehat{Y}_r$ can be fully captured by the model. That is, $\|Y_r - \widehat{Y}_r\| \longrightarrow 0$ where Y_r is the ideal part of ground truth target series Y. And, with Eq. (6) (in the main text), $\widetilde{Y}_r^{(t)} = \sqrt{\overline{\alpha}_t} Y_r$. Therefore, $\|\widetilde{Y}_r^{(t)} - \widehat{Y}_r^{(t)}\| \longrightarrow 0$ when $t \to \infty$. □

Lemma 2. *With the coupled diffusion process, the difference between diffusion noise and generation noise will be reduced, i.e., $\lim_{t\to\infty} \mathcal{D}_{\mathrm{KL}}(q(\delta_{\widetilde{Y}}^{(t)}) || p_\theta(\delta_{\widehat{Y}}^{(t)} | Z^{(t)})) < \mathcal{D}_{\mathrm{KL}}(q(\epsilon_Y) || p_\theta(\epsilon_{\widehat{Y}}))$.*

Proof. According to Proposition 1, the noise of Y consists of the estimation noise $\epsilon_{\widehat{Y}}$ and residual noise δ_Y, i.e., $\epsilon_Y = \langle \epsilon_{\widehat{Y}}, \delta_Y \rangle$ where $\epsilon_{\widehat{Y}}$ and δ_Y are independent of each other, 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 $\mathcal{D}_{\mathrm{KL}}(q(\delta_{\widetilde{Y}}^{(t)}) || p_\theta(\delta_{\widehat{Y}}^{(t)} | Z^{(t)})) \leq \varepsilon' < \mathcal{D}_{\mathrm{KL}}(q(\epsilon_Y) || p_\theta(\epsilon_{\widehat{Y}}))$. □

C Extra Implementation Details

C.1 Experimental Settings

Datasets Description. The main descriptive statistics of the real-world datasets adopted in the experiments of this work are demonstrated in Table 5.

Input Representation. We adopt the embedding method introduced in [71] and feed it to an RNN to extract the temporal dependency. Then we concatenate them as follows:

$$
X_{\text{input}} = \mathrm{CONCAT}(\mathrm{RNN}(\mathcal{E}(X)), \mathcal{E}(X)) \,,
$$

Table 5: Statistical descriptions of the real-world datasets.

Datasets	# Dims.	Full Data		Sliced Data		Target Variable	Time Interval
		Time Span	# Points	Pct. of Full Data	# Points		
Traffic	862	2015-2016	17544	5%	877	Sensor 862	1 hour
Electricity	321	2011-2014	18381	3%	551	MT_321	10 mins
Weather	21	2020-2021	36761	2%	735	CO2 (ppm)	10 mins
ETTm1	7	2016-2018	69680	1%	697	OT	15 mins
ETTh1	7	2016-2018	17420	5%	871	OT	1 hour
Wind	7	2020-2021	45550	2%	911	wind_power	15 mins

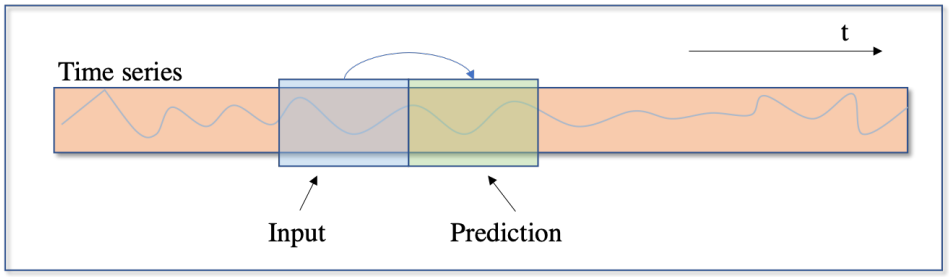


Figure 6: Forecasting process of DeepAR, TimeGrad, and GP-copula. The sliding step is set to 1.

where X is the raw time series data and $\mathcal{E}(\cdot)$ denotes the embedding operation. Here, we use a two-layer gated recurrent unit (GRU), and the dimensionality of the hidden state and embeddings are 128 and 64, respectively.

Diffusion Process Configuration. Besides, the diffusion process is configured to be $\beta_t \in [0, 0.1]$ and $T = 100$ for the **Weather** dataset, $\beta_t \in [0, 0.1]$ and $T = 1000$ for the **ETTh1** dataset, $\beta_t \in [0, 0.08]$ and $T = 1000$ for the **Wind** dataset, and $\beta_t \in [0, 0.01]$ and $T = 1000$ for the other datasets.

C.2 Implementation Details of Baselines

We select previous state-of-the-art generative models as our baselines in the experiments on synthetic and real-world datasets. Specifically, **1) GP-copula** [52] is a method based on the Gaussian process, which is devoted to high-dimensional multivariate time series, **2) DeepAR** [53] combines traditional auto-regressive models with RNNs by modeling a probabilistic distribution in an auto-encoder fashion, **3) TimeGrad** [51] is an auto-regressive model for multivariate probabilistic time series forecasting with the help of an energy-based model, **4) Vanilla VAE** (VAE for short) [38] is a classical statistical variational inference method on top of auto-encoder, **5) NVAE** [60] is a deep hierarchical VAE built for image generation using depth-wise separable convolutions and batch normalization, **6) factor-VAE** (f-VAE for short) [36] disentangles the latent variables by encouraging the distribution of representations to be factorial and independent across dimensions, and **7)** β**-TCVAE** [11] learns the disentangled representations with total correlation variational auto-encoder algorithm.

To train DeepAR, TimeGrad, and GP-copula in accordance with their original settings, the batch is constructed without shuffling the samples. The instances (sampled with the input-l_x-predict-l_y rolling window and $l_x = l_y$, as illustrated in Fig. 6) are fed to the training procedure of these three baselines in chronological order. Besides, these three baselines employ the cumulative distribution function (CDF) for training, so the CDF needs to be reverted to the real distribution for testing.

For f-VAE, β-TCVAE, and VAE, since the dimensionality of different time series varies, we design a preprocess block to map the original time series into a tensor with the fix-sized dimensionality, which can further suit the VAEs well. The preprocess block consists of three nonlinear layers with the sizes of the hidden states: $\{128, 64, 32\}$. For NVAE, we keep the original settings suggested in [60] and use Gaussian distribution as the prior. All the baselines are trained using early stopping, and the patience is set to 5.

