GRAPH2TS: STRUCTURE-CONDITIONED TIME SERIES GENERATION FROM QUANTILE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series generation requires capturing recurring temporal structure while preserving stochastic variation across its realizations. We approach this problem from a *structure–residual* perspective, representing a time series through a structural condition while modeling the remaining variation stochastically. Based on this perspective, we introduce **Graph2TS**, a structure-conditioned time-series generator that uses quantile graphs as structural representations. A quantile graph compactly summarizes empirical one-step transitions between globally shared amplitude-quantile states, providing a fixed-size, lossy representation of distributional and local transition structure. Graph2TS conditions generation on this representation and uses a latent variable to model variation not specified by the graph. All graphs and normalization statistics are derived from training data, and generation requires no test-specific conditioning information. Experiments against GAN, diffusion, and flow-matching baselines on four real-world datasets show particularly strong performance on ECG and EEG signals, while remaining competitive on Sunspot and Electricity with dataset-dependent trade-offs. Dataset analysis further suggests that these benefits are most pronounced for volatile signals with weak or irregular periodicity. Additional Electricity experiments up to length 256 show stronger relative performance over DiffusionTS at longer sequence lengths, although coverage decreases as sequence length increases. These results support the structure–residual perspective and suggest graph-based structural representations as a promising direction for time-series generation.

1 INTRODUCTION

Time series generation is important in healthcare, finance, and scientific modeling (Lin et al., 2024; Brophy et al., 2021). Temporal dependencies, oscillatory patterns, and stochastic fluctuations make realistic synthesis challenging (Gonen et al., 2025; Yang et al., 2025). A useful generator must capture recurring distributional and transition structure while modeling variation across individual realizations.

GAN-, diffusion-, and flow-matching-based models learn these properties through different objectives (Yoon et al., 2019; Yuan & Qiao, 2024; Kollovieh et al., 2025). Matching marginal distributions alone does not ensure accurate temporal or spectral structure, while generating individually realistic trajectories does not ensure adequate coverage of the data distribution. This motivates a structure–residual perspective: summarize selected temporal relationships explicitly, then model the variability that remains conditional on that summary.

We instantiate this perspective with a quantile graph. Its nodes are globally shared amplitude-quantile states, and its weighted directed edges summarize one-step transitions within a sequence. The graph has a fixed size for a chosen number of states, independent of sequence length. It provides a compact structural summary rather than specifying a unique temporal ordering or complete frequency spectrum. The generative model must therefore learn how this summary relates to the distribution of time-series realizations.

Based on this representation, we propose **Graph2TS**, a structure-conditioned generative framework for time-series synthesis. A graph encoder supplies structural features to the decoder, while a sampled latent variable represents conditional variability. Training uses paired sequences and graphs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

constructed from those sequences, with no external labels or annotations. For generation, conditioning graphs are selected only from the training set and latent variables are sampled from the prior, with held-out time series used only for evaluation.

Our main contributions are:

- We formulate a structure–residual perspective for time-series generation, motivated by a conditional-mean decomposition that separates variation explained by a structural condition from residual variation within that condition.
- We instantiate structural conditioning with fixed-size quantile graphs defined over training-set quantile boundaries, providing a compact summary of local transition structure in a globally shared amplitude state space.
- We introduce **Graph2TS**, a structure-conditioned time-series generator, and evaluate it on four real datasets using training-set graph conditions. Graph2TS shows strong fidelity with dataset-dependent trade-offs, and increasing relative advantage over DiffusionTS on Electricity up to $T = 256$.

2 Related Work

We position Graph2TS with respect to time-series generative models, conditional generation, and graph-based representations, with particular attention to the distinction between internally derived structural conditions and externally provided conditioning information.

2.1 Time-Series Generative Models

Time-series generation has been studied through several complementary frameworks. GAN-based approaches extend adversarial learning to sequential data (Goodfellow et al., 2020; Ahmed & Schmidt-Thieme, 2023), with TimeGAN combining adversarial and supervised objectives to promote temporal coherence (Yoon et al., 2019). Diffusion-based approaches learn iterative denoising processes for synthesis and probabilistic forecasting (Rasul et al., 2021; Yang et al., 2026; Lin et al., 2024), including DiffusionTS for general time-series generation (Yuan & Qiao, 2024). Flow matching provides another approach to continuous generative modeling; TSFlow combines conditional flow matching with structured priors for time-series synthesis and forecasting (Kollovieh et al., 2025).

These approaches differ in their generative objectives and inductive biases. Graph2TS instead explicitly represents selected transition structure with a quantile graph and uses this representation to guide time-series generation. The graph is constructed directly from the training time series, requiring no external annotations or auxiliary data.

2.2 Conditional Time-Series Generation

Conditional generation introduces additional information to guide time-series synthesis (Coletta et al., 2023; Li et al., 2023). Conditional GANs and related models often condition on labels, context variables, or handcrafted features provided alongside the time series (Smith & Smith, 2020; Lu et al., 2024; Ni et al., 2020; Narasimhan et al., 2024). Representation-learning approaches instead derive latent descriptions directly from time-series data (Zhang et al., 2024; Yue et al., 2022; Oublal et al., 2024). Graph2TS similarly requires no external annotations, but uses a quantile graph computed from each training sequence as an explicit structural condition. Its graph and latent variable serve complementary roles, representing structural conditions and stochastic variation, respectively. Graph2TS is conditional on a graph internally, but supports unconditional generation by sampling graph conditions directly from the training-set graph distribution.

2.3 Graph-Based Representations for Time Series

Graph representations encode selected temporal relationships as structured objects (Luque et al., 2009; Lacasa et al., 2008). Statistical, topology-inspired, and learned graph approaches capture different aspects of recurrence, dynamics, and distributions (Silva et al., 2023; Zou et al., 2019; Silva et al., 2021; Chen & Eldardiry, 2024). Among these representations, quantile graphs describe time

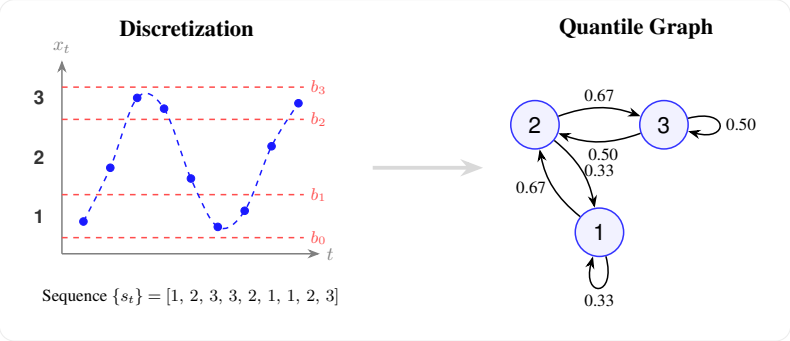


Figure 1: Time-series to quantile-graph mapping. Shared quantile boundaries discretize amplitudes into states; directed edge weights summarize observed one-step transitions.

series through transitions between amplitude-quantile states (Campanharo et al., 2011; Lopes de Oliveira Campanharo & Ramos, 2015). Graph2TS builds on this representation by using the resulting transition graph as a structural condition for time-series generation. Shared training-set quantile boundaries provide a common state space across samples, while each graph summarizes the transitions observed within an individual sequence.

3 Problem Statement

3.1 Setting and Motivation

We consider univariate time-series windows $x_i = (x_{i,1}, \ldots, x_{i,T}) \in \mathbb{R}^T$. The goal is to generate diverse new sequences while preserving characteristic distributional properties and variation among samples. A conceptual decomposition $X = X^{\text{struct}} + \varepsilon$ separates recurring structure from residual variability. This motivates an explicit conditioning representation, without assuming that the two components can be uniquely recovered from an observed sequence.

3.2 Structure-Conditioned Formulation

Let G denote a structural representation derived directly from a time series through a mapping Φ:

$$G = \Phi(X), \qquad g_i = \Phi(x_i). \tag{1}$$

We learn $p_\theta(X \mid G)$ from training pairs (x_i, g_i), where both the sequence and its structural condition are obtained from the training set. Thus, G is neither an external annotation nor information derived from held-out data. At generation time, structural conditions are sampled from those constructed from the training set, while held-out sequences are used only for evaluation (Section 4.2).

For this formulation to be meaningful, Φ should be a many-to-one structural mapping that retains selected structural properties without uniquely specifying an individual sequence. This leaves variation among sequences sharing a structural condition to be modeled stochastically.

3.3 Quantile Graph as Structural Representation

We estimate globally shared quantile boundaries $\mathcal{B} = \{b_0, \ldots, b_Q\}$ from the training data and fix them thereafter (Appendix B.4). Each observation is assigned to a state,

$$s_{i,t} = q(x_{i,t}), \qquad q(x) = k \text{ if } x \in [b_{k-1}, b_k), \tag{2}$$

so that a given state has aligned amplitude-quantile semantics across samples. Figure 1 illustrates this mapping.

For each sequence, let $C_{ab}^{(i)} = \sum_{t=1}^{T-1} \mathbf{1}[s_{i,t} = a, s_{i,t+1} = b]$ and $N_a^{(i)} = \sum_b C_{ab}^{(i)}$. For states with observed outgoing transitions, row normalization gives

$$P_{ab}^{(i)} = \begin{cases} C_{ab}^{(i)} / N_a^{(i)}, & N_a^{(i)} > 0, \\ 0, & N_a^{(i)} = 0. \end{cases} \tag{3}$$

The resulting $Q \times Q$ transition representation is the quantile graph $g_i = \Phi_{\mathcal{B}}(x_i)$. Its size depends on Q, not on T, making it suitable for a fixed-dimensional conditioning encoder. The quantile graph provides a many-to-one structural summary: it preserves empirical one-step transition frequencies while discarding the positions at which individual transitions occur. Consequently, distinct sequences can share the same graph, leaving realization-specific temporal variation to be modeled by the generator.

