EAST CHINA NORMAL UNIVERSITY

Paper Code

My **WeChat:**

K²VAE: A Koopman-Kalman Enhanced Variational AutoEncoder for Probabilistic Time Series Forecasting

Xingjian Wu[1*], Xiangfei Qiu[1*], Hongfan Gao[1], Jilin Hu[1], Chenjuan Guo[1], Bin Yang[1✉]

{xjwu,xfqiu,hfgao}@stu.ecnu.edu.cn, {jlhu,cjguo,byang}@dase.ecnu.edu.cn

[1]East China Normal University

ICML International Conference On Machine Learning

Introduction

Existing Generative Probabilistic Time Series Forecasting methods achieve promising performance but still suffer from the inherent nonlinearity and error accumulation as the forecasting horizon extends.

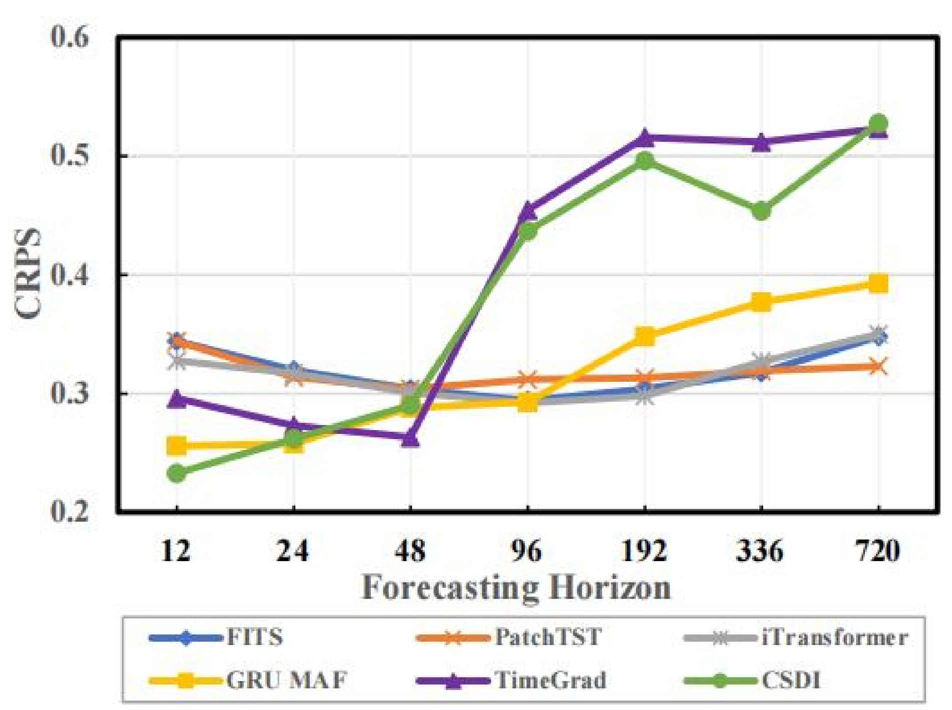

Figure 1: PTSF performance w.r.t. the forecasting horizon

As shown in Figure 1, we compare three native probabilistic forecasting models including GRU MAF, TimeGrad, and CSDI with three point forecasting models equipped with distributional heads including FITS, PatchTST, and iTransformer on ETTh1. Longer forecasting horizons lead to rapid collapse of the CRPS metric (lower is better) on probabilistic forecasting models, even worse than point forecasting models.

Contribution

- To address PTSF, we propose an efficient framework called K²VAE. It transforms nonlinear time series into a linear dynamical system. Through predicting and refining the process uncertainty of the system, K²VAE demonstrates strong generative capability and excells in both the short- and long-term probabilistic forecasting.
- To distengle the complex nonlinearity in the time series, we design a KoopmanNet to fully exploit the underlying linear dynamical characteristics in the space of measurement function, simplify the modeling, and thus contributing to high model efficiency.
- To mitigate the error accumulation in LPTSF, we devise a KalmanNet to model, and refine the prediction and uncertainty iteratively.
- Comprehensive experiments on both short- and long-term PTSF show that K²VAE outperforms state-of-the-art baselines.

