K-ProtoDiff: Key Prototypes-Guided Diffusion for Time Series Generation

Yuhang Duan, Lin Lin*, Xiaoshuai Wu

School of Software Technology, Dalian University of Technology, Dalian, 116100, China
yhduan@mail.dlut.edu.cn, lin@dlut.edu.cn, wxs@mail.dlut.edu.cn

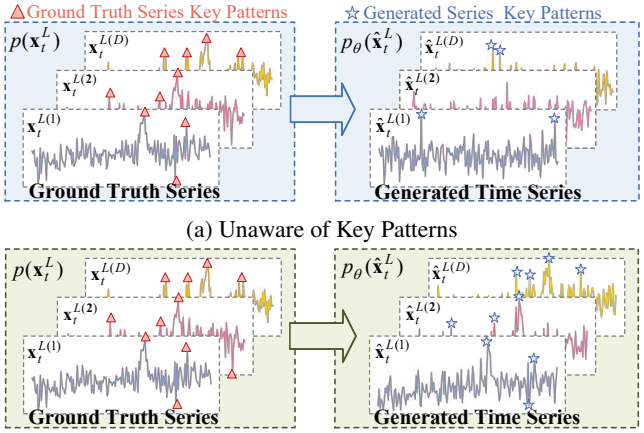
Abstract

Time series generation is essential for advancing data-driven modeling and decision-making across a wide range of domains. However, existing approaches primarily focus on global patterns, often failing to capture local key patterns such as abrupt changes or anomalies. These key patterns are crucial for interpretability and operational decision making, as they frequently represent intervention points with significant real-world impact. To bridge this gap, we propose **K**ey **Proto**types-Guided **Diff**usion (**K-ProtoDiff**) for time series generation, a new model that learns the global data distribution while preserving localized key patterns critical for temporal dynamics. In K-ProtoDiff, we first derive time series prototype representations through adaptive self-supervised learning. Then, a key prototype assignment module is used to extract prototype weights, forming key prototype-aware representations that serve as conditional guidance for generation. During sampling, to further enhance the fidelity of key patterns during the denoising process, we propose **R**eflection **Sampling** (**R-Sampling**), a step-wise refinement strategy that encourages the reverse trajectory to better align with key prototype constraints. Experiments on nine real-world datasets demonstrate that K-ProtoDiff significantly outperforms state-of-the-art baselines in key pattern retention, achieving an average **77.6%** improvement in key pattern preservation.

Code — https://github.com/YaofireJinglian/K-ProtoDiff

Introduction

In the era of data-driven intelligent systems, high-quality time series data is essential for improving model performance and decision making (Huang et al. 2025b). However, real-world constraints like acquisition costs and privacy concerns often lead to data scarcity, hindering intelligent systems (Yuan and Qiao 2024). Time Series Generation (TSG) mitigates this challenge by synthesizing diverse, high-quality data, supporting tasks such as data augmentation and privacy preservation (Coletta et al. 2023). A variety of TSG methods have been proposed to generate realistic time series, including GAN-based models (Yoon, Jarrett, and Van der Schaar 2019; Jeon et al. 2022), VAE-based models (Desai et al. 2021; Lee, Malacarne, and Aune 2023), and more recently, diffusion-based frameworks (Yuan and Qiao 2024; Li et al. 2025; Ge et al. 2025).



(a) Unaware of Key Patterns

(b) Key Patterns Aware

Figure 1: Comparison of time series generation methods. (a) Typical approach focusing on global patterns but overlooking key ones. (b) K-ProtoDiff, which preserves critical temporal structures by modeling key patterns.

Unfortunately, despite recent advances in TSG methods, most still focus on global data distributions while neglecting localized key patterns crucial for real-world applications, as shown in Figure 1a, which can undermine model reliability and lead to suboptimal decisions in high-stakes domains. This is especially challenging for diffusion-based models, where the iterative denoising process inherently favors global smoothness, unintentionally suppressing local key patterns. To mitigate this issue, Diffusion-TS (Yuan and Qiao 2024) employs a Transformer architecture to explicitly decouple the seasonal, trend, and residual components of time series, further enhanced by a Fourier synthesis layer to capture periodic structures. Nevertheless, its reliance on explicit decomposition may hinder the modeling of non-predefined yet critical dynamic patterns, thereby limiting adaptability in complex scenarios. PaD-TS (Li et al. 2025) introduces a Maximum Mean Discrepancy loss to explicitly align the group-level distributional properties of generated data, but this approach depends on predefined group metrics,

*Corresponding author is Lin Lin.

which may overlook essential patterns that are not explicitly quantified in real-world time series. Thus, effectively balancing global distribution modeling with the preservation of localized key patterns remains a crucial yet underexplored challenge in enhancing the quality of generated time series.

Fortunately, we discover that while diffusion models demonstrate strong generative capabilities, their reverse denoising process tends to average out sharp localized dynamics when optimizing toward global similarity. This observation motivates us to explore whether explicit structure-aware guidance can help diffusion models better preserve local key patterns during generation. Inspired by the effectiveness of prototype-based learning in summarizing representative patterns (Ni et al. 2023; Hautamaki, Nykanen, and Franti 2008), we extract key prototypes from time series data as soft guidance signals to enable fine-grained pattern control during denoising, thus better preserving important structural characteristics. However, we observe that static prototype guidance alone does not sufficiently constrain the generation process. In particular, the iterative nature of reverse diffusion can lead to semantic drift (Clifford et al. 2024), where the denoising trajectory gradually strays from key patterns. This problem is especially pronounced in long-range time series generation, where even small deviations may accumulate and compromise local fidelity. These observations suggest the need for a dynamic mechanism that can iteratively adjust the sampling trajectory to maintain alignment with the structural cues provided by the prototypes.