Table 6: Performance comparisons of short-term and long-term TSF in real-world datasets in terms of MSE and CRPS. For MSE and CRPS, the lower, the better. 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$_{\pm.042}$	4.140$_{\pm.320}$	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$_{\pm.038}$
	32	**0.091**$_{\pm.007}$	0.126$_{\pm.013}$	1.254$_{\pm0.019}$	0.977$_{\pm.002}$	4.234$_{\pm.139}$	5.195$_{\pm2.26}$	3.717$_{\pm.361}$	0.735$_{\pm.084}$
		0.216$_{\pm.012}$	0.422$_{\pm.012}$	0.937$_{0.007}$	0.882$_{\pm.001}$	1.367$_{\pm.015}$	1.565$_{\pm.329}$	1.342$_{\pm.048}$	0.735$_{\pm.048}$
	64	**0.125**$_{\pm.005}$	1.263$_{\pm0.014}$	0.903$_{\pm.111}$	0.936$_{\pm.190}$	3.381$_{\pm.130}$	3.692$_{\pm1.54}$	3.492$_{\pm.092}$	0.692$_{\pm.059}$
		0.244$_{\pm.006}$	0.940$_{\pm0.005}$	0.839$_{\pm.062}$	0.829$_{\pm.078}$	1.233$_{\pm.027}$	1.412$_{\pm0.257}$	1.367$_{\pm.012}$	0.710$_{\pm.035}$
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$_{\pm.029}$
	32	**0.410**$_{\pm.075}$	1.302$_{\pm0.011}$	0.844$_{\pm.025}$	0.861$_{\pm.105}$	2.402$_{\pm.156}$	2.640$_{\pm.138}$	2.880$_{\pm.221}$	0.841$_{\pm.071}$
		0.534$_{\pm.058}$	0.944$_{\pm0.005}$	0.808$_{\pm.005}$	0.797$_{\pm.037}$	1.130$_{\pm.055}$	1.234$_{\pm.027}$	1.281$_{\pm.054}$	0.790$_{\pm.026}$
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$_{\pm.068}$
	32	**0.203**$_{\pm.008}$	0.836$_{\pm0.014}$	0.439$_{\pm.394}$	0.337$_{\pm.086}$	2.518$_{\pm.546}$	1.178$_{\pm.069}$	1.065$_{\pm.145}$	0.329$_{\pm.083}$
		0.383$_{\pm.007}$	0.777$_{\pm0.007}$	0.508$_{\pm.176}$	0.461$_{\pm.031}$	0.847$_{\pm.036}$	0.724$_{\pm.021}$	0.747$_{\pm.035}$	0.459$_{\pm.045}$
	64	**0.191**$_{\pm.022}$	0.932$_{0.020}$	0.276$_{\pm.026}$	0.676$_{\pm.484}$	3.595$_{\pm.956}$	1.063$_{\pm.061}$	0.992$_{\pm.114}$	0.721$_{\pm.496}$
		0.358$_{\pm.044}$	0.836$_{0.009}$	0.463$_{\pm.026}$	0.612$_{\pm.176}$	0.994$_{\pm.100}$	0.696$_{\pm.011}$	0.699$_{\pm.016}$	0.635$_{\pm.204}$
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$_{\pm.132}$
	32	**0.707**$_{\pm.061}$	1.298$_{\pm.028}$	1.438$_{\pm.429}$	2.563$_{\pm.358}$	4.855$_{\pm.179}$	1.251$_{\pm.133}$	1.402$_{\pm.187}$	2.660$_{\pm.349}$
		0.697$_{\pm.040}$	0.893$_{\pm.010}$	0.953$_{\pm.173}$	1.330$_{\pm.104}$	1.787$_{\pm.029}$	0.822$_{\pm.032}$	0.844$_{\pm.043}$	1.367$_{\pm.083}$
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$_{\pm.055}$
	32	**0.334**$_{\pm.008}$	0.464$_{\pm0.007}$	0.477$_{\pm.035}$	0.669$_{\pm.092}$	6.153$_{\pm.715}$	1.514$_{\pm.042}$	1.922$_{\pm.032}$	0.578$_{\pm.062}$
		0.461$_{\pm.004}$	0.543$_{\pm0.004}$	0.537$_{\pm.019}$	0.646$_{\pm.048}$	1.689$_{\pm.112}$	0.925$_{\pm.016}$	1.068$_{\pm.011}$	0.597$_{\pm.035}$
	64	**0.349**$_{\pm.039}$	0.425$_{\pm.006}$	0.418$_{\pm.021}$	0.484$_{\pm.051}$	2.419$_{\pm.520}$	1.150$_{0.118}$	1.654$_{\pm.117}$	0.463$_{\pm.081}$
		0.473$_{\pm.024}$	0.523$_{0.004}$	0.517$_{\pm.013}$	0.551$_{\pm.027}$	1.223$_{\pm.127}$	0.835$_{\pm.045}$	0.987$_{\pm.036}$	0.546$_{\pm.042}$
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$_{\pm.092}$
	32	**1.224**$_{\pm.060}$	1.784$_{\pm.011}$	1.266$_{\pm.006}$	1.434$_{\pm.126}$	5.398$_{\pm.179}$	13.10$_{\pm.955}$	15.33$_{1.904}$	1.480$_{\pm.072}$
		0.869$_{\pm.074}$	1.200$_{\pm.007}$	0.872$_{\pm.010}$	0.920$_{\pm.077}$	1.434$_{\pm.013}$	1.518$_{\pm.020}$	1.614$_{\pm.118}$	0.987$_{\pm.010}$
	64	0.902$_{\pm.024}$	1.652$_{\pm.010}$	**0.786**$_{\pm.022}$	0.898$_{\pm.095}$	4.403$_{\pm.301}$	3.857$_{\pm.597}$	3.564$_{\pm.293}$	1.374$_{\pm1.02}$
		0.761$_{\pm.021}$	1.167$_{\pm.005}$	**0.742**$_{\pm.017}$	0.789$_{\pm.048}$	1.361$_{\pm.021}$	1.110$_{\pm.143}$	1.152$_{\pm.081}$	0.842$_{\pm.215}$

Table 7: Performance comparisons of TSF in 100%-Electricity and 100%-ETTm1 datasets in terms of MSE and CRPS. The best results are highlighted in boldface.