4 GRAPH2TS: STRUCTURE-CONDITIONED TIME SERIES GENERATION

Building on the structure-conditioned formulation introduced in the previous section, we instantiate the proposed formulation using Graph2TS. Graph2TS conditions time series generation on a quantile graph representation, which summarizes empirical one-step transitions, while a latent variable captures residual stochastic variability.

4.1 GRAPH2TS ARCHITECTURE

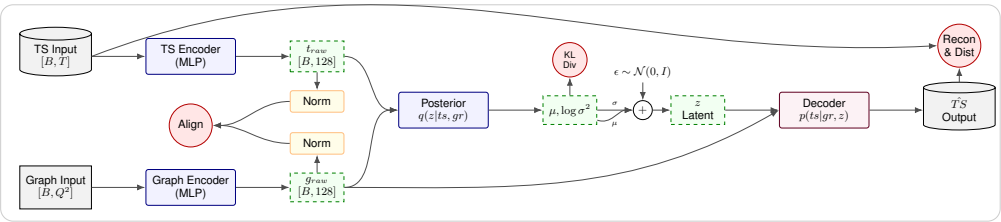


Figure 2: Graph2TS training architecture for sequence length T and Q quantile states. At generation time, only the graph encoder and decoder are used, with $z \sim \mathcal{N}(0, I)$.

Figure 2 illustrates the overall architecture. The time-series and graph encoders produce raw embeddings, which are used for posterior inference and decoding, while their normalized versions are used for contrastive alignment. The posterior network maps the concatenated embeddings to a latent distribution, from which samples are drawn via reparameterization. The decoder takes the latent variable together with the graph embedding to generate the time series. The alignment branch enforces consistency between time-series and graph embeddings but does not directly affect the generative path. All components are MLPs. The number of quantile states Q is a hyperparameter controlling the resolution of the graph representation, with the graph encoder taking a flattened Q^2-entry transition matrix as input. We use $Q = 10$ in the main experiments and examine its sensitivity in Section 6. Both encoders produce 128-dimensional embeddings, and the latent dimension is 32. Shared quantile states have fixed meanings and ordering across samples, allowing a common vector encoder rather than requiring permutation-invariant graph pooling. Layer widths are provided in Appendix B.5.

4.2 GENERATIVE PROCESS

Let $\mathcal{G}_{\text{train}} = \{\Phi_{\mathcal{B}}(x_i)\}_{i=1}^{N_{\text{train}}}$ be the training graph pool, and let $\widehat{p}_{\text{train}}(G)$ denote the distribution induced by random sampling from this pool. Generation proceeds as

$$G \sim \widehat{p}_{\text{train}}, \quad Z \sim \mathcal{N}(0, I), \quad \widetilde{X} \sim p_\theta(\cdot \mid G, Z), \tag{4}$$

with Z sampled independently of G. The resulting marginal generation distribution is

$$\begin{aligned} p_\theta^{\text{gen}}(\widetilde{x}) &= \mathbb{E}_{G \sim \widehat{p}_{\text{train}}}\left[p_\theta(\widetilde{x} \mid G)\right] \\ &= \mathbb{E}_{G \sim \widehat{p}_{\text{train}}}\left[\int p_\theta(\widetilde{x} \mid G, z) p(z)\, dz\right]. \end{aligned} \tag{5}$$

Thus, Graph2TS is conditional internally. Graph conditions may in principle be sampled, interpolated, or generated; in this work, we sample them from the training-set graph pool. Multiple latent draws can produce different realizations under one graph. At generation time, only the graph encoder and decoder are used; the time-series encoder and posterior network are used only during training.

4.3 LEARNING OBJECTIVE

We train Graph2TS with a conditional variational objective. Let $q_\phi(Z \mid X, G)$ denote the variational posterior, with learnable parameters ϕ distinct from the fixed construction $\Phi_{\mathcal{B}}$. The core learning objective is the conditional evidence lower bound (ELBO):

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi(Z|X,G)}\big[\log p_\theta(X \mid G, Z)\big] \\ - \mathrm{KL}\big(q_\phi(Z \mid X, G) \,\|\, p(Z)\big).\end{aligned} \tag{6}$$

In practice, the conditional likelihood term is instantiated as a mean-squared reconstruction loss, while the KL term is weighted by a coefficient β to control stochastic regularization. The resulting objective is

$$\begin{aligned}\min_{\theta,\phi}\ \mathcal{L} = \lambda_{\mathrm{rec}}\,\mathcal{L}_{\mathrm{rec}} + \beta\,\mathrm{KL}\big(q_\phi(Z \mid X, G) \,\|\, p(Z)\big) \\ +\lambda_{\mathrm{align}}\,\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{dist}}\,\mathcal{L}_{\mathrm{dist}}.\end{aligned} \tag{7}$$

For a mini-batch of paired samples $\{(x_i, g_i)\}_{i=1}^{B}$, let $\tilde{\mathbf{h}}_i^x$ and $\tilde{\mathbf{h}}_i^g$ denote the ℓ_2-normalized time-series and graph embeddings, respectively. The similarity score is defined as

$$s_{ij} = \frac{(\tilde{\mathbf{h}}_i^x)^\top \tilde{\mathbf{h}}_j^g}{\tau}, \tag{8}$$

where τ is a learnable temperature parameter. The positive pair for x_i is its corresponding graph g_i, while the remaining graphs in the same mini-batch are treated as negatives. This term encourages consistent cross-modal representations between matched graph–time-series pairs. The bidirectional InfoNCE alignment loss is

$$\mathcal{L}_{\mathrm{align}} = -\frac{1}{2B}\sum_{i=1}^{B}\left[\log\frac{\exp(s_{ii})}{\sum_{j=1}^{B}\exp(s_{ij})} + \log\frac{\exp(s_{ii})}{\sum_{j=1}^{B}\exp(s_{ji})}\right]. \tag{9}$$

For the distribution regularizer, let $x_{i,(1)} \le \cdots \le x_{i,(T)}$ and $\hat{x}_{i,(1)} \le \cdots \le \hat{x}_{i,(T)}$ denote the sorted values, i.e., the empirical order statistics, of the real and generated sequences. We define

$$\mathcal{L}_{\mathrm{dist}} = \frac{1}{BT}\sum_{i=1}^{B}\sum_{k=1}^{T}\big(\hat{x}_{i,(k)} - x_{i,(k)}\big)^2. \tag{10}$$

This loss regularizes the marginal amplitude distribution of each reconstructed sequence by matching its empirical order statistics to those of the corresponding real sequence. Thus, $\mathcal{L}_{\mathrm{dist}}$ matches sorted values within each sequence rather than enforcing additional pointwise temporal alignment.

4.4 INTERPRETATION OF STRUCTURE-CONDITIONED GENERATION

We briefly interpret how conditioning on a structural representation G shapes the behavior of Graph2TS.

Proposition 4.1 (Structure–residual decomposition)**.** *Given a time-series random variable $X \in \mathbb{R}^T$ and its deterministic structural representation $G = \Phi_{\mathcal{B}}(X)$, define*

$$f(G) \triangleq \mathbb{E}[X \mid G], \qquad R \triangleq X - f(G). \tag{11}$$

Then

$$X = f(G) + R, \qquad \mathbb{E}[R \mid G] = 0. \tag{12}$$

Here, $f(G)$ is the conditional mean under the data distribution and R is its zero-conditional-mean residual. For a fixed condition, variability around this mean belongs to R. This identity motivates conditional modeling but does not establish that a learned decoder recovers the decomposition exactly.

This induces the variance decomposition

$$\mathrm{Var}(X) = \mathrm{Var}(\mathbb{E}[X \mid G]) + \mathbb{E}[\mathrm{Var}(X \mid G)]\,, \tag{13}$$

where the first term corresponds to structural variability and the second to conditional stochastic variation. A detailed derivation is provided in Appendix B.1.

In Graph2TS, the conditional decoder $p_\theta(x \mid g, z)$ can be interpreted as modeling

$$x \approx f_\theta(g) + r_\theta(g, z), \tag{14}$$

where $f_\theta(g)$ and $r_\theta(g, z)$ describe a graph-dependent mean and stochastic variation conceptually. The decoder is not explicitly factorized into these two networks, and the identity does not guarantee latent disentanglement or exact graph preservation. Because the graph is lossy, temporal ordering and spectral properties not determined by g must be learned from the training sequences and evaluated empirically.

5 Experiments

5.1 Setup

Baselines and datasets. We compare Graph2TS with TimeGAN (Yoon et al., 2019), DiffusionTS (Yuan & Qiao, 2024), and TSFlow (Kollovieh et al., 2025), representing GAN, diffusion, and flow-matching approaches, respectively. We evaluate on CHB–MIT Scalp EEG (CHB-MIT) (Guttag, 2010), Daily Sunspot Number (Sunspot) (World Data Center SILSO, 2019), Electricity Load (Electricity) (Trindade, 2015), and MIT-BIH Arrhythmia ECG (ECG) (Moody & Mark, 2001). The main benchmark uses univariate windows of length $T = 32$; we additionally evaluate longer Electricity windows. Dataset details appear in Appendix B.6.

Generation and evaluation protocol. Graph2TS randomly samples graphs from the training pool and independently draws latent variables from $\mathcal{N}(0, I)$; no test sequence or test-derived graph is supplied to the generator. Graphs are deterministic training-data summaries, not external labels or annotations. Z-score statistics and shared quantile boundaries are fitted to the training split and fixed for evaluation. All methods use the same split and training-fitted normalization, with equal real and synthetic evaluation sample counts. TSFlow uses unconditional complete-window synthesis at $T = 32$, without a test history, forecasting prefix, or other test-specific context. The comparison therefore matches data access, not architectural assumptions: Graph2TS remains structure-conditioned internally, while the mixture in equation 5 yields samples without target-specific information. Further implementation details appear in Appendix B.5.

Metrics. Wasserstein distance and KS assess marginal distributions. ACF MAE compares mean autocorrelation curves, and PSD ℓ_2 distance compares mean Welch power spectra. For each real window, let d_i be its Euclidean distance to the nearest generated window. Prototype error reports the mean and median of d_i; Cov@q is the fraction with $d_i \leq \tau_q$, where τ_q is the q-quantile of real-to-real nearest-neighbor distances ($q = 0.5, 0.9$). MDR is the distance between real and synthetic medoids divided by the mean real-to-medoid distance. Lower is better except for coverage. These complementary metrics distinguish marginal fidelity, temporal summaries, representative samples, and support overlap; no single metric establishes overall generative quality. Formal definitions appear in Appendix B.7.