Based on the above motivations, we propose the **K**ey **Proto**types-Guided **Diff**usion for time series generation (**K-ProtoDiff**), a new method that models global data distributions while explicitly preserving localized key patterns crucial for temporal dynamics, as shown in Figure 1b. Specifically, K-ProtoDiff first employs a Key Prototypes Learner (KPL) to derive prototype representations via adaptive self-supervised learning, and then utilizes a prototype assignment module to construct prototype-aware representations that guide the generation process. In addition, drawing inspiration from the self-refinement mechanisms in large language models (LLMs) (Bai et al. 2024; Shinn et al. 2023), we propose Reflection Sampling (R-sampling), a stepwise refinement strategy that progressively aligns the reverse sampling trajectory with key prototype signals for better structural fidelity. The contributions are summarized as follows:

- To tackle the challenge of the key pattern loss in time series generation, we propose K-ProtoDiff, which leverages adaptively learned key prototypes to guide the diffusion process, enabling fine-grained control and improved retention of localized key patterns.
- We propose the KPL module, which extracts representative temporal prototypes to guide generation, and the R-sampling, a stepwise refinement strategy that preserves key patterns during the denoising process.
- We conduct extensive evaluations of K-ProtoDiff on nine real-world datasets across five diverse domains, demonstrating superior performance in generation realism and localized pattern retention.

Related Work

Models for Time Series Generation

In time series generation, early research mainly adopted Generative Adversarial Networks. TimeGAN (Yoon, Jarrett, and Van der Schaar 2019) combined adversarial training with autoregressive structures by introducing an embedding network and supervised loss to enhance temporal coherence. GT-GAN (Jeon et al. 2022) addressed irregular sampling by integrating Neural ODEs and Continuous-Time Flow Processes. Beyond adversarial approaches, Variational Autoencoder -based models have also been explored. TimeVAE (Desai et al. 2021) employs a convolutional encoder and explicitly models polynomial trends and seasonal components in the decoder, jointly optimizing reconstruction and regularization objectives. More recently, diffusion-based models have achieved strong performance. Diffusion-TS (Yuan and Qiao 2024) employs a Transformer with seasonal-trend decomposition and Fourier-based loss for high-quality generation. Compared to methods that focus solely on global distribution, our approach leverages key prototypes to guide the reverse sampling process, thereby enhancing the preservation of key patterns.

Models for Time Series Prototype

Time series prototypes (Ma et al. 2020; Shen 2025) have been widely used as fundamental representations of key temporal patterns to enhance interpretability and controllability in time series tasks. BasisFormer (Ni et al. 2023) learns interpretable temporal bases through contrastive learning between past and future views for personalized forecasting, demonstrating the effectiveness of shared prototypes in modeling complex temporal dynamics. ProtoAD (Li, Jentsch, and Müller 2023) leverages prototypes in anomaly detection by learning the distribution of normal patterns, thereby facilitating the identification of deviations indicative of anomalies. In the generative context, TimeDP (Huang et al. 2025b) treats prototypes as 'temporal tokens' to construct domain prompts, guiding diffusion-based generation across multiple domains while preserving domain-specific structures. In contrast to these approaches, we introduce key prototypes to guide the diffusion process, thereby improving the retention of localized key patterns during time series generation.

Preliminaries

Diffusion Models

Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) have shown impressive advances in time series generation, typically relying on forward and reverse Markov processes to model the data distribution. In the forward process, a clean time series $\mathbf{x}_0$ is progressively corrupted into a noisy sequence $\{\mathbf{x}_t\}_{t=1}^{T}$ via:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\mathbf{x}_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)\mathbf{I}\right), \tag{1}$$