Data Flow

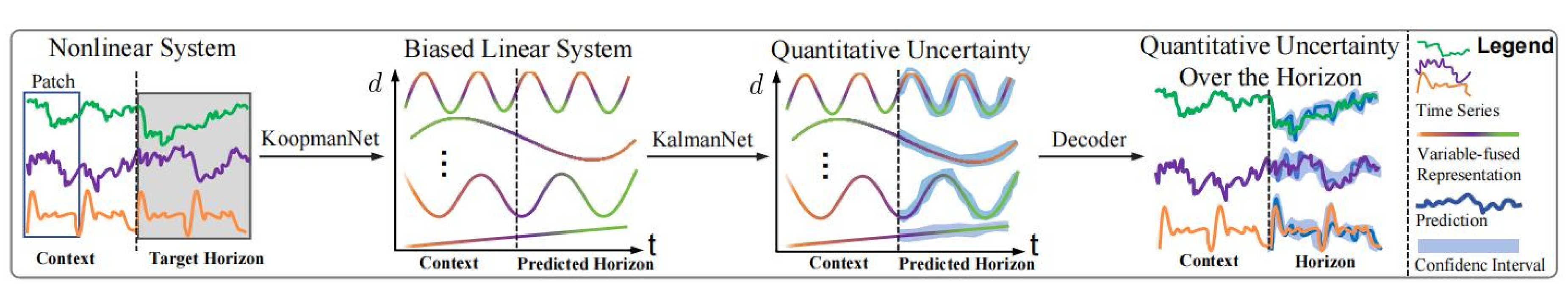

Figure 2: The data flow of K²VAE

K²VAE Framework

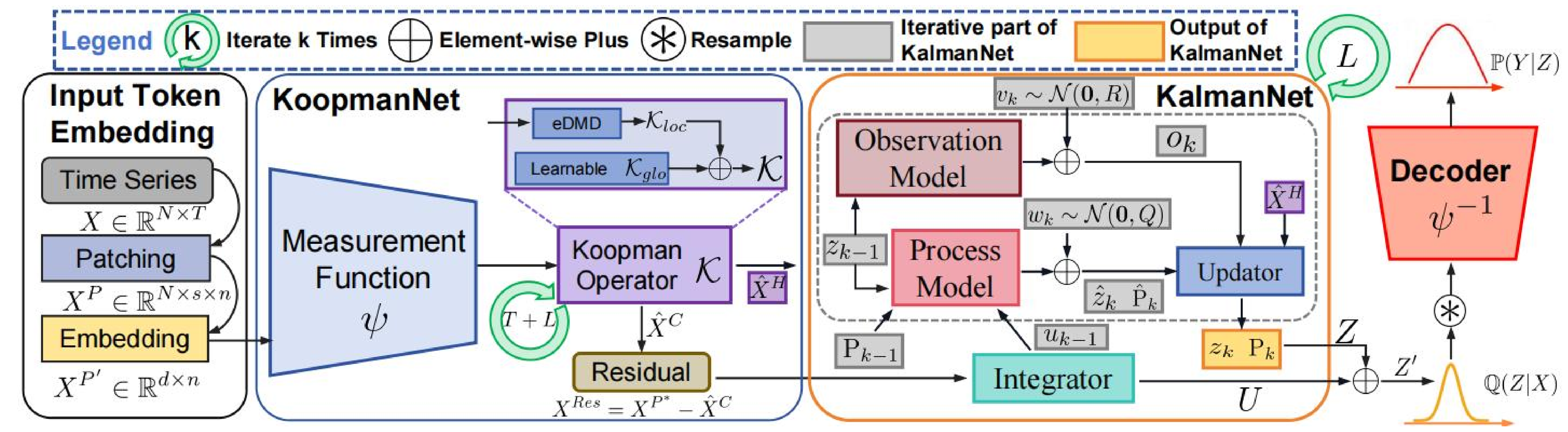

Figure 3: The framework of K²VAE

Input Token Embedding

K²VAE works like an autoregressive dynamic system to model the state transition procedure. Different from those Channel-Independent models which divides patches for each channel and projects them independently, we consider multivariate patches as tokens to implicitly model the cross-variable interaction during state transition. We divide the context series into non-overlapping patches:

$$X^P = [x_1^P, x_2^P, \cdots, x_n^P] \in \mathbb{R}^{N\times s\times n},\ X^{P'} = \text{Projection}(\text{Flatten}(X^P))$$