5.2 Overall Generation Quality

Table 1 compares all four models under this protocol. Graph2TS attains the best values on seven of nine metrics on CHB-MIT and all nine on ECG, and the lowest PSD discrepancy on three of the four datasets. Performance is more mixed on Sunspot and Electricity, where competing methods achieve better results on several metrics.

Overall comparison. Across the four datasets, Graph2TS shows its strongest performance on CHB-MIT and ECG. On CHB-MIT, it outperforms TSFlow in marginal fidelity, ACF, both prototype errors, and both coverage thresholds, while TSFlow achieves lower PSD discrepancy and MDR. On ECG, Graph2TS leads on all nine metrics, with mean prototype error 4.27 versus 72.02 for TSFlow and Cov@0.5 of 0.895 versus zero for all three baselines. Results are more mixed on Sunspot and Electricity. On Sunspot, Graph2TS achieves the lowest Wasserstein, KS, and PSD discrepancies, while DiffusionTS and TSFlow perform better on several temporal, prototype, and coverage metrics.

Table 1: Generation quality at $T = 32$. Graph2TS uses only training-set graphs and samples latent variables from the prior; all methods generate complete windows without test-specific context and are evaluated against held-out time series. Best values within each dataset are bold. Wass: Wasserstein distance; ACF: ACF MAE; PSD: PSD ℓ_2 distance; P-Err: prototype error; MDR: medoid distance ratio; Cov: manifold coverage.

Dataset	Model	Distribution (↓)		Temporal (↓)		Representativeness (↓)			Coverage (↑)	
		Wass	KS	ACF	PSD	P-Err (avg)	P-Err (med)	MDR	Cov@0.5	Cov@0.9
CHB-MIT	Graph2TS	**3.65e-6**	**0.014**	**0.043**	8.05e-10	**2.341**	**2.080**	0.559	**0.984**	**0.991**
	DiffusionTS	1.10e-5	0.153	0.085	4.12e-9	2.647	2.231	0.637	0.972	0.988
	TimeGAN	1.29e-5	0.134	0.085	3.81e-9	3.404	3.058	0.780	0.977	0.989
	TSFlow	4.67e-6	0.039	0.057	**6.24e-10**	2.817	2.543	**0.460**	0.978	0.990
Sunspot	Graph2TS	**5.76**	**0.081**	0.066	**3.36e2**	4.612	4.030	0.579	0.841	0.970
	DiffusionTS	35.49	0.320	0.021	3.74e3	**3.765**	**3.307**	**0.366**	**0.929**	0.995
	TimeGAN	14.74	0.190	0.098	1.05e3	5.108	4.613	0.755	0.806	0.965
	TSFlow	7.233	0.098	**0.017**	4.44e2	3.954	3.57	0.840	0.928	**0.997**
Electricity	Graph2TS	16.01	0.143	0.036	**5.05e2**	4.664	4.176	**0.455**	0.929	0.981
	DiffusionTS	**4.030**	**0.059**	0.027	3.55e3	**3.793**	3.963	0.542	0.889	0.990
	TimeGAN	21.11	0.235	0.064	2.16e3	4.951	4.523	0.730	0.860	0.960
	TSFlow	16.62	0.137	**0.018**	7.81e2	4.231	**3.949**	0.879	**0.938**	**0.994**
ECG	Graph2TS	**0.014**	**0.033**	**0.051**	**0.0059**	**4.27**	**3.760**	**0.498**	**0.895**	**0.961**
	DiffusionTS	0.114	0.201	0.147	0.198	596.0	596.3	549.9	0.000	0.000
	TimeGAN	0.101	0.282	0.172	0.108	249.6	250.0	135.6	0.000	0.000
	TSFlow	0.048	0.170	0.163	0.035	72.02	72.32	145.55	0.000	0.000

On Electricity, Graph2TS achieves the lowest PSD discrepancy and MDR, while DiffusionTS and TSFlow lead on the remaining metrics.

Interpretation across datasets. Overall, explicit graph conditioning provides competitive generation performance with dataset-dependent trade-offs. The dataset analysis in Appendix A.1 offers additional context: ECG exhibits substantially heavier-tailed increments than the smoother Sunspot series, while CHB-MIT also shows pronounced local variability. Graph2TS achieves its strongest relative results on these more irregular signals and more mixed results on Sunspot and Electricity, suggesting that quantile-transition conditioning may be particularly useful when local structural relations remain informative under strong amplitude variation. Qualitative comparisons are provided in Appendix A.2 and Appendix A.3.

5.3 Effect of Sequence Length

We evaluate Electricity at $T \in \{32, 64, 128, 256\}$ using training-set graph conditions and prior sampling throughout. Table 2 compares Graph2TS with DiffusionTS at each length.

Graph2TS shows a larger relative advantage at longer lengths. At $T = 128$, it performs better on eight of the nine metrics, with DiffusionTS retaining higher Cov@0.9. At $T = 256$, Graph2TS performs better on all reported metrics: Wasserstein distance is 14.82 versus 201.3, and PSD discrepancy is 1.77×10^3 versus 6.12×10^5. Coverage at the stricter threshold is 0.361 versus 0.002. However, Graph2TS's own Cov@0.5 decreases from 0.929 at $T = 32$ to 0.361 at $T = 256$, and prototype error increases. Thus, the results show a relative advantage over this baseline in the tested long-sequence setting, alongside reduced coverage at longer windows.

6 Ablation Study

Structural conditioning and stochasticity. Table 3 shows that the full model improves all six reported metrics over both ablated variants. Removing graph conditioning increases ACF error from 0.051 to 0.088 and PSD discrepancy from 0.0059 to 0.0179; removing stochasticity increases them

Table 2: Effect of sequence length on Electricity, using training-set conditioning graphs and prior sampling for Graph2TS. The $T = 32$ rows reproduce Table 1. Best values at each length are bold. Metric definitions and directions follow Table 1.

T	Model	Wass	KS	ACF	PSD	P-Err (avg)	P-Err (med)	MDR	Cov@0.5	Cov@0.9
32	Graph2TS	16.01	0.143	0.036	**5.05e2**	4.664	4.176	**0.455**	**0.929**	0.981
	DiffusionTS	**4.030**	**0.059**	**0.027**	3.55e3	**3.793**	**3.963**	0.542	0.889	**0.990**
64	Graph2TS	13.32	0.117	**0.034**	**1.29e3**	**6.099**	**5.643**	**0.612**	**0.760**	0.967
	DiffusionTS	**5.277**	**0.047**	0.054	2.64e3	7.186	6.808	1.620	0.6285	**0.994**
128	Graph2TS	**14.07**	**0.130**	**0.051**	**1.76e3**	**6.742**	**6.357**	**0.789**	**0.719**	0.983
	DiffusionTS	36.64	0.183	0.089	9.16e4	7.744	7.561	1.897	0.431	**0.993**
256	Graph2TS	**14.82**	**0.140**	**0.065**	**1.77e3**	**8.430**	**8.200**	**1.132**	**0.361**	**0.9785**
	DiffusionTS	201.3	0.448	0.195	6.12e5	13.57	13.51	13.72	0.002	0.406

Table 3: Ablation study on ECG dataset. We compare the full model with (i) identity graph conditioning (w/o graph) and (ii) a deterministic Graph2TS model without stochastic latent variables. The full-model row matches Table 1. Lower is better (↓), higher is better (↑).

Model	**Wass.** ↓	**KS** ↓	**ACF MAE** ↓	**PSD L2** ↓	**ProtoErr** ↓	**Coverage@**$\tau_{0.5}$ ↑
Full (Graph2TS)	**0.014**	**0.033**	**0.051**	**0.0059**	**4.27**	**0.895**
w/o graph	0.049	0.125	0.088	0.0179	4.41	0.875
w/o stochasticity	0.078	0.228	0.084	0.0310	5.26	0.752

Table 4: Ablation study on CHB-MIT and Sunspot datasets. Each ablated row removes a single loss component; full-model rows match Table 1. Metrics: **Wass**: Wasserstein distance, **KS**: KS statistic, **ACF**: ACF MAE, **PSD**: PSD L2 distance, **P-Err**: ProtoErr, **Cov**: Coverage@$\tau_{0.5}$. (↓: Lower is better, ↑: Higher is better).

Dataset	**Ablation**	**Distribution** (↓)		**Temporal** (↓)		**Representativeness** (↓)	**Coverage** (↑)
		Wass	**KS**	**ACF**	**PSD**	**P-Err(avg)**	**Cov@0.5**
CHB-MIT	Full model	3.65e-6	0.014	**0.043**	8.05e-10	2.341	0.984
	$\mathcal{L}_{recon} = 0$	**2.73e-6**	**0.013**	0.203	1.00e-9	4.478	0.711
	$\mathcal{L}_{align} = 0$	3.91e-6	0.024	0.044	**7.08e-10**	2.310	**0.990**
	$\mathcal{L}_{dist} = 0$	4.22e-6	0.021	0.051	9.50e-10	**2.302**	**0.990**
	$\beta_{\mathrm{KL}} = 0$	5.64e-5	0.265	0.115	2.18e-8	6.288	0.382
Sunspot	Full model	5.76	**0.081**	**0.066**	**3.36e2**	4.612	**0.841**
	$\mathcal{L}_{recon} = 0$	**1.709**	0.082	0.179	3.69e3	8.300	0.282
	$\mathcal{L}_{align} = 0$	2.605	0.112	**0.066**	9.21e2	**4.482**	0.831
	$\mathcal{L}_{dist} = 0$	2.510	0.107	0.074	4.88e2	4.594	0.825
	$\beta_{\mathrm{KL}} = 0$	29.78	0.219	0.117	1.09e4	7.297	0.398

to 0.084 and 0.0310, respectively. Both ablations also worsen marginal fidelity, prototype error, and coverage, supporting the complementary roles of structural conditioning and stochastic variation in the ECG setting. Figure 3 provides corresponding PSD and embedding-space comparisons, with the complete qualitative analysis including ACF provided in Appendix A.4.