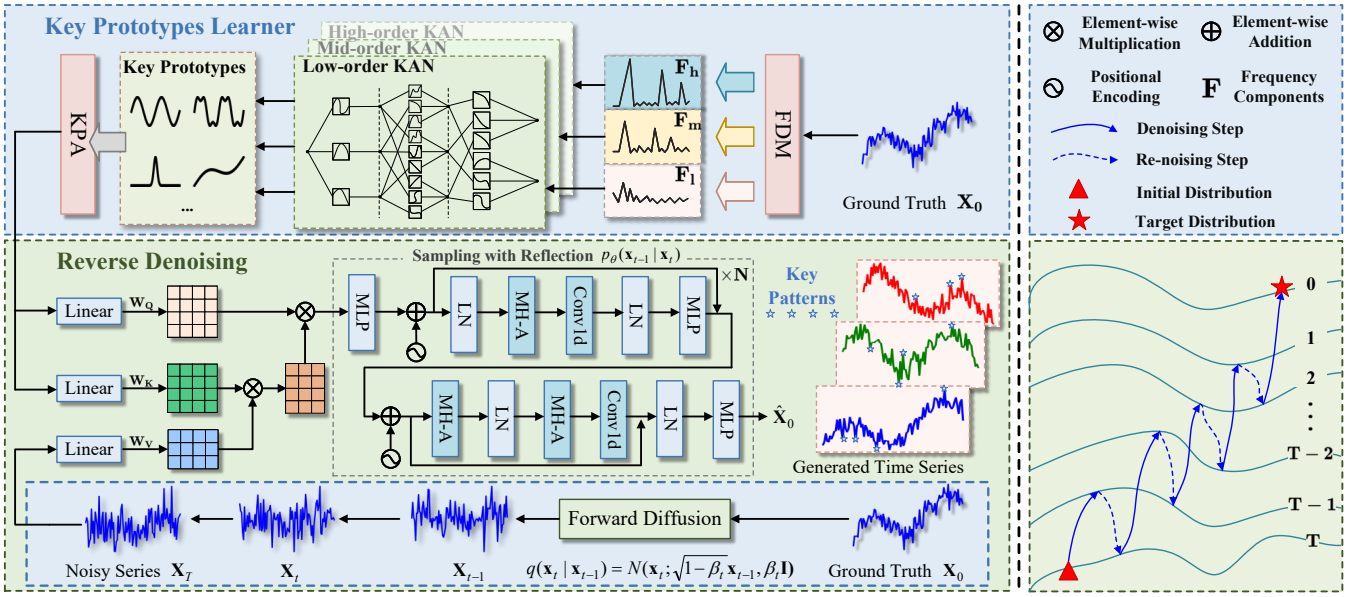


Figure 2: Overview of K-ProtoDiff model. The framework generates time series conditioned on key prototypes extracted by the KPL module (top left). The forward diffusion transforms the original series into noise, while the reverse process performs R-sample (bottom right) to align the denoising trajectory with key prototypes, producing high-quality time series.

where α_t is a predefined noise schedule. Due to the Markov property, $\mathbf{x}_t$ can also be directly sampled from $\mathbf{x}_0$ as:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \quad (2)$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

In the reverse process, a neural network ϵ_θ is trained to estimate the noise added at each timestep. The training objective minimizes the difference between the true noise ϵ and the predicted noise:

$$L_\theta(\mathbf{x}_0, t) = \left\| \epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right) \right\|^2. \quad (3)$$

After training, the model synthesizes time series by progressively denoising Gaussian noise via the learned reverse trajectory.

Problem Statement

The goal of time series generation is to learn a model that captures the underlying temporal dynamics of observed sequences and generates realistic synthetic time series. Formally, given a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, where each $\mathbf{x}_i = (x_1, x_2, \ldots, x_L)$ denotes a univariate or multivariate time series of length L, the objective is to learn a generative model $p_\theta(\mathbf{x} \mid \mathcal{P})$, where $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ is a set of representative prototypes capturing key temporal patterns shared across the dataset. The model is expected to generate samples $\hat{\mathbf{x}} \sim p_\theta(\cdot \mid \mathcal{P})$ that preserve both global distributions and localized key patterns guided by the selected prototypes.

Method

In this section, we first introduce **the K-ProtoDiff architecture**, as shown in Figure 2. We then detail two key components that enable the key pattern aware time series generation:

- **Key Prototypes Learner (KPL)**: A dedicated module responsible for extracting representative temporal patterns known as key prototypes, which capture critical structural information within the time series, as shown in Figure 2 (top left).
- **Reflection Sampling (R-sampling)**: A new refinement strategy that aligns the reverse diffusion trajectory with the learned prototypes through a stepwise correction process. This ensures that the generated time series retain important local and global dynamics, as shown in Figure 2 (bottom right).

Together, these modules work in concert to guide the generative process, ensuring that the generated time series remains to the key patterns.

The K-ProtoDiff Architecture

Figure 2 presents the overall framework of K-ProtoDiff. The model consists of two main stages: a prototype learning stage and a key prototype-guided diffu- sion stage. In the prototype learning stage, a set of key temporal patterns $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ are extracted from the training set $\mathcal{D}$, serving as structural priors. In the key prototype-guided diffusion stage, the generation process starts from Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the reverse denoising model $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathcal{P})$, implemented via a diffusion transformer module (Peebles and Xie 2023), progressively reconstructs the target series $\hat{\mathbf{x}}_0$. The learned prototypes $\mathcal{P}$ are incorporated into each denoising step via a reflection sampling mechanism, guiding the reverse trajectory toward the preservation of localized key patterns and ensuring structural consistency in the generated time series. Specifically, Algorithm 1 outlines the procedure of K-ProtoDiff.

Algorithm 1: K-ProtoDiff - Overall Architecture

Require: Training dataset $\mathcal{D}$, number of prototypes K, diffusion steps T, batch size B, total iterations I
Initialize: Prototype encoder $\phi(\cdot)$, diffusion model $g_\theta(\cdot)$, prototype memory $\mathcal{P} = \emptyset$

Extract $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\} \leftarrow \phi(\mathcal{D})$
for iter = 1 to I **do**
  Sample batch $X = \{\mathbf{x}_i\}_{i=1}^{B} \sim \mathcal{D}$
  Prototype assignments: $\{\mathbf{m}_i\}_{i=1}^{B} \leftarrow \phi(X)$
  for each sample $\mathbf{x}_i$ in X **do**
    Sample timestep $t \sim \text{Uniform}(1, T)$
    Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
    Corrupt input: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_i + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon$
    Predict noise: $\hat{\epsilon} = g_\theta(\mathbf{x}_t, t, \mathbf{m}_i)$
    Accumulate loss: $\mathcal{L} \leftarrow \mathcal{L} + \|\epsilon - \hat{\epsilon}\|^2$
  end for
  Update θ using gradient of total loss $\mathcal{L}$
end for
Initialize $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, select prototype $\mathbf{m} \in \mathcal{P}$
for $t = T$ to 1 **do**
  Predict noise: $\hat{\epsilon}_t = g_\theta(\mathbf{x}_t, t, \mathbf{m})$
  $\hat{\epsilon}_t \leftarrow \texttt{Reflect}(\hat{\epsilon}_t, \mathcal{P})$
  Reverse step: $\mathbf{x}_{t-1} = \texttt{Step}(\mathbf{x}_t, \hat{\epsilon}_t, t)$
end for
return Generated time series $\hat{\mathbf{x}}_0$

Key Prototypes Learner

The KPL module is built upon the Kolmogorov–Arnold Network (KAN) (Liu et al. 2024; Huang et al. 2025a) backbone, which enables compact representations of fundamental temporal patterns through functional decomposition. KAN replaces linear weights in MLPs with learnable univariate functions, enabling more expressive and interpretable modeling with fewer parameters. As shown in Figure 2 (top left), the module consists of three main components: a Frequency Decomposition Module (FDM), a multi-order KAN, and a Key Prototype Assignment (KPA) module.