KoopmanNet

K²VAE then applies Koopman Theory to construct the measurement function to project the system states into measurements which can be modeled as a linear system. Practically, we use a learnable MLP-based network to serve as the measurement function ψ and then use the one-step eDMD to model the linear system:

$$X^{P^*} = \psi(X^{P'}) = \left[x_1^{P^*}, x_2^{P^*}, \cdots, x_n^{P^*}\right], X_{back}^{P^*} = \left[x_1^{P^*}, x_2^{P^*}, \cdots, x_{n-1}^{P^*}\right],\ X_{fore}^{P^*} = \left[x_2^{P^*}, x_3^{P^*}, \cdots, x_n^{P^*}\right],$$

$$\mathcal{K}_{loc} = X_{fore}^{P^*}(X_{back}^{P^*})^{\dagger},\ \hat{X}^C = [\hat{x}_1^C, \hat{x}_2^C, \cdots, \hat{x}_n^C],\ \hat{X}^H = [\hat{x}_1^H, \hat{x}_2^H, \cdots, \hat{x}_m^H],$$

$$\mathcal{K} = \mathcal{K}_{loc} + \mathcal{K}_{glo},\ \hat{x}_i^C = (\mathcal{K})^{i-1}x_1^{P^*},\ \hat{x}_i^H = (\mathcal{K})^{i+n-1}x_1^{P^*}$$

KalmanNet

Since we adopt a data-driven paradigm to model the measurement function ψ and Koopman Operator $\mathcal{K}$, it exists bias between the generated $\hat{X}^C$ and X^{P^*} during optimization, known as a biased linear system. We then devise a KalmanNet to model and refine the uncertainty using Kalman Gain, and align it with the variational distribution in the latent measurement space:

$$X^{Res} = X^{P^*} - \hat{X}^C,\ U = \text{Integrator}(X^{Res}) = [u_1, u_2, \cdots, u_m],$$

Predict Step: $\hat{z}_k = Az_{k-1} + Bu_k,\ \hat{\mathrm{P}}_k = A\mathrm{P}_{k-1}A^T + Q,$

Update Step: $K_k = \hat{\mathrm{P}}_k H^T(H\hat{\mathrm{P}}_k H^T + R)^{-1},\ z_k = \hat{z}_k + K_k(\hat{x}_k^H - H\hat{z}_k),\ \mathrm{P}_k = (I - K_k H)\hat{\mathrm{P}}_k$

Decoder

To fully utilize the ability of the Integrator, we make a skip connection: $Z' = Z + U$, then the variational distribution is obtained through $\mathbb{Q}(Z|X) = \mathcal{N}(Z', \mathrm{P})$. Finally, we utilize the Decoder to map the samples back to the original space and model the $\mathbb{P}(Y|Z)$:

$$Z^{sample} = \text{Resample}(\mathbb{Q}(Z|X)),\ \mu = \psi_\mu^{-1}(Z^{sample}),\ \sigma = \psi_\sigma^{-1}(Z^{sample}),\ \mathbb{P}(Y|Z) = \mathcal{N}(\mu, \sigma)$$