Loss components. Table 4 reports loss ablations on CHB-MIT and Sunspot. Removing reconstruction worsens ACF, PSD, prototype error, and coverage on both datasets despite improved marginal fidelity, while removing KL regularization degrades all reported metrics. In contrast, removing the alignment or distribution loss produces smaller and metric-dependent changes. Together with the component ablations, these results indicate that the main gains come from structure-conditioned generation via the quantile graph, while the auxiliary losses provide incremental regularization.

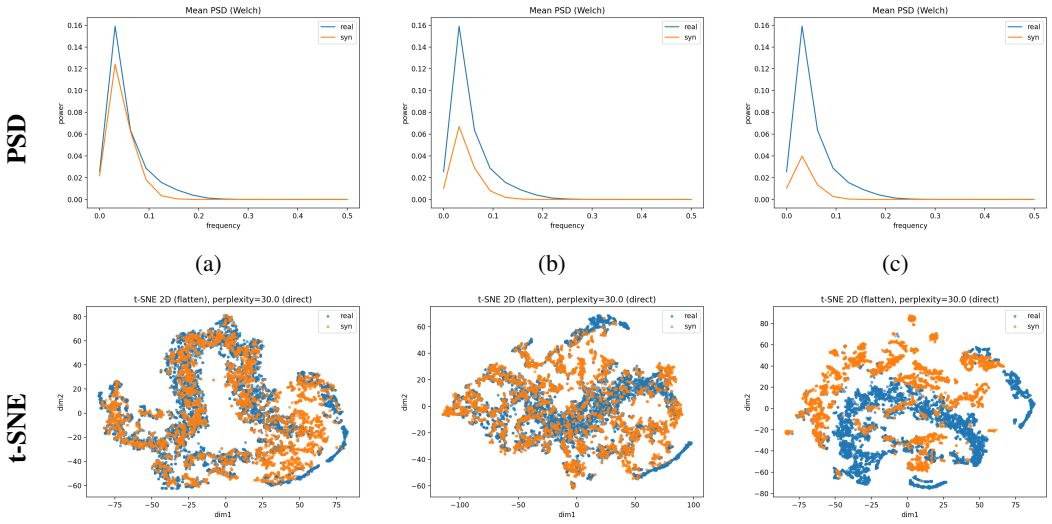


Figure 3: Qualitative ablation analysis on ECG. Rows show PSD and t-SNE; columns show the full model, w/o graph conditioning, and w/o stochasticity. The complete analysis including ACF is provided in Appendix A.4.

Table 5: Sensitivity analysis of the number of quantile bins Q on the ECG dataset. The $Q = 10$ row matches the main benchmark. Lower is better for Distribution, Temporal, and Representativeness; higher is better for Coverage.

Dataset	Q	Distribution (↓)		Temporal (↓)		Representativeness (↓)	Coverage (↑)
		Wass	KS	ACF	PSD	P-Err(avg)	Cov@0.5
ECG	5	0.019	0.043	0.054	8.18e-3	3.960	0.916
	8	0.015	0.038	0.054	8.81e-3	3.917	0.923
	10	0.014	0.033	0.051	0.0059	4.27	0.895
	12	0.012	0.032	0.060	9.54e-3	3.874	0.926
	15	0.015	0.037	0.057	1.05e-2	3.883	0.932
	20	0.011	0.026	0.055	1.35e-2	4.031	0.902

Quantile-bin sensitivity.

Table 5 varies Q on ECG at $T = 32$. Performance remains broadly stable across $Q = 5$–20, with no monotonic trend as the graph resolution increases. Different values yield small metric-specific trade-offs, suggesting that Graph2TS is not highly sensitive to Q in this setting.

7 Conclusion

Graph2TS combines quantile-graph conditioning with latent stochastic variation for time-series generation. Under a matched generation protocol without test-specific conditioning, Graph2TS shows strong fidelity with dataset-dependent trade-offs across GAN, diffusion, and flow-matching baselines. Ablation and sensitivity analyses further support the complementary roles of structural conditioning and stochastic variation and show limited sensitivity to graph resolution in the tested setting. These results establish quantile graphs as a useful instance of structural conditioning and motivate learned graph priors, time-local or multiscale structural representations, and broader long-sequence evaluation as directions for future work.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, and the refinement of mathematical exposition and presentation. They were not used to generate experimental data, run experiments, select results, or make final scientific conclusions. All AI-assisted text was reviewed and revised by the authors to ensure its accuracy and consistency with the underlying research. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide additional implementation details and experimental settings in the Appendix to facilitate reproducibility. The implementation will also be made publicly available.

REFERENCES

Nourhan Ahmed and Lars Schmidt-Thieme. Sparse self-attention guided generative adversarial networks for time-series generation. *International Journal of Data Science and Analytics*, 16(4): 421–434, 2023.

Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. Generative adversarial networks in time series: A survey and taxonomy. *CoRR*, abs/2107.11098, 2021. URL https://arxiv.org/abs/2107.11098.

Andriana SLO Campanharo, M Irmak Sirer, R Dean Malmgren, Fernando M Ramos, and Luís A Nunes Amaral. Duality between time series and networks. *PloS one*, 6(8):e23378, 2011.

Hongjie Chen and Hoda Eldardiry. Graph time-series modeling in deep learning: a survey. *ACM Transactions on Knowledge Discovery from Data*, 18(5):1–35, 2024.

Andrea Coletta, Sriram Gopalakrishnan, Daniel Borrajo, and Svitlana Vyetrenko. On the constrained time-series generation problem. *Advances in Neural Information Processing Systems*, 36:61048–61059, 2023.

Tal Gonen, Itai Pemper, Ilan Naiman, Nimrod Berman, and Omri Azencot. Time series generation under data scarcity: A unified generative modeling approach. *CoRR*, abs/2505.20446, 2025. doi: 10.48550/ARXIV.2505.20446. URL https://doi.org/10.48550/arXiv.2505.20446.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11): 139–144, 2020. doi: 10.1145/3422622. URL https://doi.org/10.1145/3422622.

John Guttag. CHB-MIT Scalp EEG Database. *PhysioNet*, June 2010. doi: 10.13026/C2K01R. URL https://doi.org/10.13026/C2K01R. Version 1.0.0.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=uxVBbSlKQ4.

Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuño. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008. doi: 10.1073/pnas.0709247105. URL https://www.pnas.org/doi/abs/10.1073/pnas.0709247105.

Hongming Li, Shujian Yu, and Jose Principe. Causal recurrent variational autoencoder for medical time series generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8562–8570, 2023.

Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion models for time-series applications: a survey. *Frontiers Inf. Technol. Electron. Eng.*, 25(1):19–41, 2024. doi: 10.1631/FITEE.2300310. URL https://doi.org/10.1631/FITEE.2300310.

Andriana Susana Lopes de Oliveira Campanharo and Fernando Manuel Ramos. Quantile graphs for the characterization of chaotic dynamics in time series. In *2015 Third World Conference on Complex Systems (WCCS)*, pp. 1–4, 2015. doi: 10.1109/ICoCS.2015.7483302.

Chang Lu, Chandan K. Reddy, Ping Wang, Dong Nie, and Yue Ning. Multi-label clinical time-series generation via conditional GAN. *IEEE Trans. Knowl. Data Eng.*, 36(4):1728–1740, 2024. doi: 10.1109/TKDE.2023.3310909. URL https://doi.org/10.1109/TKDE.2023.3310909.

B. Luque, L. Lacasa, F. Ballesteros, and J. Luque. Horizontal visibility graphs: Exact results for random time series. *Phys. Rev. E*, 80:046103, Oct 2009. doi: 10.1103/PhysRevE.80.046103. URL https://link.aps.org/doi/10.1103/PhysRevE.80.046103.

G.B. Moody and R.G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001. doi: 10.1109/51.932724.

Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep P. Chinchali. Time weaver: A conditional time series generation model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=WpKDeixmFr.

Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *CoRR*, abs/2006.05421, 2020. URL https://arxiv.org/abs/2006.05421.

Khalid Oublal, Saïd Ladjal, David Benhaiem, Emmanuel Le-borgne, and François Roueff. Disentangling time series representations via contrastive independence-of-support on l-variational inference. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=iI7hZSczxE.

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*, pp. 8857–8868. PMLR, 2021.

Vanessa Freitas Silva, Maria Eduarda Silva, Pedro Ribeiro, and Fernando Silva. Time series analysis via network science: Concepts and algorithms. *WIREs Data Mining and Knowledge Discovery*, 11 (3), March 2021. ISSN 1942-4795. doi: 10.1002/widm.1404. URL http://dx.doi.org/10.1002/widm.1404.

Vanessa Freitas Silva, Maria Eduarda Silva, Pedro Ribeiro, and Fernando Silva. Multilayer quantile graph for multivariate time series analysis and dimensionality reduction, 2023. URL https://arxiv.org/abs/2311.11849.

Kaleb E. Smith and Anthony O. Smith. Conditional GAN for timeseries generation. *CoRR*, abs/2006.16477, 2020. URL https://arxiv.org/abs/2006.16477.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

World Data Center SILSO. Daily sunspot number, 1818–2019. https://www.sidc.be/silso/, 2019. Royal Observatory of Belgium.

Chao Yang, Yakun Chen, Zihao Li, Xianzhi Wang, Kaize Shi, Lina Yao, Guandong Xu, and Zhongwen Guo. Deep multimodal learning for time series analysis in social computing: a survey. *Int. J. Multim. Inf. Retr.*, 14(2):15, 2025. doi: 10.1007/S13735-025-00363-X. URL https://doi.org/10.1007/s13735-025-00363-x.

Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, et al. A survey on diffusion models for time series and spatio-temporal data. *ACM Computing Surveys*, 58(8):1–39, 2026.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5509–5519, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html.

Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 8980–8987. AAAI Press, 2022. doi: 10.1609/AAAI.V36I8.20881. URL https://doi.org/10.1609/aaai.v36i8.20881.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y. Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, and Shirui Pan. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46 (10):6775–6794, 2024. doi: 10.1109/TPAMI.2024.3387317. URL https://doi.org/10.1109/TPAMI.2024.3387317.

Yong Zou, Reik V. Donner, Norbert Marwan, Jonathan F. Donges, and Jürgen Kurths. Complex network approaches to nonlinear time series analysis. *Physics Reports*, 787:1–97, January 2019. ISSN 0370-1573. doi: 10.1016/j.physrep.2018.10.005. URL http://dx.doi.org/10.1016/j.physrep.2018.10.005.

A Appendix: Additional Experimental Results and Analysis

A.1 Distributional Characteristics of Sunspot and ECG

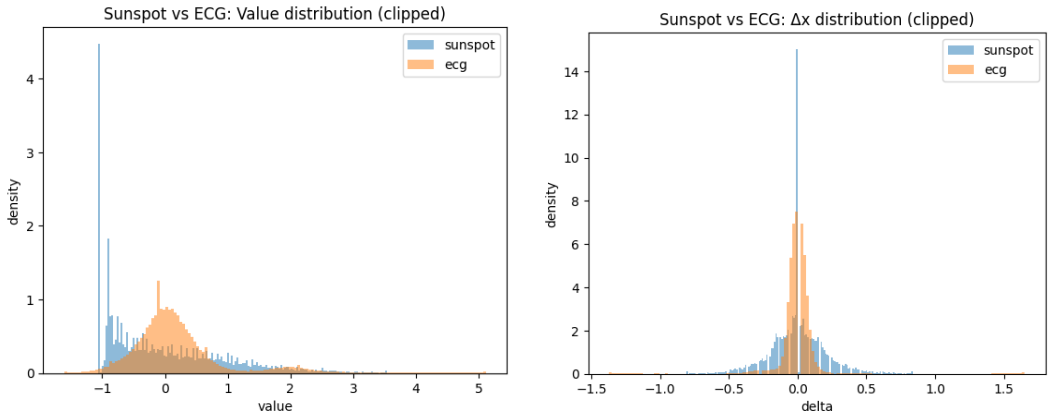
Table 6: Distribution statistics in **z-score space** for Sunspot vs ECG windows (length $T = 32$). We report summary statistics for raw values x and first-order differences Δx. Excess kurtosis indicates tail-heaviness (larger $\Rightarrow$ heavier tails).

Dataset	B	x **statistics**				Δx **statistics**		
		mean	std	excess kurt.	quantile range (0.1% $\rightarrow$ 99.9%)	mean	std	excess kurt.
Sunspot	1758	-0.0360	0.9668	0.5321	[-1.0741, 3.5320]	-7.5e-05	0.1994	2.2466
ECG	20312	0.2148	0.8283	7.3562	[-1.6078, 5.1106]	4.4e-05	0.1819	41.5587

Table 6 and Fig. 4 summarize the distributions of z-score normalized windows. Sunspot has low value-distribution excess kurtosis (0.53) and moderately heavy-tailed increments (2.25). ECG has heavier tails in both values (7.36) and increments (41.56), indicating more frequent abrupt changes in the sampled windows. These descriptive differences help characterize the two benchmarks. They do not, by themselves, establish why a particular generator succeeds or fails, or imply that heavy-tailed target data violate the assumptions of diffusion models. The comparisons in Table 1 provide the empirical evidence for the methods' performance on these datasets.

A.2 Qualitative Comparison of Real and Generated Samples

Figure 5 shows ten generated sequences for each of three training-set graph conditions, with independent latent samples. The real sequence in each row is the training sequence from which that conditioning graph is constructed; it is not a test sequence supplied at generation time. Figure 6 compares real and generated sequences and their quantile graphs.



(a) Value distribution $p(x)$ (clipped for visualization). (b) First-order difference distribution $p(\Delta x)$ (clipped).

Figure 4: Sunspot vs ECG in **z-score space**. ECG exhibits substantially heavier tails, especially in Δx, indicating more frequent abrupt transitions compared to the smoother Sunspot dynamics.

These examples illustrate the variation possible under a shared graph condition. Short-window graphs can contain localized transitions and loops, but the examples do not establish accurate transition-probability estimation from only $T - 1$ observations. Generated graphs can also differ from their conditions: the learned decoder is not constrained to reproduce every transition weight exactly. We therefore use these plots as qualitative illustrations and assess fidelity with the quantitative metrics in the main text.

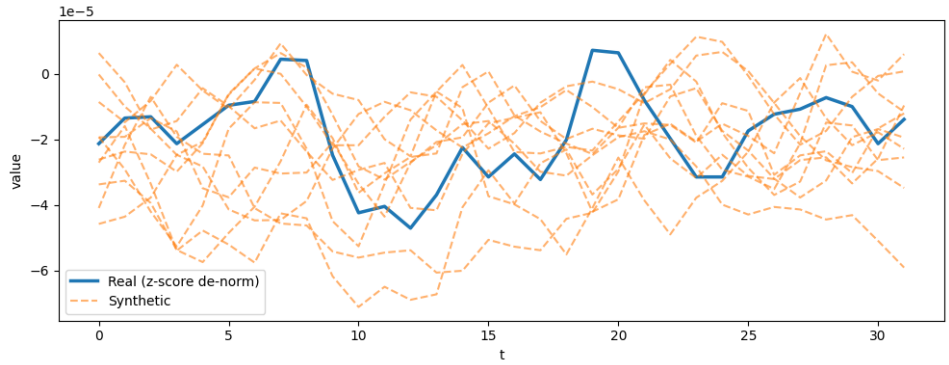
A.3 Qualitative Analysis of Temporal and Distributional Structure

Figures 7, 8, and 9 compare mean ACF, mean PSD, and t-SNE embeddings across datasets. Graph2TS samples use training-set graph conditions. The ACF and PSD plots illustrate temporal and spectral differences, while the embedding plots visualize real and generated samples in two dimensions. Such projections provide qualitative context rather than a direct measure of support coverage. These figures compare Graph2TS with TimeGAN and DiffusionTS; TSFlow is included in the quantitative main benchmark, not these visualizations. In Table 1, Graph2TS has the lowest PSD discrepancy on Sunspot, Electricity, and ECG, while TSFlow has the lowest on CHB-MIT. ACF performance is dataset-dependent, with TSFlow achieving the lowest error on Sunspot and Electricity at $T = 32$.

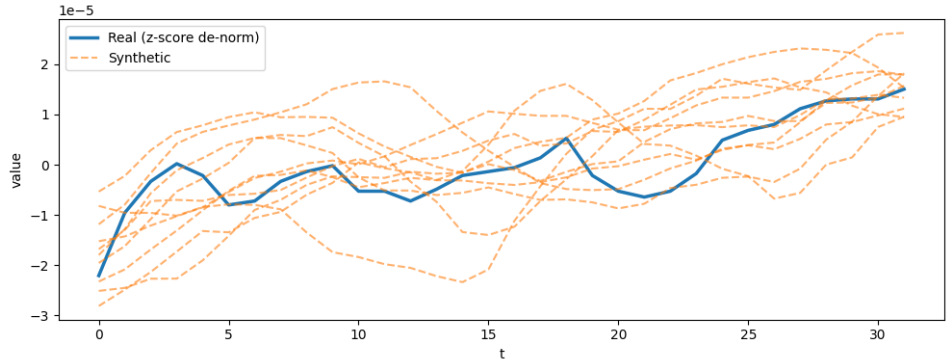
A.4 Additional Ablation Results

Structural conditioning and stochasticity. Figure 10 supplements Table 3 with ACF, PSD, and embedding plots for the full model and two ablations. The full model improves all six reported metrics over both ablated variants in Table 3, including ACF and PSD discrepancies. The deterministic architecture is shown in Appendix B.3.

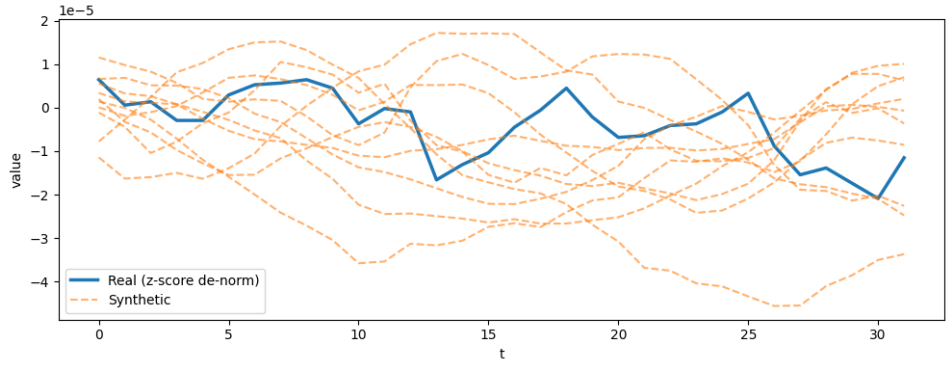
The quantitative loss-component and quantile-resolution studies are reported in the main text (Tables 4 and 5), so that the evidence for these design choices can be assessed without relying on the appendix.



(a) Example 1.

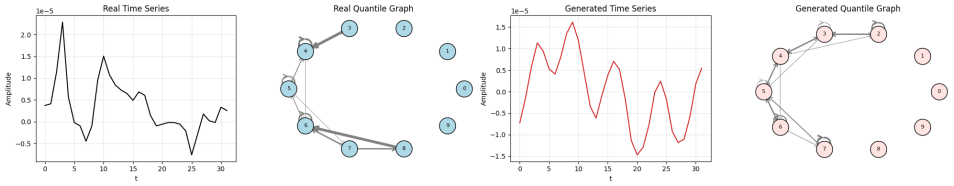


(b) Example 2.