Given an input time series $\mathbf{X}_0 \in R^{T \times d}$, the FDM first transforms it into the frequency domain via the Fast Fourier Transform (FFT) (Duhamel and Vetterli 1990):

$$\tilde{\mathbf{X}} = \mathcal{F}(\mathbf{X}_0) = \texttt{FFT}(\mathbf{X}_0), \tag{4}$$

where $\tilde{\mathbf{X}} \in C^{T \times d}$ denotes the frequency spectrum.

The spectrum is partitioned into three disjoint frequency bands, and each is projected back into the time domain via inverse FFT:

$$\mathbf{F}_k = \mathcal{F}^{-1}(\tilde{\mathbf{X}}_{[f_{k-1}:f_k]}), \quad k = l, m, h \tag{5}$$

where $\mathcal{F}^{-1}(\cdot)$ denotes the inverse FFT, and frequency intervals $[0 : f_l], [f_l : f_h], [f_h : f_{\max}]$ define low-, mid-, and high-frequency bands.

Here, we adopt TaylorKAN (Liu et al. 2025) to learn the representation of each frequency component. Each component $\mathbf{F}_o$, where $o \in \{l, m, h\}$, is then processed by a KAN, which approximates a multivariate function via superpositions of univariate functions:

$$\mathbf{H}_o = \text{KAN}^{(o)}(\mathbf{F}_o) = \sum_{q=1}^{Q} \phi_q^{(o)} \left(\sum_{p=1}^{d} \psi_{q,p}^{(o)} (\mathbf{F}_{o,p}) \right), \tag{6}$$

where $\phi_q^{(o)}$ and $\psi_{q,p}^{(o)}$ are learnable univariate functions, and Q denotes the number of superposition terms.

The outputs $\mathbf{H}_l, \mathbf{H}_m, \mathbf{H}_h$ are aggregated to form a unified representation $\mathbf{H} \in R^{T \times d'}$, which is then fed into the KPA module. KPA computes soft attention weights over a learnable prototype set $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\} \in R^{T \times d'}$:

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{H}\mathbf{W}_Q (\mathcal{P}\mathbf{W}_K)^\top}{\sqrt{d'}} \right), \quad \mathbf{Z}_p = \mathbf{A} \cdot \mathcal{P}, \tag{7}$$

where $\mathbf{W}_Q, \mathbf{W}_K \in R^{d' \times d'}$ are learnable projection matrices, and $\mathbf{Z}_p$ denotes key prototypes.

Reflection Sampling

To enhance structural alignment with the learned key prototypes $\mathcal{P}$ during reverse diffusion, we propose Reflection Sampling (R-sampling), a refinement step that projects the predicted noise at each denoising step toward the prototype-guided subspace.

At step t, let $\hat{\boldsymbol{\epsilon}}_t = \epsilon_\theta(\mathbf{x}_t, t)$ denote the predicted noise from the reverse model. Instead of directly using $\hat{\boldsymbol{\epsilon}}_t$, we compute its orthogonal projection onto the subspace spanned by the prototypes:

$$\Pi_{\mathcal{P}}(\hat{\boldsymbol{\epsilon}}_t) = \arg \min_{\mathbf{z} \in \text{Span}(\mathcal{P})} \|\hat{\boldsymbol{\epsilon}}_t - \mathbf{z}\|^2. \tag{8}$$

To balance between the original prediction and prototype conformity, we form a convex interpolation:

$$\hat{\boldsymbol{\epsilon}}_t^{\text{r}} = (1 - \lambda_t)\hat{\boldsymbol{\epsilon}}_t + \lambda_t \Pi_{\mathcal{P}}(\hat{\boldsymbol{\epsilon}}_t), \tag{9}$$

where $\lambda_t \in [0, 1]$ controls the strength of the reflection toward the prototype subspace. This reflection operation can be interpreted as a proximal step toward the subspace $\text{Span}(\mathcal{P})$:

$$\hat{\boldsymbol{\epsilon}}_t^{\text{r}} = \text{prox}_{\lambda_t \cdot \mathcal{I}_{\text{Span}(\mathcal{P})}}(\hat{\boldsymbol{\epsilon}}_t), \tag{10}$$

with $\mathcal{I}$ being the indicator function of the subspace. Finally, the refined noise $\hat{\boldsymbol{\epsilon}}_t^{\text{r}}$ is used to update the state in the reverse diffusion process:

$$\mathbf{x}_{t-1} = \Phi_t(\mathbf{x}_t, \hat{\boldsymbol{\epsilon}}_t^{\text{r}}), \tag{11}$$

where $\Phi_t(\cdot)$ denotes the deterministic reverse step defined by the diffusion framework.

Experiments

We conducted a comprehensive evaluation on nine datasets across five domains to assess the performance of K-ProtoDiff, aiming to answer the following key research questions:

- **Q1**: How does K-ProtoDiff perform compared to existing state-of-the-art time series generation methods in terms of distributional realism and pattern fidelity?
- **Q2**: How effectively does K-ProtoDiff preserve localized key patterns relative to existing state-of-the-art models?