Model		D^{3}VAE	NVAE	β-TCVAE	f-VAE	DeepAR	TimeGrad	GP-copula	VAE
Electricity	16	**0.330**$_{\pm.033}$	1.408$_{\pm.015}$	0.801$_{\pm.001}$	0.765$_{\pm.026}$	33.93$_{\pm1.85}$	46.69$_{\pm3.13}$	50.25$_{\pm4.39}$	0.680$_{\pm.022}$
		0.445$_{\pm.020}$	0.999$_{\pm.006}$	0.723$_{\pm.001}$	0.710$_{\pm.013}$	2.650$_{\pm.030}$	2.702$_{\pm.079}$	2.796$_{\pm.072}$	0.675$_{\pm.008}$
	32	**0.336**$_{\pm.017}$	1.403$_{\pm.014}$	0.802$_{\pm.001}$	0.748$_{\pm.033}$	46.10$_{\pm2.00}$	30.94$_{\pm1.70}$	32.13$_{\pm1.96}$	0.727$_{\pm.033}$
		0.444$_{\pm.015}$	0.997$_{\pm.007}$	0.724$_{\pm.001}$	0.703$_{\pm.016}$	2.741$_{\pm.011}$	2.476$_{\pm.042}$	2.591$_{\pm.064}$	0.692$_{\pm.014}$
ETTm1	16	**0.018**$_{\pm.002}$	2.577$_{\pm.047}$	0.918$_{\pm.015}$	1.285$_{\pm.236}$	73.82$_{\pm3.25}$	68.26$_{\pm2.04}$	66.97$_{\pm2.02}$	1.335$_{\pm.156}$
		0.102$_{\pm.003}$	1.509$_{0.016}$	0.766$_{\pm.005}$	0.911$_{\pm.090}$	1.136$_{\pm.013}$	1.153$_{\pm.019}$	1.111$_{\pm.016}$	0.923$_{\pm.056}$
	32	**0.034**$_{\pm.001}$	2.622$_{\pm.057}$	0.929$_{\pm.010}$	1.420$_{\pm.073}$	68.11$_{\pm2.60}$	53.47$_{\pm26.1}$	63.67$_{\pm1.14}$	1.223$_{\pm.213}$
		0.144$_{\pm.006}$	1.524$_{\pm.018}$	0.770$_{\pm.004}$	0.960$_{\pm.021}$	1.121$_{\pm.024}$	1.083$_{\pm.109}$	1.097$_{\pm.008}$	0.888$_{\pm.082}$

D Supplementary Experimental Results

D.1 Comparisons of Predictive Performance for TSF Under Different Settings

Longer-Term Time Series Forecasting. To further inspect the performance of our method, we additionally conduct more experiments for longer-term time series forecasting. In particular, by configuring the output length with 32 and 64[1], we compare D^{3}VAE to other baselines in terms of MSE and CRPS, and the results (including short-term and long-term) are reported in Table 6. We can conclude that D^{3}VAE also outperforms the competitive baselines consistently under the longer-term forecasting settings.

Time Series Forecasting in Full Datasets. Moreover, we evaluate the predictive performance for time series forecasting in two "full-version" datasets, i.e. 100%-Electricity and 100%-ETTm1. The split of train/validation/test is 7:1:2 which is the same as the main experiments. The comparisons in terms of MSE and CRPS can be found in Table 7. With sufficient data, compared to previous state-of-the-art generative models, the MSE and CRPS reductions of our method are also satisfactory under different settings (including input-16-predict-16 and input-32-predict-32). For example, in the Electricity dataset, compared to the second best results, D^{3}VAE achieves 52% (0.680 $\rightarrow$ 0.330) and 54% (0.727 $\rightarrow$ 0.336) MSE reductions, and 34% (0.675 $\rightarrow$ 0.445) and 36% (0.692 $\rightarrow$ 0.444) CRPS reductions, under input-16-predict-16 and input-32-predict-32 settings, respectively.

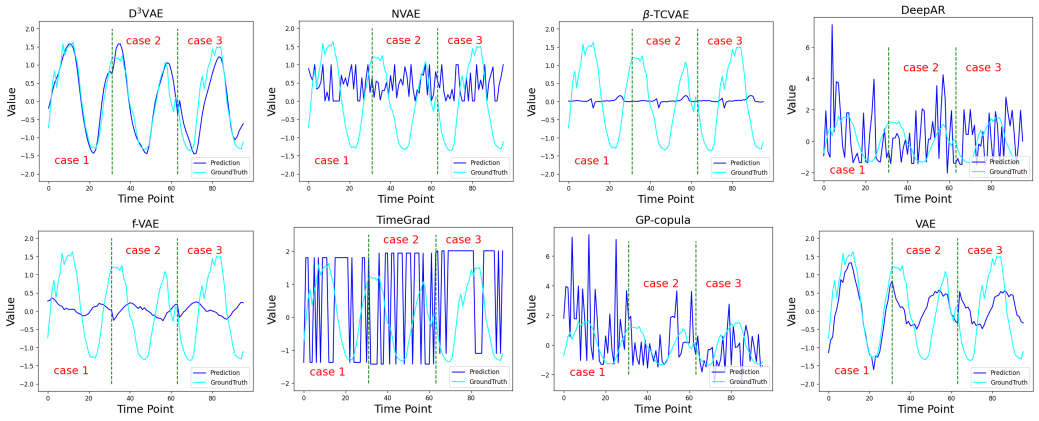


Figure 7: The case study of forecasting results on the Traffic dataset under input-32-predict-32 settings. Only the last dimension is plotted. To demonstrate the forecasting results in a long range, we show the predictions of three cases ordered chronologically without overlapping.

[1]The length of the input time series is the same as the output time series.