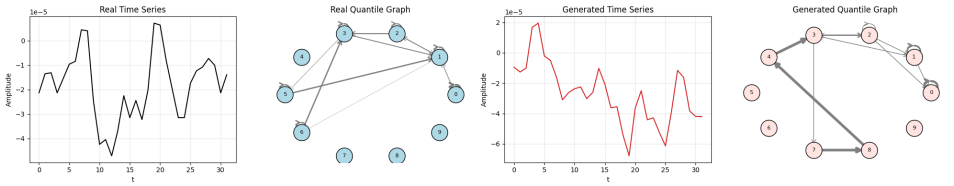


(c) Example 3.

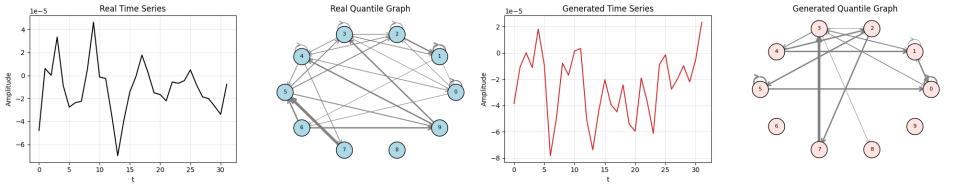
Figure 5: Training sequences and corresponding generated examples. Each row displays ten samples generated from one training-set quantile graph using independent latent samples. Different rows use different graph conditions.



(a) Sample 1: Comparison of real (left) and generated (right) time series and graphs.

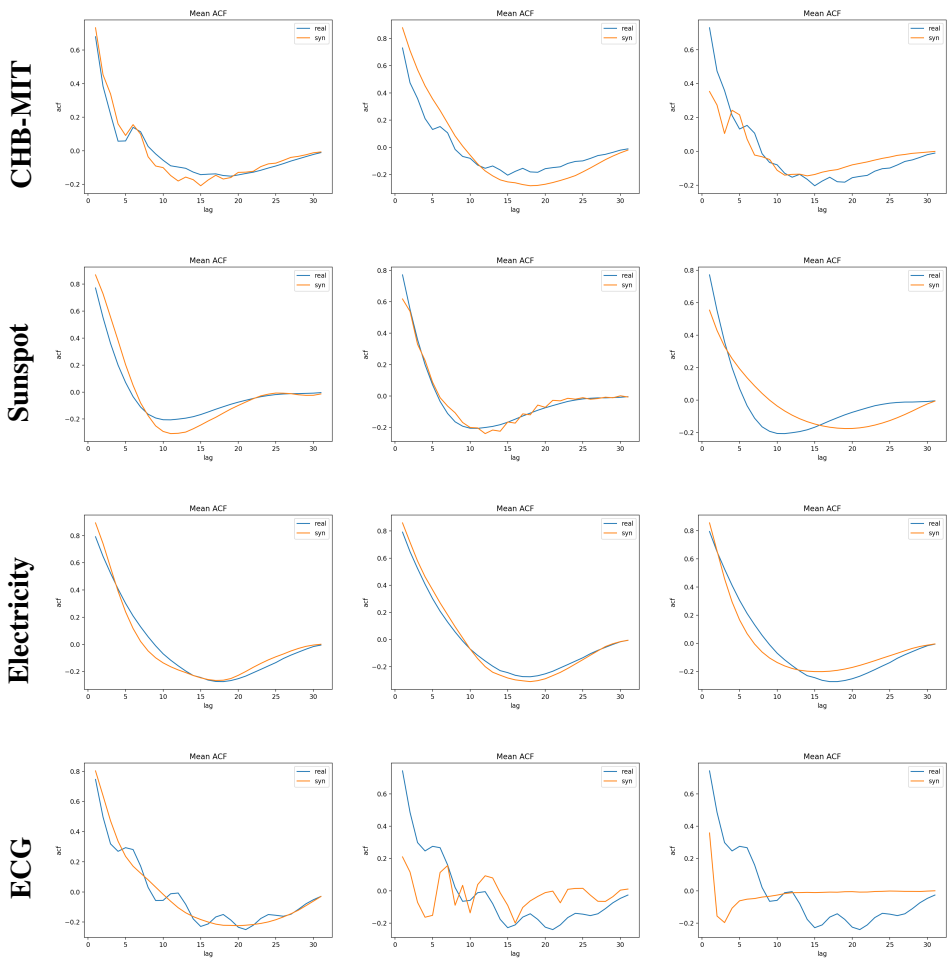


(b) Sample 2: Comparison of real (left) and generated (right) time series and graphs.



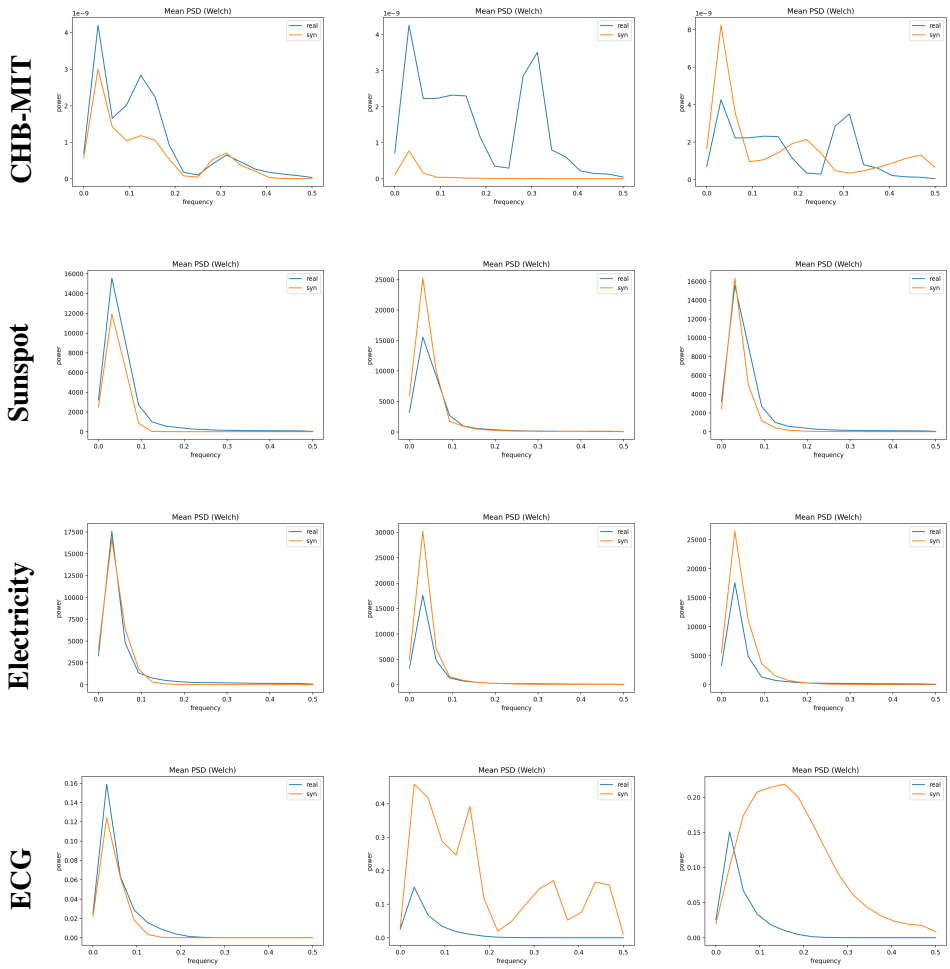
(c) Sample 3: Comparison of real (left) and generated (right) time series and graphs.

Figure 6: Training and generated time series and quantile graphs for three conditions. Each row shows the training sequence and its conditioning graph (left), together with a generated sequence and its extracted graph (right). The examples illustrate conditional variation rather than exact graph reconstruction.



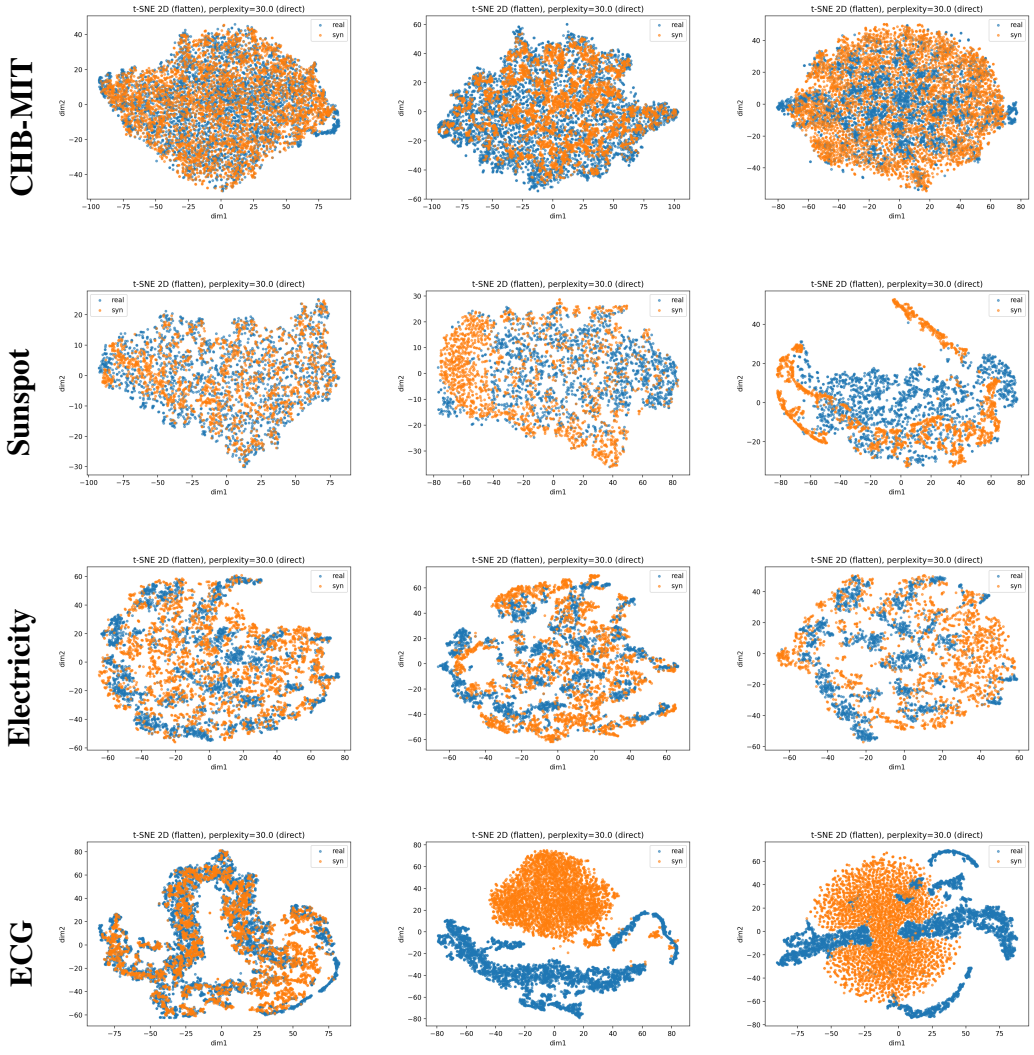
(a) Graph2TS (ours) **(b) DiffusionTS** **(c) TimeGAN**

Figure 7: Mean autocorrelation functions (ACF) on four datasets. Rows: CHB-MIT, Sunspot, Electricity, and ECG; columns: Graph2TS, DiffusionTS, and TimeGAN. Graph2TS samples use training-set graph conditions. Temporal agreement varies by dataset; aggregate ACF errors are reported in Table 1.