Metric	Methods	ETTh	Electricity	Energy	Traffic	Weather	Illness	Exchange	Stocks	EEG
C-FID ↓	Diffusion-TS	0.139±0.003	0.073±0.005	0.110±0.018	0.818±0.019	0.362±0.035	0.071±0.044	0.049±0.004	0.150±0.028	0.126±0.017
	PaD-TS	2.020±1.399	5.195±1.656	2.002±1.059	4.055±4.269	4.456±1.870	4.169±1.434	2.154±0.776	2.636±1.692	2.662±1.078
	TimeGAN	0.223±0.015	0.716±0.062	1.063±0.185	1.745±0.188	0.189±0.009	1.357±0.085	0.438±0.061	0.109±0.017	5.798±0.165
	GT-GAN	7.514±0.423	2.847±0.284	5.174±0.994	OOM	2.972±0.394	2.489±0.471	0.813±0.239	4.263±0.962	4.854±0.983
	TimeVAE	0.809±0.107	0.081±0.006	1.636±0.082	0.203±0.031	0.306±0.021	0.172±0.009	0.071±0.024	0.208±0.048	**0.018±0.001**
	TimeVQVAE	3.299±0.195	14.504±1.217	5.956±0.285	19.978±1.390	14.347±8.943	7.856±0.771	9.965±1.094	9.872±1.040	8.118±2.836
	K-ProtoDiff	**0.014±0.001**	**0.011±0.000**	**0.019±0.001**	**0.037±0.005**	**0.008±0.000**	**0.022±0.002**	**0.006±0.001**	**0.067±0.036**	0.116±0.009
KL ↓	Diffusion-TS	0.008	0.007	**0.011**	0.017	**0.019**	**0.011**	**0.014**	0.222	0.367
	PaD-TS	0.417	0.405	0.178	0.312	0.418	0.110	0.150	1.091	8.875
	TimeGAN	0.034	0.012	0.028	0.016	**0.019**	0.266	0.125	0.650	0.127
	GT-GAN	0.056	0.010	0.044	OOM	0.051	0.070	0.096	0.331	15.194
	TimeVAE	0.067	0.014	0.077	0.024	0.069	0.045	0.040	0.184	0.157
	TimeVQVAE	0.206	0.605	0.442	1.652	0.044	0.901	3.674	1.304	**0.116**
	K-ProtoDiff	**0.006**	**0.001**	0.013	**0.002**	0.029	0.029	0.017	**0.183**	0.607
DS ↓	Diffusion-TS	0.086	0.387	0.126	0.497	0.167	0.173	0.078	0.172	**0.053**
	PaD-TS	0.500	0.500	0.500	0.500	0.500	0.499	0.500	0.500	0.500
	TimeGAN	0.053	0.500	0.495	0.500	0.307	0.336	0.202	0.189	0.281
	GT-GAN	0.444	0.500	0.498	OOM	0.493	0.376	0.298	0.374	0.500
	TimeVAE	0.147	0.456	0.499	0.492	0.446	0.182	0.029	0.140	0.103
	TimeVQVAE	0.456	0.500	0.500	0.500	0.499	0.497	0.499	0.446	0.499
	K-ProtoDiff	**0.002**	**0.119**	**0.050**	**0.475**	**0.069**	**0.015**	**0.016**	**0.103**	0.491
PS ↓	Diffusion-TS	0.127	0.021	0.251	0.010	0.002	0.032	0.045	0.034	0.449
	PaD-TS	0.245	0.255	0.489	0.848	0.340	0.112	0.180	0.072	0.452
	TimeGAN	0.121	0.035	0.316	0.021	0.002	0.048	0.051	0.034	0.551
	GT-GAN	0.256	0.037	0.346	OOM	0.003	0.037	0.049	0.144	0.502
	TimeVAE	0.125	0.021	0.285	0.013	0.002	0.034	0.038	0.041	0.451
	TimeVQVAE	0.796	0.645	0.999	0.627	0.082	0.578	0.618	0.654	0.978
	K-ProtoDiff	**0.120**	**0.019**	**0.128**	**0.008**	**0.001**	**0.027**	**0.037**	**0.025**	**0.426**

Table 1: Performance comparison results on multiple time series datasets. OOM indicates out-of-memory error.

- **Q3**: How does each core component of K-ProtoDiff contribute to the overall quality of generated time series?
- **Q4**: How sensitive is K-ProtoDiff to key hyperparameters?
- **Q5**: How does K-ProtoDiff perform in long-term time series generation?

Experimental Settings

All experiments were conducted on a single NVIDIA RTX 4090 (24GB) with Ubuntu 20.04. All the reproduced baseline models were entirely built upon the configurations provided in the original papers or official code repositories for each model.

Datasets. We evaluated our model on nine real-world datasets, including ETT, Electricity, Traffic, Weather, Illness, Exchange (Wu et al. 2021), Energy (Candanedo 2017), Google Stocks (Aroussi 2023), and EEG (Yuan and Qiao 2024), which span five critical domains: energy, transportation, climate, healthcare, and finance.

Evaluation Metrics. To systematically address the aforementioned research questions, we evaluate performance using five widely adopted metrics:

- **Context-FID Score (C-FID)** (Jeha et al. 2022) quantifies the distributional difference between real and synthetic sequences based on context-aware embeddings.
- **Kullback-Leibler Divergence (KL)** (Nielsen 2022) measures the global distributional discrepancy by comparing the marginal probability distributions of real and synthetic data.
- **Discriminative Score (DS)** (Yoon, Jarrett, and Van der

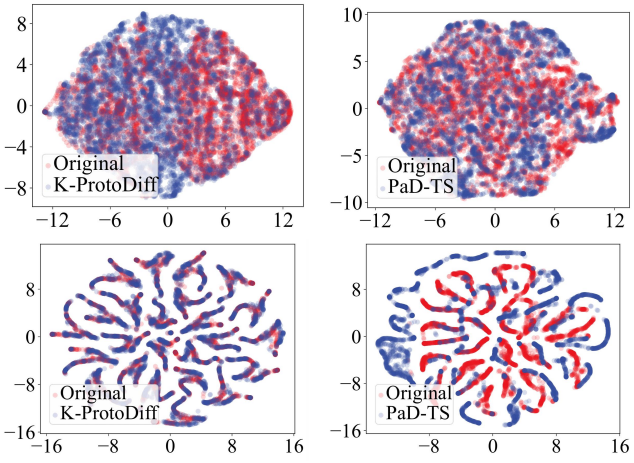


Figure 3: t-SNE visualizations of the time series generated by K-ProtoDiff and PaD-TS on the ETTh (top) and Electricity (bottom) datasets.