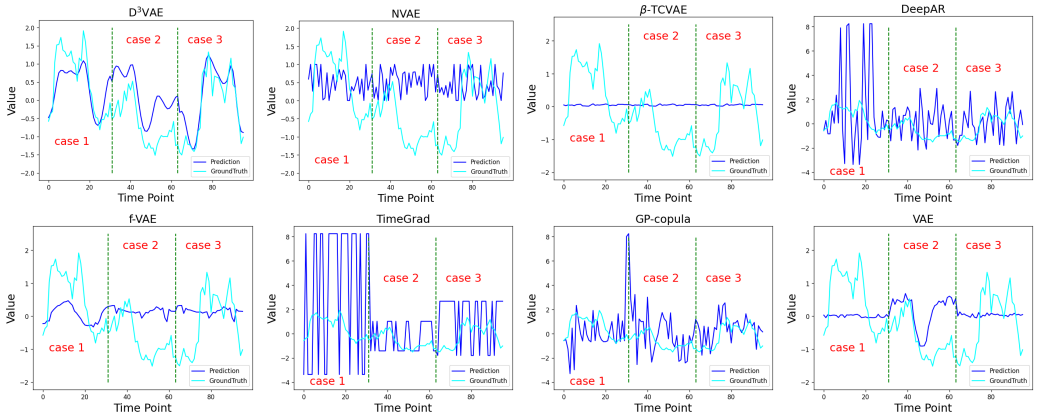


Figure 8: Case study of the forecasting results from the Electricity dataset (same settings as Fig. 7).

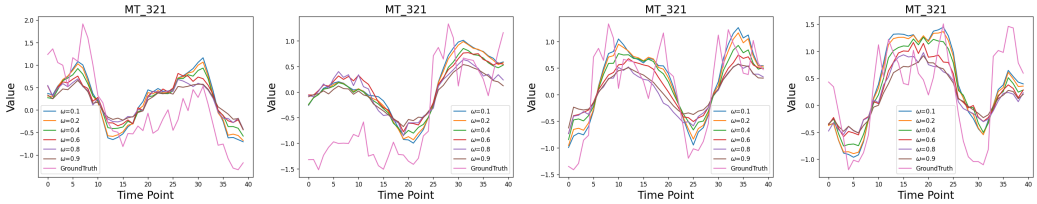


Figure 9: Forecasting results (under the input-40-predict-40 setting) of a case from the Electricity dataset with ω increasing from 0.1 to 0.9.

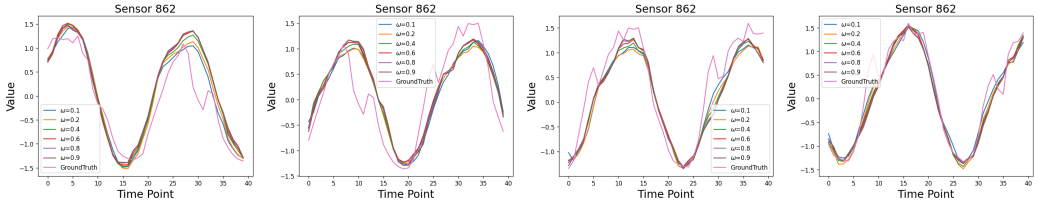


Figure 10: Forecasting results of a case from the Traffic dataset under the input-40-predict-40 setting.

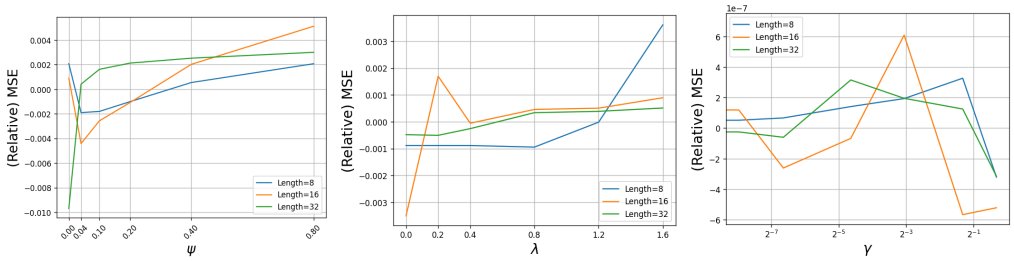


Figure 11: Sensitivity analysis of the trade-off hyperparameters in reconstruction loss $\mathcal{L}$. To highlight the changes in prediction performance against hyperparameters, the relative value of MSE is used.

D.2 Case Study

We showcase the prediction results of our model and seven baseline models on the Traffic and Electricity datasets in Figs. 7 and 8. Our model can provide the most accurate forecasting results regarding trends and variations.

D.3 The Effect of Scale Parameter ω

We demonstrate the forecasting results with different values of ω on Electricity and Traffic datasets, and the results are plotted in Figs. 9 and 10. It can be depicted that larger or smaller ω would lead

Table 8: Performance comparisons of D^{3}VAE w.r.t. varying the length of (input and output) time series and the data size. The results are reported on the Electricity dataset.

Length	Metric	Percentage of Full Electricity Data						
		100%	80%	60%	40%	20%	10%	5%
8	MSE	$0.258_{\pm.019}$	$0.227_{\pm.016}$	$0.368_{\pm.019}$	$0.389_{\pm.034}$	$3.861_{\pm.480}$	$0.693_{\pm.223}$	$0.206_{\pm.018}$
	CRPS	$0.383_{\pm.015}$	$0.355_{\pm.015}$	$0.453_{\pm.009}$	$0.504_{\pm.034}$	$1.728_{\pm.110}$	$0.673_{\pm.132}$	$0.352_{\pm.016}$
16	MSE	$0.330_{\pm.033}$	$0.253_{\pm.018}$	$0.343_{\pm.024}$	$0.463_{\pm.089}$	$4.428_{\pm.694}$	$0.401_{\pm.068}$	$0.247_{\pm.056}$
	CRPS	$0.445_{\pm.020}$	$0.373_{\pm.014}$	$0.433_{\pm.015}$	$0.562_{\pm.049}$	$1.858_{\pm.147}$	$0.496_{\pm.047}$	$0.378_{\pm.036}$
32	MSE	$0.336_{\pm.017}$	$0.300_{\pm.039}$	$0.484_{\pm.048}$	$0.739_{\pm.209}$	$5.029_{\pm.811}$	$0.884_{\pm.237}$	$0.304_{\pm.094}$
	CRPS	$0.444_{\pm.015}$	$0.413_{\pm.034}$	$0.537_{\pm.025}$	$0.693_{\pm.099}$	$1.989_{\pm.172}$	$0.723_{\pm.112}$	$0.418_{\pm.065}$

Table 9: Performance comparisons of D^{3}VAE w.r.t. varying the length of (input and output) time series and the data size. The results are reported on the Traffic dataset.