Experiments

Main Results

Table 1: Results of Long-term Probabilistic Time Series Forecasting

Model	Metric	ETTm1-L	ETTm2-L	ETTh1-L	ETTh2-L	Electricity-L	Traffic-L	Weather-L	Exchange-L	ILI-L
FITS	CRPS	0.305±0.024	0.449±0.034	0.348±0.025	0.314±0.022	0.115±0.024	0.374±0.004	0.267±0.003	0.074±0.011	0.211±0.011
	NMAE	0.406±0.072	0.540±0.052	0.468±0.012	0.401±0.022	0.149±0.012	0.453±0.022	0.317±0.021	0.097±0.011	0.245±0.017
PatchTST	CRPS	0.304±0.029	0.229±0.036	0.323±0.020	0.304±0.018	0.127±0.015	0.214±0.001	0.142±0.005	0.097±0.007	0.233±0.019
	NMAE	0.382±0.066	0.288±0.034	0.428±0.024	0.371±0.021	0.164±0.024	0.253±0.012	0.152±0.029	0.126±0.001	0.287±0.023
iTransformer	CRPS	0.455±0.021	0.311±0.024	0.350±0.019	0.542±0.015	0.109±0.044	0.284±0.004	0.133±0.004	0.087±0.023	0.222±0.020
	NMAE	0.490±0.038	0.385±0.042	0.449±0.022	0.667±0.012	0.140±0.009	0.361±0.030	0.147±0.019	0.113±0.015	0.278±0.017
Koopa	CRPS	0.295±0.027	0.233±0.025	0.318±0.009	0.293±0.026	0.113±0.018	0.358±0.022	0.140±0.007	0.091±0.012	0.228±0.022
	NMAE	0.377±0.037	0.290±0.033	0.412±0.008	0.286±0.042	0.149±0.025	0.432±0.032	0.162±0.009	0.116±0.022	0.288±0.031
TSDiff	CRPS	0.478±0.027	0.344±0.046	0.516±0.027	0.406±0.056	0.478±0.005	0.391±0.002	0.152±0.003	0.082±0.010	0.263±0.022
	NMAE	0.622±0.045	0.416±0.065	0.657±0.017	0.482±0.022	0.622±0.142	0.478±0.006	0.141±0.026	0.142±0.009	0.272±0.020
GRU NVP	CRPS	0.546±0.036	0.561±0.273	0.502±0.039	0.539±0.090	0.114±0.013	0.211±0.004	0.110±0.004	0.079±0.009	0.307±0.005
	NMAE	0.707±0.050	0.749±0.385	0.643±0.046	0.688±0.161	0.144±0.017	0.264±0.006	0.135±0.008	0.103±0.009	0.333±0.005
GRU MAF	CRPS	0.536±0.033	0.272±0.029	0.393±0.043	0.990±0.023	0.106±0.007	-	0.122±0.006	0.160±0.019	0.172±0.034
	NMAE	0.711±0.081	0.355±0.048	0.496±0.019	1.092±0.019	0.136±0.098	-	0.149±0.034	0.182±0.010	0.216±0.014
Trans MAF	CRPS	0.688±0.043	0.355±0.043	0.363±0.053	0.327±0.033	-	-	0.113±0.004	0.148±0.017	0.155±0.018
	NMAE	0.822±0.034	0.475±0.029	0.455±0.025	0.412±0.020	-	-	0.148±0.040	0.191±0.006	0.183±0.019
TimeGrad	CRPS	0.621±0.037	0.470±0.054	0.523±0.027	0.445±0.016	0.108±0.003	0.220±0.002	0.113±0.011	0.099±0.015	0.295±0.083
	NMAE	0.793±0.034	0.561±0.044	0.672±0.015	0.550±0.018	0.134±0.004	0.263±0.001	0.136±0.020	0.113±0.016	0.325±0.068
CSDI	CRPS	0.448±0.038	0.239±0.035	0.528±0.012	0.302±0.040	-	-	0.087±0.003	0.143±0.020	0.283±0.012
	NMAE	0.578±0.051	0.306±0.040	0.657±0.014	0.382±0.030	-	-	0.102±0.005	0.173±0.020	0.299±0.013
K^2VAE	CRPS	**0.294**±0.026	**0.221**±0.023	**0.314**±0.011	**0.280**±0.014	**0.057**±0.005	**0.200**±0.001	**0.084**±0.003	**0.069**±0.005	**0.142**±0.008
	NMAE	**0.373**±0.032	**0.275**±0.035	**0.396**±0.012	**0.278**±0.020	**0.117**±0.019	**0.248**±0.010	**0.099**±0.009	**0.084**±0.017	**0.167**±0.007

Due to the excessive time and memory consumption, some results are unavailable and denoted as -.

- K²VAE consistly achieves state-of-the-art performance on 9 LPTSF and 8 SPTSF scenarios.
- Resources: https://github.com/decisionintelligence/k2vae