Metric	Methods	ETTh	Electricity	Energy	Traffic	Weather	Illness	Exchange	Stocks	EEG
Segment-wise DTW Segment (Length = 6) ↓	Diffusion-TS	1.221	9.193	3.272	10.552	**2.259**	1.274	2.005	1.224	1.214
	PaD-TS	3.641	29.109	9.257	63.890	8.444	5.389	4.843	5.432	8.291
	TimeGAN	1.251	9.263	3.261	10.623	2.374	1.308	**1.926**	1.258	1.374
	GT-GAN	1.533	9.653	3.490	OOM	2.746	1.514	1.991	1.548	2.674
	TimeVAE	1.239	9.283	3.257	11.107	2.366	1.338	1.996	1.169	1.231
	TimeVQVAE	7.315	48.608	14.265	71.041	12.314	6.425	6.858	5.879	7.208
	K-ProtoDiff	**1.124**	**9.176**	**2.994**	**9.848**	2.289	**1.261**	1.956	**1.120**	**1.207**
Segment-wise DTW Segment (Length = 8) ↓	Diffusion-TS	1.411	10.496	3.767	12.023	**2.610**	1.482	2.316	1.415	1.404
	PaD-TS	4.206	33.770	10.690	73.928	9.761	6.223	5.599	6.262	9.574
	TimeGAN	1.445	10.538	3.765	11.991	2.742	1.516	**2.223**	1.451	1.588
	GT-GAN	1.773	11.205	4.041	OOM	3.187	1.739	2.300	1.802	3.081
	TimeVAE	1.431	10.585	3.749	12.703	2.734	1.559	2.305	1.351	1.425
	TimeVQVAE	8.449	56.011	16.471	82.223	14.230	7.432	7.929	6.807	8.406
	K-ProtoDiff	**1.299**	**10.134**	**3.459**	**11.253**	2.643	**1.468**	2.260	**1.295**	**1.395**
Segment-wise DTW Segment (Length = 10) ↓	Diffusion-TS	1.576	11.513	4.201	13.110	**2.922**	1.661	2.589	1.580	1.571
	PaD-TS	4.703	37.932	11.953	82.767	10.902	6.980	6.254	7.012	10.704
	TimeGAN	1.612	11.503	4.203	12.985	3.067	1.703	**2.487**	1.623	1.777
	GT-GAN	1.985	12.250	4.515	OOM	3.578	1.958	2.572	2.015	3.445
	TimeVAE	1.596	11.598	4.183	13.916	3.057	1.746	2.577	1.510	1.597
	TimeVQVAE	9.426	62.274	18.413	91.516	16.014	8.330	8.865	7.636	9.498
	K-ProtoDiff	**1.453**	**11.267**	**3.867**	**12.289**	2.958	**1.643**	2.526	**1.447**	**1.561**

Table 2: Comparison of key patterns preservation results on multiple time series datasets. OOM indicates out-of-memory error.

Schaar 2019) evaluates the distinguishability between real and synthetic data by training a classifier in a supervised setting.

- **Predictive Score (PS)** (Yoon, Jarrett, and Van der Schaar 2019) assesses the utility of synthetic data for downstream tasks.
- **Segment-wise Dynamic Time Warping (Segment-wise DTW)** (Belkhouja, Yan, and Doppa 2022) evaluates local pattern similarity and fine-grained temporal alignment by computing DTW distances over fixed-length segments.

Baselines. We carefully select six widely recognized state-of-the-art models as benchmarks for our evaluation, including: (1) VAE-based methods: TimeVAE (Desai et al. 2021) and TimeVQVAE (Lee, Malacarne, and Aune 2023); (2) GAN-based methods: TimeGAN (Yoon, Jarrett, and Van der Schaar 2019) and GT-GAN (Jeon et al. 2022); and (3) Diffusion-based methods: Diffusion-TS (Yuan and Qiao 2024) and PaD-TS (Li et al. 2025).

Performance Comparison (A1)

Table 1 presents the 24 time-step length generation results, a common setting in related work. K-ProtoDiff consistently achieves top performance across all four evaluation metrics. It records the best C-FID on 8 out of 9 datasets (88.9%), outperforming Diffusion-TS and other baselines. For KL, it ranks first on 4 datasets, indicating better distributional alignment. On DS, K-ProtoDiff achieves the lowest values on 8 datasets, reflecting strong distributional fidelity. It also secures the best PS across all datasets, demonstrating its capability to preserve temporal and feature dependencies. These results highlight K-ProtoDiff's effectiveness in generating realistic, structure-aware time series across diverse domains. Its advantage is especially evident on complex datasets like Traffic and Electricity, where it shows large gains in C-FID and DS. To further illustrate generation quality, we synthesize 5,000 sequences on ETTh and Electricity, and use t-SNE (Van der Maaten and Hinton 2008) to project real and generated samples into 2D. As shown in Figure 3, K-ProtoDiff shows significantly better alignment with the real distribution than PaD-TS, demonstrating its strength across both low- and high-dimensional settings.