Length	Metric	Percentage of Full Traffic Data						
		100%	80%	60%	40%	20%	10%	5%
8	MSE	$0.370_{\pm.021}$	$0.215_{\pm.016}$	$0.063_{\pm.002}$	$0.062_{\pm.002}$	$0.054_{\pm.004}$	$0.210_{\pm.012}$	$0.081_{\pm.003}$
	CRPS	$0.415_{\pm.013}$	$0.347_{\pm.015}$	$0.184_{\pm.003}$	$0.179_{\pm.005}$	$0.172_{\pm.008}$	$0.251_{\pm.005}$	$0.207_{\pm.003}$
16	MSE	$0.272_{\pm.007}$	$0.189_{\pm.006}$	$0.063_{\pm.001}$	$0.058_{\pm.003}$	$0.056_{\pm.003}$	$0.178_{\pm.006}$	$0.081_{\pm.009}$
	CRPS	$0.334_{\pm.009}$	$0.321_{\pm.008}$	$0.180_{\pm.002}$	$0.168_{\pm.006}$	$0.169_{\pm.005}$	$0.239_{\pm.007}$	$0.200_{\pm.003}$
32	MSE	$0.307_{\pm.015}$	$0.197_{\pm.005}$	$0.064_{\pm.002}$	$0.063_{\pm.002}$	$0.056_{\pm.004}$	$0.191_{\pm.011}$	$0.091_{\pm.007}$
	CRPS	$0.363_{\pm.008}$	$0.335_{\pm.004}$	$0.179_{\pm.002}$	$0.179_{\pm.003}$	$0.170_{\pm.005}$	$0.235_{\pm.008}$	$0.216_{\pm.012}$

to deviated prediction, which is far from the ground truth. Therefore, the value of ω does affect the prediction performance, which should be tuned properly.

D.4 Sensitivity Analysis of Trade-off Parameters in Reconstruction Loss $\mathcal{L}$

To examine the effect of the trade-off hyperparameters in loss $\mathcal{L}$, we plot the mean square error (MSE) against different values of trade-off parameters, i.e., ψ, λ and γ, in the Traffic dataset. Note that the relative value of MSE is plotted to ensure the difference is distinguishable. This experiment is conducted under different settings: input-8-predict-8, input-16-predict-16, and input-32-predict-32. For ψ, the value ranges from 0 to 0.8, λ ranges from 0 to 1.6, and γ ranges from 0 to 0.5. The results are shown in Fig. 11. We can see that the model's performance varies slightly as the trade-off parameters take different values, which shows that our model is robust enough against different trade-off parameters.

D.5 Scalability Analysis of Varying Time Series Length and Dataset Size

We additionally investigate the scalability of D^{3}VAE against different lengths of the time series and varying amounts of available data. The experiments are conducted on the Electricity and Traffic datasets, and the results are reported in Tables 8 and 9, respectively. We can observe that the predictive performance of D^{3}VAE is relatively stable under different settings. In particular, the longer the target series to predict, the worse performance might be obtained. Besides, when the amount of available data is shrunk, D^{3}VAE performs more stable than expected. Note that on the 20%-Electricity dataset, the performance of D^{3}VAE is much worse than other subsets of the Electricity dataset, mainly because the sliced 20%-Electricity dataset involves more irregular values.

E Disentanglement for Time Series Forecasting

Fig. 13 illustrates the disentanglement of latent variable Z for time series forecasting. It is difficult to choose suitable disentanglement factors under the unsupervised learning of disentanglement.