(a) Graph2TS (ours) **(b) DiffusionTS** **(c) TimeGAN**

Figure 8: Mean power spectral densities (PSD) on four datasets. Rows: CHB-MIT, Sunspot, Electricity, and ECG; columns: Graph2TS, DiffusionTS, and TimeGAN. Graph2TS samples use training-set graph conditions. Table 1 reports the corresponding spectral discrepancies.



(a) Graph2TS (ours) **(b) DiffusionTS** **(c) TimeGAN**

Figure 9: t-SNE visualizations on four datasets. Rows: CHB-MIT, Sunspot, Electricity, and ECG. Red and blue points denote real and generated samples, respectively. Graph2TS samples use training-set graph conditions. Two-dimensional overlap is a qualitative illustration; quantitative coverage is reported in Table 1.

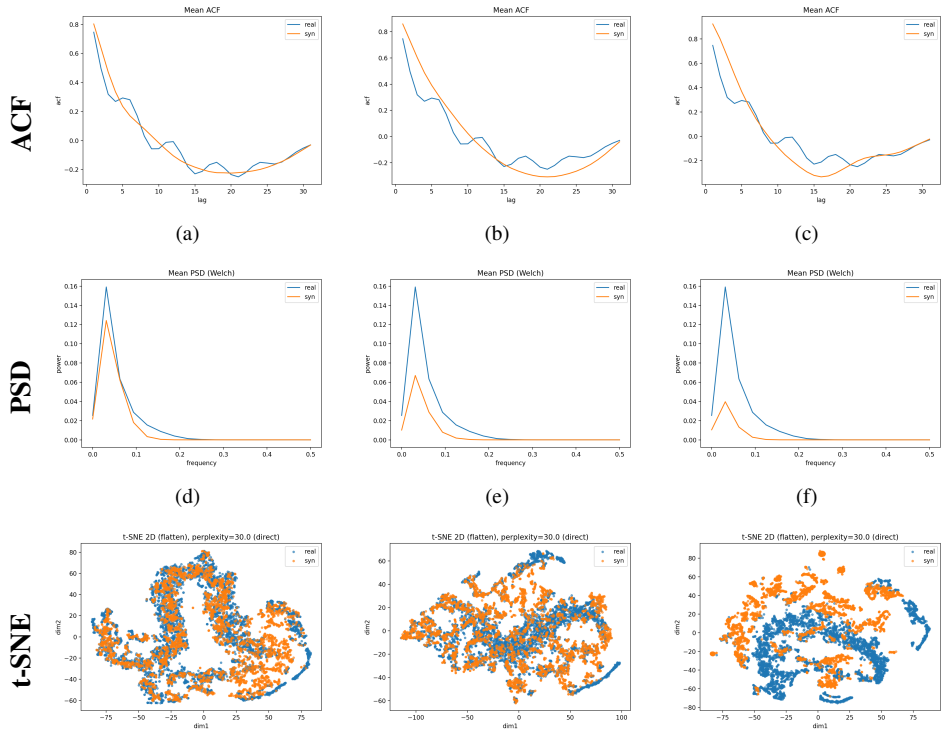


Figure 10: Ablation analysis on ECG. Rows: ACF, PSD, and t-SNE. Columns: full model, w/o graph, and w/o stochasticity.

B Appendix: Theoretical and Implementation Details

B.1 Derivation of Variance Decomposition

Let $\mu(G) \triangleq \mathbb{E}[X \mid G]$. By definition,

$$\mathrm{Var}(X) = \mathbb{E}\big[\|X - \mathbb{E}[X]\|^2\big]. \tag{15}$$

Insert and subtract $\mu(G)$:

$$X - \mathbb{E}[X] = \big(X - \mu(G)\big) + \big(\mu(G) - \mathbb{E}[X]\big). \tag{16}$$

Therefore,

$$\mathrm{Var}(X) = \mathbb{E}\Big[\big\|(X - \mu(G)) + (\mu(G) - \mathbb{E}[X])\big\|^2\Big] \tag{17}$$

$$= \mathbb{E}\big[\|X - \mu(G)\|^2\big] + \mathbb{E}\big[\|\mu(G) - \mathbb{E}[X]\|^2\big] + 2\,\mathbb{E}\big[(X - \mu(G))^\top(\mu(G) - \mathbb{E}[X])\big]. \tag{18}$$

The cross term is zero:

$$\mathbb{E}\big[(X - \mu(G))^\top(\mu(G) - \mathbb{E}[X])\big] = \mathbb{E}\big[\mathbb{E}\big[(X - \mu(G))^\top(\mu(G) - \mathbb{E}[X]) \mid G\big]\big] \tag{19}$$

$$= \mathbb{E}\big[(\mu(G) - \mathbb{E}[X])^\top \mathbb{E}[X - \mu(G) \mid G]\big] \tag{20}$$

$$= \mathbb{E}\big[(\mu(G) - \mathbb{E}[X])^\top \cdot 0\big] = 0, \tag{21}$$

because $\mathbb{E}[X - \mu(G) \mid G] = \mathbb{E}[X \mid G] - \mu(G) = 0$. Hence,

$$\mathrm{Var}(X) = \mathbb{E}\big[\|X - \mu(G)\|^2\big] + \mathbb{E}\big[\|\mu(G) - \mathbb{E}[X]\|^2\big]. \tag{22}$$

Finally, observe that

$$\mathbb{E}\big[\|X - \mu(G)\|^2\big] = \mathbb{E}[\mathrm{Var}(X \mid G)], \qquad \mathbb{E}\big[\|\mu(G) - \mathbb{E}[X]\|^2\big] = \mathrm{Var}(\mathbb{E}[X \mid G]), \tag{23}$$

which yields the law of total variance:

$$\mathrm{Var}(X) = \mathrm{Var}(\mathbb{E}[X \mid G]) + \mathbb{E}[\mathrm{Var}(X \mid G)]. \tag{24}$$

B.2 Why not graph neural networks?

Although the quantile graph admits a graph interpretation, it is fundamentally a fixed-size transition matrix with globally aligned semantics across samples. Each node corresponds to a shared quantile state, and each edge weight represents a normalized transition probability.

In this setting, the primary modeling objective is to encode global transition statistics rather than local neighborhood structure. We therefore adopt a simple MLP encoder to embed the quantile graph, which is sufficient to capture these statistics while avoiding the additional complexity and inductive assumptions introduced by graph neural networks.

B.3 Deterministic Graph to Time Series Model

Fig. 11 shows the architecture of the deterministic version of the **Graph2TS**.

B.4 Quantile Boundary Estimation

Quantile boundaries are estimated once from the training split and then fixed for all validation, test, and generated samples. Specifically, we collect all scalar observations from the training time-series windows and compute the empirical quantiles

$$b_k = \mathrm{Quantile}\left(\{x_{i,t} : x_i \in \mathcal{D}_{\text{train}},\ t = 1, \ldots, T\}, \frac{k}{Q}\right), \qquad k = 0, \ldots, Q.$$

The resulting boundary set $\mathcal{B} = \{b_0, \ldots, b_Q\}$ defines Q globally shared amplitude-rank bins. Using training-set boundaries avoids information leakage from the test set and ensures that each quantile state has consistent semantics across samples. After estimating $\mathcal{B}$, every time-series value $x_{i,t}$ is assigned to a discrete state according to equation 2.

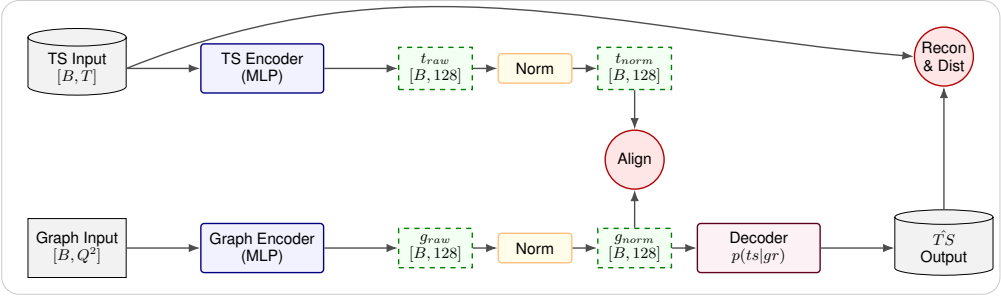


Figure 11: Deterministic graph to time series model (Graph2TS without stochasticity).

B.5 Detailed Setup

Hardware and platform All experiments were conducted on a single NVIDIA A40 GPU.