Preservation of Key Patterns Performance (A2)

To quantitatively evaluate the preservation of localized key patterns, we adopt segment-wise DTW with a fixed stride of 4 and varying segment lengths $L \in \{6, 8, 10\}$ across all datasets. This metric measures alignment errors between local segments, providing a fine-grained assessment of temporal fidelity. As shown in Table 2, K-ProtoDiff achieves the lowest segment-wise DTW scores on the majority of datasets and segment lengths, demonstrating superior ability to retain fine-grained temporal structures compared to state-of-the-art baselines. Notably, on complex datasets such as Traffic, Electricity, and EEG, K-ProtoDiff achieves substantial reductions in DTW error, outperforming TimeVQVAE by over 85% on average, highlighting its robustness in modeling localized temporal dynamics. These improvements are particularly significant given the inherent variability and noise in real-world signals, where local temporal alignment plays a crucial role. Overall, these results confirm that K-ProtoDiff effectively preserves localized key patterns, which are critical for high-fidelity time series synthesis and downstream temporal reasoning tasks.

Ablation Study (A3)

To validate the effectiveness of individual components in K-ProtoDiff, we conductd comprehensive ablation studies, including both component replacement (Replace) and removal (w/o) experiments. The results are listed in Table 3. In particular, we replace the proposed KPL module, which leverages a KAN-based architecture, with a MLP-based prototype module (MP) (Ni et al. 2023), and R-sampling (R-s) is replaced with P-sampling (P-s). Notably, the combination of R-sampling (R-s) and KPL (i.e., full K-ProtoDiff) yields the best results, with the lowest PS and Seg-DTW scores on both datasets. In contrast, replacing KPL with MP or removing it leads to clear performance drops, particularly under the R-sampling setting. These results demonstrate a strong synergy between R-sampling and KPL, suggesting that KPL's structural guidance is most effective when paired with the diverse sampling of R-sampling.

Design	Variants		Stocks		EEG	
	R-s	KPL	PS↓	Seg-DWT↓	PS↓	Seg-DWT↓
Replace	R-s	MP	0.138	1.302	0.483	1.394
	P-s	KPL	0.034	1.274	0.451	1.316
	P-s	MP	0.048	1.445	0.499	1.511
w/o	R-s	w/o	0.141	1.578	0.584	1.644
	P-s	w/o	0.101	1.378	0.491	1.442
Ours	**R-s**	**KPL**	**0.025**	**1.120**	**0.426**	**1.207**

Table 3: Ablation study on the effectiveness of K-ProtoDiff components on Stocks and EEG datasets.

Hyperparameter Sensitivity Analysis (A4)

We evaluated the sensitivity of the KPL module on the Energy dataset with respect to the number of prototypes and the order of the KAN, using Predictive Score and Segment-wise DTW as evaluation metrics. As shown in Figure 4, the lowest PS (0.128) is observed when using four prototypes with a KAN order of 2, indicating optimal preservation of global patterns. In contrast, the lowest Seg-DTW (2.990) is achieved with the same number of prototypes but a KAN order of 1, suggesting that simpler configurations may better

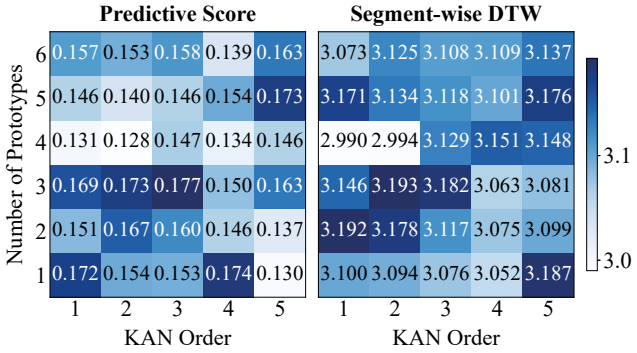


Figure 4: Sensitivity analysis on the Energy dataset.

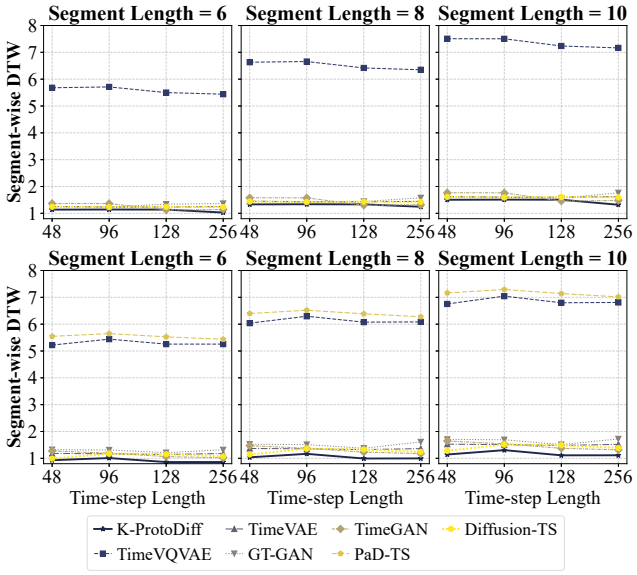


Figure 5: Seg-DTW across different segment lengths.

capture local temporal structures. These findings underscore the importance of balancing global and local similarity when tuning model complexity and highlight the robustness of our approach to variations in hyperparameters.

Long-term Time series Generation (A5)

We evaluated model performance on the EEG (top) and Stocks (bottom) datasets using Segment-wise DTW across varying segment lengths (6, 8, and 10) and generation lengths (48–256). As shown in Figure 5, most models exhibit increasing DTW scores with longer sequence lengths, indicating a decline in local pattern preservation. In contrast, K-ProtoDiff consistently achieves the lowest DTW scores across all settings, demonstrating its strong ability to retain fine-grained local structures. Models such as TimeVQVAE and PaD-TS, while competitive in other metrics, often yield significantly higher DTW values, suggesting weaker structural alignment. These findings underscore the robustness of K-ProtoDiff in generating time series that maintain structural coherence over extended temporal horizons.