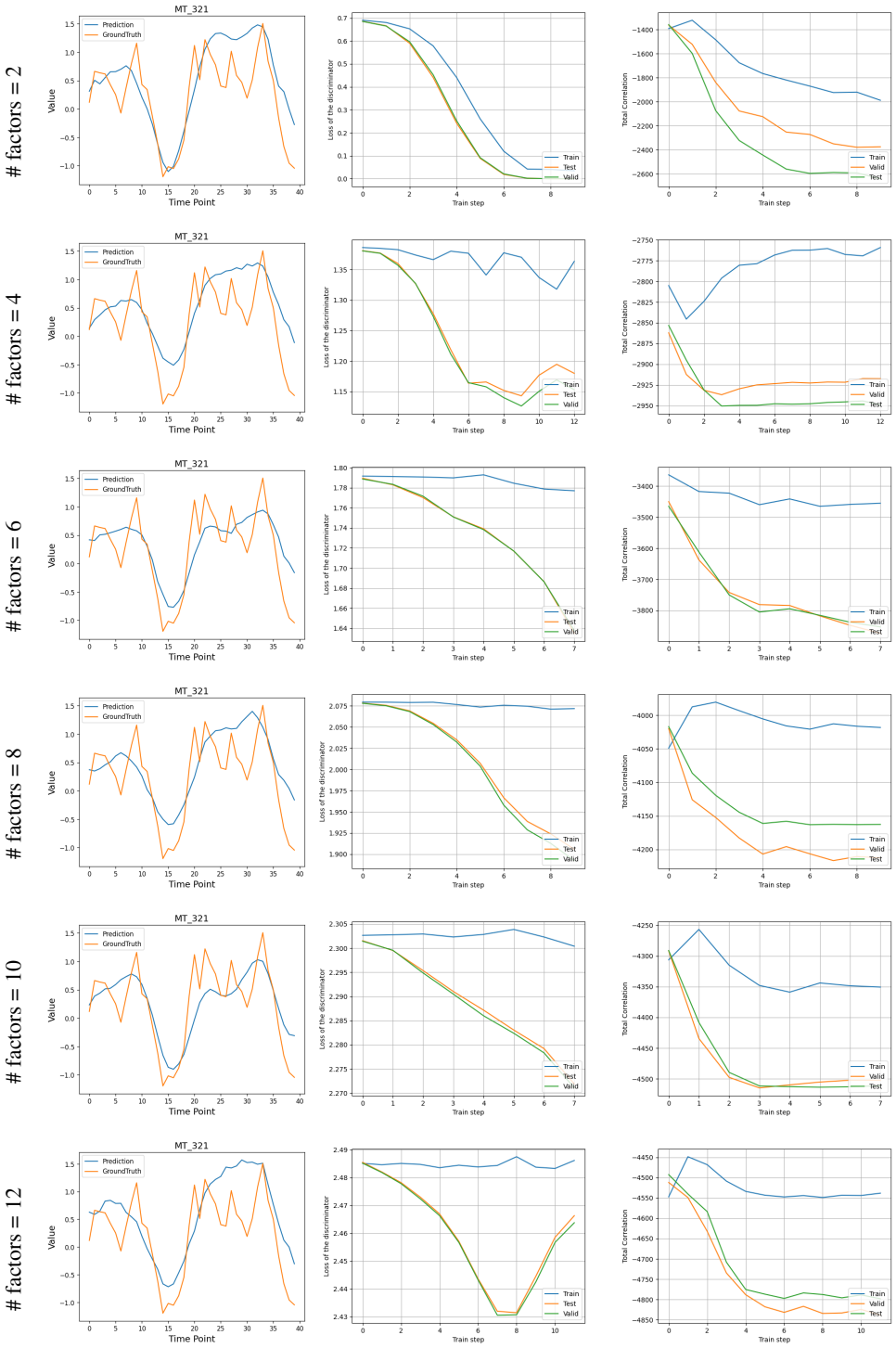


Figure 12: We showcase an instance from the Electricity dataset and demonstrate the results when different numbers of factors in disentanglement are adopted. For each row, from left to right, the prediction result of TSF, the learning curve of the discriminator, and the total correlation are plotted, respectively.

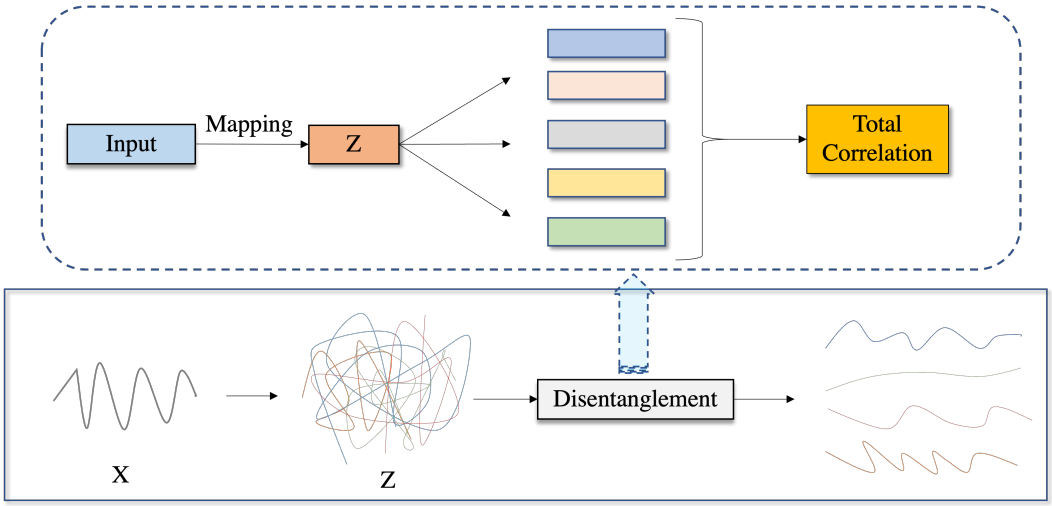


Figure 13: Disentangling latent variable Z of time series. Specifically, the input X is first mapped into Z. Then $\forall z_i \in Z$ is decomposed as $z_i = [z_{i,1}, \cdots, z_{i,m}]$ and the metric of total correlation is utilized to minimize the inter-dependencies among "hand-crafted" factors. In this way, the disentangled factors tend to be not only discriminative but also informative.

Algorithm 3 Train a discriminator for time series disentanglement.

1: **repeat**
2: Initialize the loss of a discriminator $\mathcal{D}_\varphi$: $L(\mathcal{D}_\varphi) = 0$
3: Decompose the latent variable generated in Algorithm 1 as $z_i = [z_{i,1}, z_{i,2}, ..., z_{i,m}]$ $(i = 1, \cdots, n)$
4: **for** z_i in Z **do**
5: $L = L + \sum_{j=1}^{m} \|\mathcal{D}_\varphi(z_{i,j}) - j\|^2$
6: **end for**
7: Optimize the discriminator: $\varphi \leftarrow \text{argmin}(L)$
8: **until** Convergence

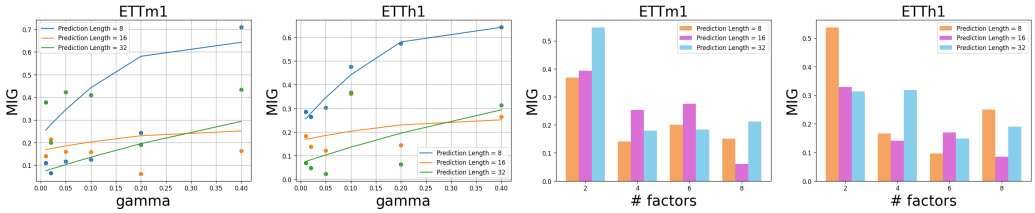
Therefore, we attempt to inspect the TSF performance against different numbers of factors to be disentangled. We implement a simple classifier as a discriminator to further evaluate the disentanglement quality in Fig. 12 (and Algorithm 3 demonstrates the training procedure of the discriminator). To be specific, we take different dimensions of Z as the factors to be disentangled: $z_i = [z_{i,1}, \cdots, z_{i,m}]$ $(z_i \in Z)$, then an instance consisting of factor and label $(z_{i,j}, j)$ is constructed. We shuffle these m examples for each z_i and attempt to classify them with a discriminator, then the disentanglement can be evaluated by measuring the loss of the discriminator. The learning curve of the discriminator can be leveraged to assess the disentanglement, and the discriminator is implemented by an MLP with six nonlinear layers and 100 hidden states. The results of prediction, discriminator loss, and the total correlation w.r.t. different numbers of factors are plotted in Fig. 12, respectively. As shown in Fig. 12, the number of factors does affect the prediction performance, as well as the disentanglement quality. On the other hand, the learning curves can be converged when different factors are adopted, which validates that the disentanglement of the latent factors is of high quality.