Baselines. We compare against representative time-series generators: **TimeGAN** (GAN-based) and **DiffusionTS** (diffusion-based). These two baselines use their official implementations. We additionally compare with **TSFlow** (Kollovieh et al., 2025), a flow-matching model with Gaussian-process priors. TSFlow uses unconditional complete-window generation at $T = 32$, without test history, a forecasting prefix, or other test-specific context. All three baselines use the same data split, training-fitted normalization, and evaluation sample count as Graph2TS. The length experiment compares Graph2TS and DiffusionTS only.

Training protocol and fairness. When computing metrics, we generate the same number of synthetic samples as the number of real samples in the evaluation split (4000 sequences) and apply balanced subsampling if needed. Note that while the sorting permutation itself is non-differentiable, standard auto-differentiation frameworks propagate gradients through the sorted values. This allows the order-statistics loss to effectively guide the generator towards matching the marginal amplitude distribution. Training uses Adam with learning rate 3×10^{-4}, batch size 4096. Reconstruction, alignment, and distribution loss weights are 5, 1, and 1; the KL weight increases linearly to 0.05 over 50 epochs. We select the checkpoint with the lowest average training objective.

Sampling protocol. During training, the time-series and graph encoders process paired training sequences and their deterministic quantile graphs. During generation, conditioning graphs are randomly sampled from the training graph pool, and independent $z \sim \mathcal{N}(0, I)$ samples are supplied to the decoder. The reported experiments do not use interpolated graphs. Neither test time series nor test-derived graphs are supplied to the generator, and the variational posterior is not used for generation. Unless stated otherwise, we generate one sequence per selected graph condition and match the real and synthetic evaluation sample counts. The same training-graph protocol is used for the Electricity length experiments.

Other experimental settings. The main benchmark uses univariate windows of length $T = 32$; the Electricity length experiment additionally uses $T = 64, 128, 256$. We condition generation on quantile-transition graphs with $Q = 10$ global bins, represented as a flattened first-order transition matrix in $\mathbb{R}^{Q^2}$. Training minimizes a weighted sum of (i) a symmetric InfoNCE alignment loss between time-series and graph embeddings with a learnable temperature initialized to 0.07, (ii) an ℓ_2 reconstruction loss $\|x - \hat{x}\|_2^2$, (iii) an order-statistics matching loss $\|\mathrm{sort}(\hat{x}) - \mathrm{sort}(x)\|_2^2$, and (iv) a KL regularizer $\mathrm{KL}(q_\phi(z \mid x, g) \,\|\, \mathcal{N}(0, I))$ with linear KL annealing. Unless otherwise stated, we set $w_{\text{align}} = 1$, $w_{\text{recon}} = 5$, $w_{\text{dist}} = 1$, and $\beta_{\max} = 0.05$ with a warmup of 50 epochs. Optimization is performed using Adam with learning rate 3×10^{-4} and batch size 4096, training for 300 epochs; we select the checkpoint with the lowest average training objective and report the corresponding parameters.

Architecture details. All components in Graph2TS are implemented as multi-layer perceptrons (MLPs). For window length T, the time-series encoder maps $x \in \mathbb{R}^T$ to a 128-dimensional raw embedding through

$$T \to 256 \to 128,$$

with a LeakyReLU activation of negative slope 0.2 after the first linear layer. The graph encoder maps the flattened graph $g \in \mathbb{R}^{100}$ to a 128-dimensional raw embedding using

$$100 \to 256 \to 128.$$

The raw embeddings are used for posterior inference and decoding, while their ℓ_2-normalized versions are used only for contrastive alignment.

The posterior network $q_\phi(z \mid x, g)$ takes the concatenation of the raw time-series and graph embeddings as input, resulting in a 256-dimensional vector. It is parameterized by

$$256 \to 256 \to 2d_z,$$

with a LeakyReLU activation after the first linear layer, and outputs $(\mu, \log \sigma^2) \in \mathbb{R}^{2d_z}$ with $d_z = 32$. The decoder $p_\theta(x \mid g, z)$ takes the concatenation of the 128-dimensional raw graph embedding and the latent variable $z \in \mathbb{R}^{32}$ as input. It is parameterized by

$$160 \to 256 \to 512 \to T,$$

with LeakyReLU activations after the first two linear layers and a final linear output layer. At $T = 32$ and $Q = 10$, the total number of trainable parameters is 371,553. The input and output dimensions depend on T; the graph dimension remains Q^2.

B.6 DATASETS

The main benchmark uses fixed-length windows of $T = 32$. Electricity is additionally evaluated with longer windows as described in Section 5.3. Z-score means and standard deviations are fitted to the training split and then fixed for validation and test transformations. Shared quantile boundaries are likewise fitted using only training data.

CHB–MIT Scalp EEG (CHB-MIT). The CHB–MIT Scalp EEG Database (Guttag, 2010) consists of long-term scalp EEG recordings from 22 pediatric subjects with intractable epilepsy. The recordings span multiple days per subject and include expert annotations of seizure onset and offset. Following standard practice, we segment the continuous EEG signals into fixed-length windows of $T = 32$ and apply z-score normalization with statistics fitted to the training split. This dataset is used to evaluate the ability of generative models to capture noisy biomedical signals and to support downstream seizure-related tasks under class imbalance.

Daily Sunspot Number (Sunspot). The Daily Sunspot Number dataset (World Data Center SILSO, 2019) records historical measurements of solar activity from 1818 to 2019. The data exhibit strong long-term periodicity and non-stationary temporal dynamics. We extract overlapping windows of length $T = 32$ from the normalized daily series to evaluate the model's ability to preserve global oscillatory structure while maintaining realistic variability.

Electricity Load (Elec). The Electricity Load dataset (Trindade, 2015) contains electricity consumption measurements from multiple clients over time. The data are characterized by seasonal patterns, abrupt changes, and heterogeneous dynamics across users. We follow common preprocessing procedures and segment each univariate series into windows of length $T = 32$, which are normalized using the training-split z-score statistics.

MIT-BIH Arrhythmia ECG (ECG). The MIT-BIH Arrhythmia Database (Moody & Mark, 2001) provides ECG recordings annotated with cardiac arrhythmia events. The signals contain structured low-frequency morphology combined with high-frequency noise and beat-level variability. We extract fixed-length ECG segments of $T = 32$ and apply z-score normalization using training-split statistics, using this dataset to assess structural fidelity and temporal alignment in physiological signal generation.

B.7 METRICS DEFINITION

Let $\mathcal{X} = \{x_i\}_{i=1}^N$ denote a set of real time series samples and $\hat{\mathcal{X}} = \{\hat{x}_j\}_{j=1}^N$ denote an equally sized set of synthetic samples, where each $x \in \mathbb{R}^T$.

Distribution fidelity. Let μ and $\hat{\mu}$ be the empirical distributions of scalar amplitudes pooled over all windows and time points in $\mathcal{X}$ and $\hat{\mathcal{X}}$, respectively. The **Wasserstein distance** is the one-dimensional Wasserstein-1 distance between these amplitude distributions:

$$W_1(\mu, \hat{\mu}) = \inf_{\gamma \in \Pi(\mu, \hat{\mu})} \mathbb{E}_{(u,v)\sim\gamma}\big[|u - v|\big], \tag{25}$$

where $\Pi(\mu, \hat{\mu})$ is the set of all couplings with marginals μ and $\hat{\mu}$.

The **Kolmogorov–Smirnov (KS) statistic** is defined as

$$\mathrm{KS} = \sup_{v \in \mathbb{R}} \big|F_{\mathcal{X}}(v) - F_{\hat{\mathcal{X}}}(v)\big|, \tag{26}$$

where $F_{\mathcal{X}}$ and $F_{\hat{\mathcal{X}}}$ are the empirical cumulative distribution functions of real and synthetic samples, respectively.

Temporal structure alignment. Let $\mathrm{ACF}(x, \ell)$ denote the autocorrelation of time series x at lag ℓ. The **ACF MAE** is defined as

$$\text{ACF-MAE} = \frac{1}{L} \sum_{\ell=1}^{L} \big|\mathbb{E}_{x\sim\mathcal{X}}[\mathrm{ACF}(x, \ell)] - \mathbb{E}_{\hat{x}\sim\hat{\mathcal{X}}}[\mathrm{ACF}(\hat{x}, \ell)]\big|, \tag{27}$$

where L is the maximum lag considered.

Let $\mathrm{PSD}(x)$ denote the power spectral density of x estimated using Welch's method. The **PSD ℓ_2 distance** is defined as

$$\text{PSD-}\ell_2 = \big\|\mathbb{E}_{x\sim\mathcal{X}}[\mathrm{PSD}(x)] - \mathbb{E}_{\hat{x}\sim\hat{\mathcal{X}}}[\mathrm{PSD}(\hat{x})]\big\|_2 . \tag{28}$$

Representativeness and prototype quality. For each real sample $x_i \in \mathcal{X}$, define its nearest synthetic neighbor as

$$d_i = \min_{\hat{x}_j \in \hat{\mathcal{X}}} \|x_i - \hat{x}_j\|_2. \tag{29}$$

The **Prototype Error (ProtoErr)** is reported as both the mean and median of $\{d_i\}_{i=1}^N$.

Let $m_{\mathcal{X}}$ and $m_{\hat{\mathcal{X}}}$ denote the medoids of the real and synthetic sets, respectively. The **Medoid Distance Ratio** is defined as

$$\mathrm{MDR} = \frac{\|m_{\mathcal{X}} - m_{\hat{\mathcal{X}}}\|_2}{\mathbb{E}_{x\sim\mathcal{X}}\big[\|x - m_{\mathcal{X}}\|_2\big]}. \tag{30}$$

Coverage of the real data manifold. Let τ_q denote the q-quantile of the real-to-real nearest-neighbor distance distribution. The **Coverage@τ_q** metric is defined as

$$\mathrm{Coverage@}\tau_q = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\Big(\min_{\hat{x}_j \in \hat{\mathcal{X}}} \|x_i - \hat{x}_j\|_2 \le \tau_q\Big), \tag{31}$$

where $\mathbf{1}(\cdot)$ is the indicator function. Higher coverage indicates better support overlap between synthetic and real data manifolds.