Conclusions

In this paper, we propose K-ProtoDiff, a prototype-guided diffusion framework for time series generation. It leverages adaptively learned prototypes as structural priors to guide generation. A prototype assignment module produces prototype-aware representations, enabling conditioning on localized patterns. To mitigate semantic drift during denoising, we introduce Reflection Sampling, a stepwise refinement strategy aligning the reverse trajectory with key prototypes. Experiments on nine real-world datasets across five domains demonstrate that K-ProtoDiff achieves state-of-the-art performance in both generation realism and fine-grained pattern retention.

Acknowledgments

This research is supported by the National Key R&D Program of China under Grant 2021YFA1003003, SONY Sensing Solution University Joint Development Project 2024-2025, the National Natural Science Foundation of China under Grant 62076053.

References

Aroussi, R. 2023. yfinance: Yahoo! Finance market data downloader. https://github.com/ranaroussi/yfinance. Accessed: 2025-04-26.

Bai, L.; Shao, S.; Zhou, Z.; Qi, Z.; Xu, Z.; Xiong, H.; and Xie, Z. 2024. Zigzag Diffusion Sampling: Diffusion Models Can Self-Improve via Self-Reflection. In *The Thirteenth International Conference on Learning Representations*.

Belkhouja, T.; Yan, Y.; and Doppa, J. R. 2022. Dynamic time warping based adversarial framework for time-series domain. *IEEE transactions on pattern analysis and machine intelligence*, 45(6): 7353–7366.

Candanedo, L. 2017. Appliances Energy Prediction. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5VC8G.

Clifford, P.; Brown, D.; Anderson, P.; Angelopoulos, R.; and Wainwright, M. 2024. Enhancing large language models with stochastic semantic drift: A novel conceptual enhancement. *Authorea Preprints*.

Coletta, A.; Gopalakrishnan, S.; Borrajo, D.; and Vyetrenko, S. 2023. On the constrained time-series generation problem. *Advances in Neural Information Processing Systems*, 36: 61048–61059.

Desai, A.; Freeman, C.; Wang, Z.; and Beaver, I. 2021. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*.

Duhamel, P.; and Vetterli, M. 1990. Fast Fourier transforms: a tutorial review and a state of the art. *Signal processing*, 19(4): 259–299.

Ge, Y.; Li, J.; Zhao, Y.; Wen, H.; Li, Z.; Qiu, M.; Li, H.; Jin, M.; and Pan, S. 2025. T2S: High-resolution Time Series Generation with Text-to-Series Diffusion Models. *arXiv preprint arXiv:2505.02417*.

Hautamaki, V.; Nykanen, P.; and Franti, P. 2008. Time-series clustering by approximate prototypes. In *2008 19th International conference on pattern recognition*, 1–4. IEEE.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Huang, S.; Zhao, Z.; Li, C.; and BAI, L. 2025a. TimeKAN: KAN-based Frequency Decomposition Learning Architecture for Long-term Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.

Huang, Y.-H.; Xu, C.; Wu, Y.; Li, W.-J.; and Bian, J. 2025b. TimeDP: Learning to Generate Multi-Domain Time Series with Domain Prompts. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*.

Jeha, P.; Bohlke-Schneider, M.; Mercado, P.; Kapoor, S.; Nirwan, R. S.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2022. PSA-GAN: Progressive self attention GANs for synthetic time series. In *The Tenth International Conference on Learning Representations*.

Jeon, J.; Kim, J.; Song, H.; Cho, S.; and Park, N. 2022. GT-GAN: General purpose time series synthesis with generative adversarial networks. *Advances in Neural Information Processing Systems*, 35: 36999–37010.

Lee, D.; Malacarne, S.; and Aune, E. 2023. Vector quantized time series generation with a bidirectional prior model. *arXiv preprint arXiv:2303.04743*.

Li, B.; Jentsch, C.; and Müller, E. 2023. Prototypes as explanation for time series anomaly detection. *arXiv preprint arXiv:2307.01601*.

Li, Y.; Meng, H.; Bi, Z.; Urnes, I. T.; and Chen, H. 2025. Population Aware Diffusion for Time Series Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17): 18520–18529.

Liu, Y.; Zhang, X.; Zhang, Z.; and Feng, H. 2025. Adaptive Taylor Kolmogorov-Arnold Network for Hyperspectral Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Ma, D.; Wang, Z.; Xie, J.; Yu, Z.; Guo, B.; and Zhou, X. 2020. Modeling Multivariate Time Series via Prototype Learning: a Multi-Level Attention-based Perspective. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 687–693.

Ni, Z.; Yu, H.; Liu, S.; Li, J.; and Lin, W. 2023. Basisformer: Attention-based Time Series Forecasting with Learnable and Interpretable Basis. In *Advances in Neural Information Processing Systems*.

Nielsen, F. 2022. The Kullback–Leibler divergence between lattice Gaussian distributions. *Journal of the Indian Institute of Science*, 102(4): 1177–1188.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Shen, K.-Y. 2025. Learn hybrid prototypes for multivariate time series anomaly detection. In *The Thirteenth International Conference on Learning Representations*.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: language agents with verbal reinforcement learning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 8634–8652. Curran Associates, Inc.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. pmlr.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*.

Yoon, J.; Jarrett, D.; and Van der Schaar, M. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.

Yuan, X.; and Qiao, Y. 2024. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In *The Twelfth International Conference on Learning Representations*.