In addition to the above method evaluating the disentanglement indirectly, we adopt another metric named Mutual Information Gap (MIG) [11] to evaluate the quality of disentanglement in a more straightforward way. Specifically, for a latent variable $z_i \in Z$, the mutual information between $z_{i,j}$, and a factor $v_k \in [1, m]$ can be calculated by

$$I_d(z_{i,j}, v_k) = \mathbb{E}_{q(z_{i,j}, v_k)}[log \sum_{d \in \mathcal{S}_{v_k}} q(z_{i,j}|d)p(d|v_k)] + H(z_{i,j}), \tag{16}$$

where d denotes the sample of $z_{i,j}$ and $\mathcal{S}_{v_k}$ is the support set of v_k. Then, for $z_{i,j}$

$$MIG(z_{i,j}) = \frac{1}{m} \sum_{1}^{m} \frac{1}{H(v_k)} (max(I_d(z_{i,j}, v_k)) - submax(I_d(z_{i,j}, v_k))), \tag{17}$$



(a) (b) (c) (d)

Figure 14: We evaluate the quality of disentanglement on ETTm1 and ETTh1 datasets regarding the mutual information gap (MIG). **(a-b)** The scatter plots of the MIG against varying weights γ in loss function (refer to Eq. (14) in the main text). **(c-d)** MIG v.s. different numbers of factors.

where $submax$ means the second max value of $I_d(z_{i,j}, v_k)$, then the MIG of Z can be obtained as

$$MIG(Z) = \sum_{i=1}^{n} MIG(z_i), \quad MIG(z_i) = \frac{1}{m} \sum_{j=1}^{m} MIG(z_{i,j}) \,. \tag{18}$$

We evaluate the quality of disentanglement in terms of MIG on ETTm1 and ETTh1 datasets, respectively, which can be seen in Fig. 14. From Figs. 14a and 14b, when the weight of disentanglement (i.e., γ in Eq. (14) of the main text) grows, the disentangled factors are of higher quality. In other words, the latent variables can be disentangled with the help of the disentanglement module in D^{3}VAE. In addition, we examine the changes in MIG against different numbers of factors. We can observe that the difficulty of disentanglement climbs up as the number of factors increases.

F Model Inspection: Coupled Diffusion Process

To gain more insights into the coupled diffusion process, we demonstrate how a time series can be diffused under different settings in terms of variance schedule $\boldsymbol{\beta}$ and the max number of diffusion steps T. The examples are illustrated in Fig. 15. It can be seen that when larger diffusion steps or a wider variance schedule is employed, the diffused series deviates far from the original data gradually, which may result in the loss of useful signals, like, temporal dependencies. Therefore, it is important to choose a suitable variance schedule and diffusion steps to ensure that the distribution space is deviated enough without losing useful signals.

G Necessity of Data Augmentation for Time Series Forecasting

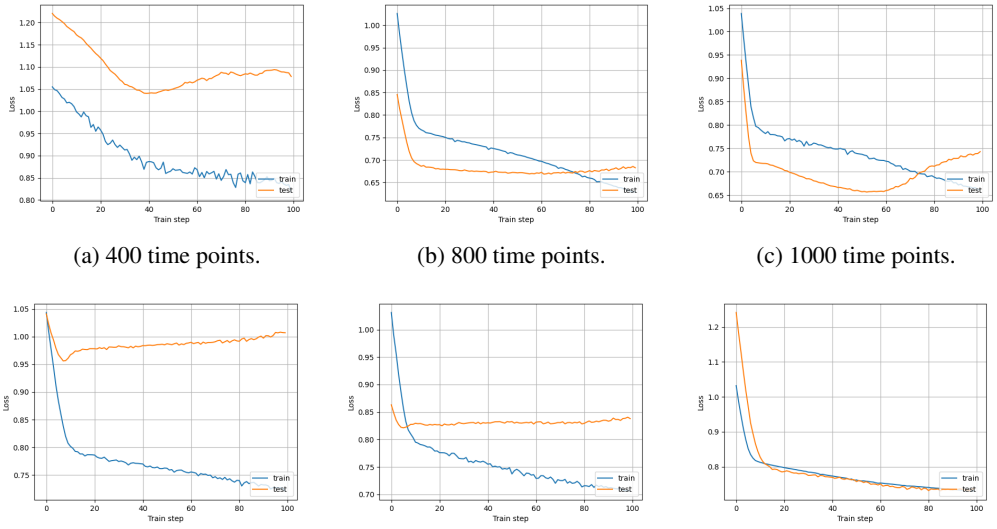
Limited data would result in overfitting and poor performance. To demonstrate the necessity of enlarging the size of data for time series forecasting when deep models are employed, we implement a two-layer RNN and evaluate how many time points are required to ensure the generalization ability. A synthetic dataset is adopted for this demonstration.

According to [17], we generate a toy time series dataset with n time points in which each point is a d-dimension variable:

$$w_t = 0.5w_{t-1} + \tanh(0.5w_{t-2}) + \sin(w_{t-3}) + \epsilon \,, \quad X = [w_1, w_2, ..., w_n] * F + v$$

where $w_t \in \mathcal{R}^2$, $F \in \mathcal{R}^{2\times d} \sim \mathcal{U}[-1, 1]$, $\epsilon \sim \mathcal{N}(0, I)$, $v \sim \mathcal{N}(0, 0.5I)$, and $d = 5$. An input-8-predict-8 window is utilized to roll this synthetic dataset. We split this synthetic dataset into training and test sets with a ratio of 7:3. We train the RNN in 100 epochs at most, and the MSE loss of training and testing are plotted in Fig. 16. It can be seen that the inflection points of the loss curves move back gradually and disappear as increasing the size of the dataset. Besides, with fewer time points, like, 400, the model can be overfitted more easily.