Efficiency Analysis

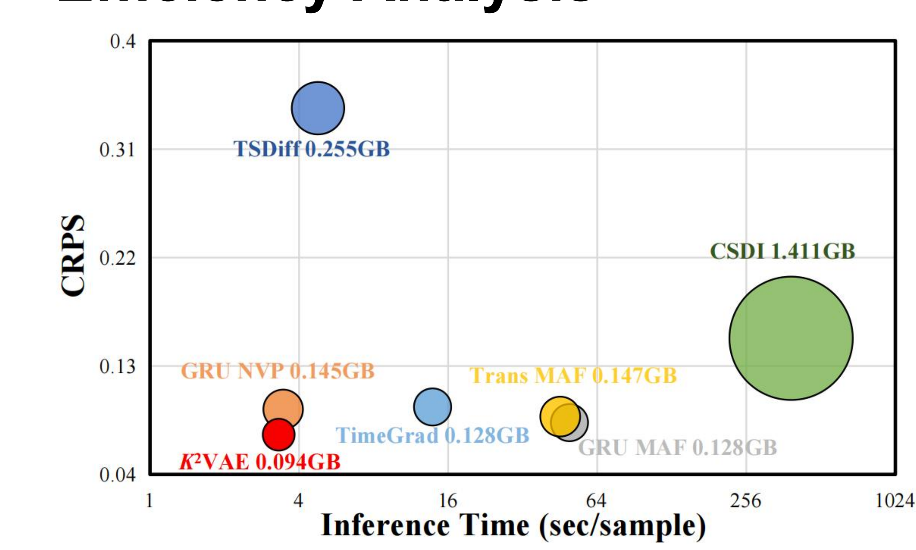

Figure 4: Efficiency Analysis of K²VAE

- K²VAE is the most lightweight generative probabilistic time series forecasting model.
- Compared with diffusion-based and flow-based models, K²VAE has the fastest inference speed.

Visualization

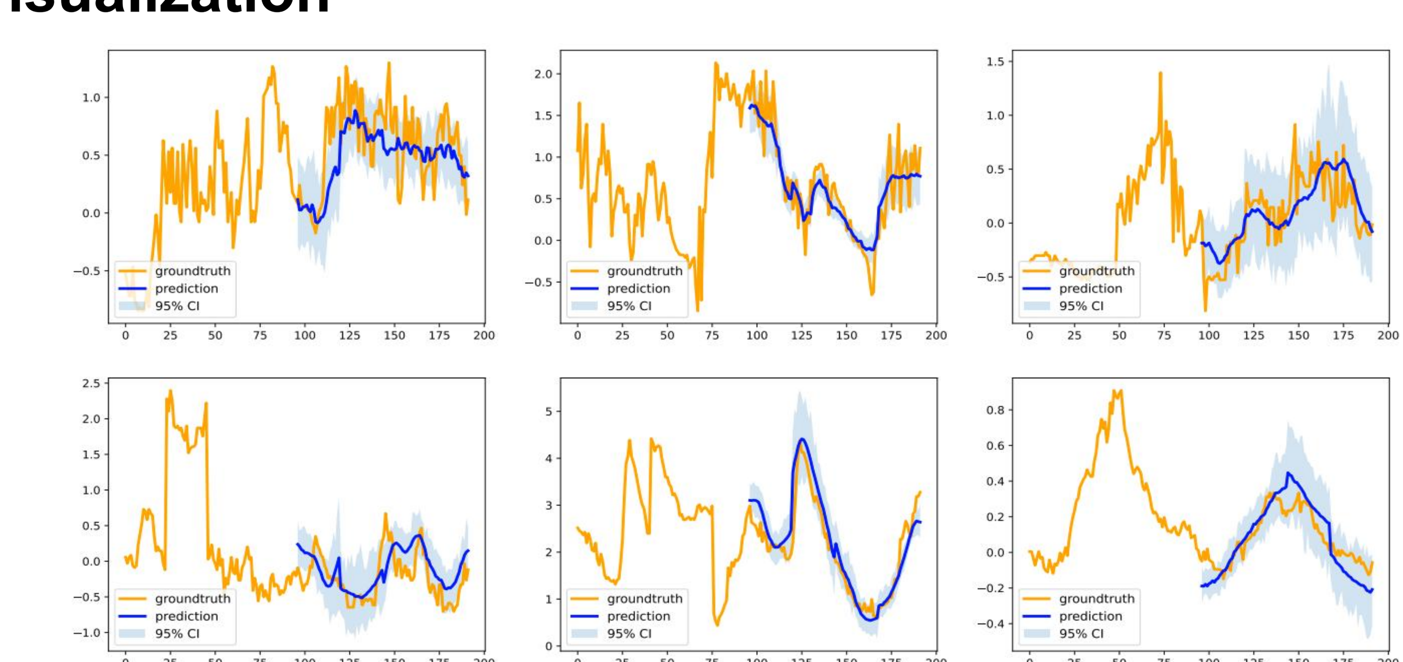

Figure 5: Visualization of input-96-predict-96 results on the ETTm1-L dataset