Figure 15: Diffused time series with different variance schedules and diffusion steps. We randomly choose a sample series from the synthetic dataset D2 and plot the original time series data, as well as the diffused series.



(a) 400 time points. (b) 800 time points. (c) 1000 time points.

(d) 1200 time points. (e) 1400 time points. (f) 1600 time points.

Figure 16: The curves of training and testing losses when the available time series data are of different sizes.

References

[1] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

[2] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. Stock price prediction using the ARIMA model. In *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 106–112. IEEE, 2014.

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[4] Mauricio Barahona and Chi-Sang Poon. Detection of nonlinear dynamics in short, noisy time series. *Nature*, 381(6579):215–217, 1996.

[5] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *STAT*, 1050:16, 2017.

[6] George EP Box and Gwilym M Jenkins. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

[7] Sofiane Brahim-Belhouari and Amine Bermak. Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, 47(4):705–712, 2004.

[8] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, 33:17766–17778, 2020.

[9] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Sotirios P Chatzis. Recurrent latent variable conditional heteroscedasticity. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2711–2715. IEEE, 2017.

[11] 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.

[12] 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, 2015.

[13] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[14] Emmanuel de Bézenac, Syama Sundar Rangapuram, Konstantinos Benidis, Michael Bohlke-Schneider, Richard Kurle, Lorenzo Stella, Hilaf Hasson, Patrick Gallinari, and Tim Januschowski. Normalizing kalman filters for multivariate time series analysis. *Advances in Neural Information Processing Systems*, 33:2995–3007, 2020.

[15] Ankur Debnath, Govind Waghmare, Hardik Wadhwa, Siddhartha Asthana, and Ankur Arora. Exploring generative data augmentation in multivariate time series forecasting: Opportunities and challenges. *Solar-Energy*, 137:52–560, 2021.

[16] Stephen Ellner and Peter Turchin. Chaos in a noisy world: New methods and evidence from time-series analysis. *The American Naturalist*, 145(3):343–375, 1995.

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

[18] Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. Generating synthetic time series to augment sparse datasets. In *IEEE International Conference on Data Mining*, pages 865–870. IEEE, 2017.

[19] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. SOM-VAE: Interpretable discrete representation learning on time series. In *International Conference on Learning Representations*, 2019.

[20] WR Foster, F Collopy, and LH Ungar. Neural network forecasting of short, noisy time series. *Computers & Chemical Engineering*, 16(4):293–297, 1992.

[21] 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*, 30, 2017.

[22] C Lee Giles, Steve Lawrence, and Ah Chung Tsoi. Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning*, 44(1):161–183, 2001.

[23] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *arXiv preprint arXiv:2008.12595*, 2020.

[24] 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.

[25] Vinayaka Gude, Steven Corns, and Suzanna Long. Flood prediction and uncertainty estimation using deep learning. *Water*, 12(3):884, 2020.

[26] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.

[27] 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.

[28] Steven Craig Hillmer and George C Tiao. An ARIMA-model-based approach to seasonal adjustment. *Journal of the American Statistical Association*, 77(377):63–70, 1982.

[29] Siu Lau Ho and Min Xie. The use of ARIMA models for reliability forecasting and analysis. *Computers & Industrial Engineering*, 35(1-2):213–216, 1998.

[30] 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.

[31] Aya Abdelsalam Ismail, Mohamed Gunady, Luiz Pessoa, Hector Corrada Bravo, and Soheil Feizi. Input-cell attention reduces vanishing saliency of recurrent neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[32] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[33] Dulakshi SK Karunasinghe and Shie-Yui Liong. Chaotic time series prediction with a global model: Artificial neural network. *Journal of Hydrology*, 323(1-4):92–105, 2006.

[34] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[35] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018.

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

[37] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[39] Naoto Kunitomo and Seisho Sato. A robust-filtering method for noisy non-stationary multivariate time series with econometric applications. *Japanese Journal of Statistics and Data Science*, 4(1):373–410, 2021.

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

[41] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[42] 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.

[43] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

[44] 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.

[45] Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2):115–125, 2009.

[46] Danielle C Maddix, Yuyang Wang, and Alex Smola. Deep factors with Gaussian processes for forecasting. *arXiv preprint arXiv:1812.00098*, 2018.

[47] Nam Nguyen and Brian Quanz. Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 9117–9125, 2021.

[48] Fotios Petropoulos, Rob J Hyndman, and Christoph Bergmeir. Exploring the sources of uncertainty: Why does bagging for time series forecasting work? *European Journal of Operational Research*, 268(2):545–554, 2018.

[49] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *International Joint Conference on Artificial Intelligence*, 2017.

[50] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems*, 31, 2018.

[51] 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.

[52] 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.

[53] 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.

[54] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in Neural Information Processing Systems*, 32, 2019.

[55] 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 *AAAI Conference on Artificial Intelligence*, volume 34, pages 5758–5766, 2020.

[56] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8):1421–1441, 2019.

[57] Sameer Singh. Noisy time-series prediction using pattern recognition techniques. *Computational Intelligence*, 16(1):114–133, 2000.

[58] Slawek Smyl and Karthik Kuber. Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks. In *International Symposium on Forecasting*, 2016.

[59] Huan Song, Deepta Rajan, Jayaraman J Thiagarajan, and Andreas Spanias. Attend and diagnose: Clinical time series analysis using attention models. In *AAAI Conference on Artificial Intelligence*, 2018.

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

[61] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

[62] Mike West. Bayesian forecasting of multivariate time series: scalability, structure uncertainty and decisions. *Annals of the Institute of Statistical Mathematics*, 72(1):1–31, 2020.

[63] Bingzhe Wu, Jintang Li, Chengbin Hou, Guoji Fu, Yatao Bian, Liang Chen, and Junzhou Huang. Recent advances in reliable deep graph learning: Adversarial attack, inherent noise, and distribution shift. *arXiv preprint arXiv:2202.07114*, 2022.

[64] Mike Wu, Michael Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[65] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020.

[66] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence*, pages 1907–1913, 2019.

[67] 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.

[68] Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. ScoreGrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *arXiv preprint arXiv:2106.10121*, 2021.

[69] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[70] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[71] 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 *AAAI Conference on Artificial Intelligence*, 2021.