MARKOVTS: PRETRAINED STATE–TRANSITION REPRESENTATIONS FOR ZERO-SHOT TIME SERIES GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Across heterogeneous real-world domains, time-series generation supports simulation, model development, and data augmentation by generating sequences that preserve target-domain statistics and temporal dependencies when real observations are scarce or costly. However, existing methods predominantly address in-domain generation, limiting their reuse in unseen domains. We study reference-conditioned zero-shot time series generation: target domains are unseen during pretraining, generation uses a single target-domain reference sequence, and no model parameters are updated. MarkovTS characterizes sequence evolution through local morphologies and their ordered transitions at multiple temporal scales. Its pretrained encoder quantizes normalized time-series patches into multi-scale morphological states with a state codebook, while a transition codebook encodes contextual transitions between adjacent intervals. These transferable state–transition representations condition a pretrained flow-matching model. At generation, the frozen model combines a reference-derived condition with independent Gaussian noise to produce stochastic samples that retain the reference's temporal organization. Across 12 unseen target datasets, MarkovTS outperforms strong baselines in most distributional comparisons and generates diverse, reference-consistent sequences. As downstream augmentation, its samples yield overall gains across forecasting, classification, and anomaly detection benchmarks and diverse model families.

1 INTRODUCTION

Time-series generation estimates the distribution of observed sequences and generates new sequences with corresponding statistical properties and temporal dependencies. It is particularly valuable when representative measurements are scarce, costly to collect, or difficult to share across healthcare, energy, transportation, finance, and environmental monitoring (Yoon et al., 2019; Li et al., 2023; Gonen et al., 2025). The resulting generated time series support simulation, model development, and downstream data augmentation (Yuan & Qiao, 2024; Huang et al., 2025).

Existing generators typically acquire target-specific temporal structure from collections of target-domain sequences (Figure 1(a)). Dataset-specific models fit their parameters to these observations (Yoon et al., 2019; Desai et al., 2021; Yuan & Qiao, 2024), while pretrained generators use target examples for domain conditioning or few-shot adaptation (Huang et al., 2025; Gonen et al., 2025; Yao et al., 2026). Reusing pretrained parameters thus leaves the challenge of obtaining sufficient temporal information from a single reference in an unseen domain.

Real-world scenarios often lack sufficient data for model fitting and instead call for generating additional task-specific samples directly from a single reference. We term this zero-shot time series generation, in which the generator must capture and exploit transferable temporal structure within the reference without target-domain parameter updates.

Representation granularity determines which temporal information can be transferred from a single reference (Figure 1(b)). Global sequence codes summarize the entire series but offer limited flexibility across lengths and complexities (Fabius & van Amersfoort, 2014; Desai et al., 2021); point-level states preserve numerical detail but require costly contextual modeling (Yoon et al., 2019).

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.

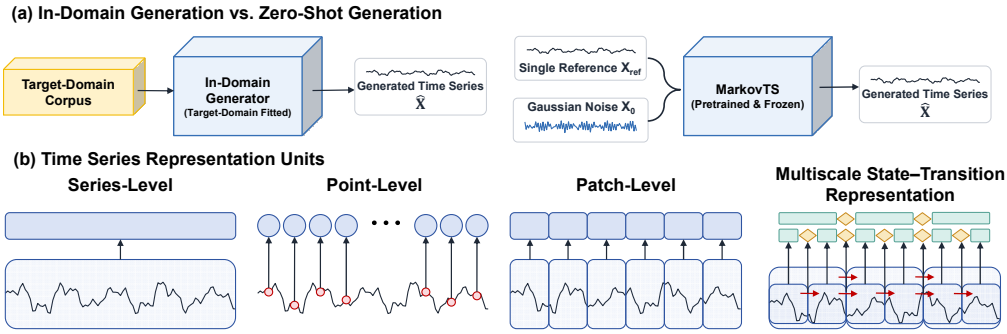


Figure 1: Generation settings and representation units. **(a)** In-domain training versus single-reference generation with frozen MarkovTS. **(b)** Series-, point-, and patch-level units versus the proposed multiscale state-transition representation.

Fixed-scale patches provide a local compromise, yet expose only one temporal scale (Zhou et al., 2026). Beyond granularity, tokenization and its learning objectives (Talukder et al., 2024; Wen et al., 2024; Tao et al., 2025) raise a further concern: downstream objectives may bias tokenizers toward local morphology while neglecting sequence evolution. A representation that captures local shapes but omits how they evolve provides an incomplete condition for reproducing the reference's temporal organization.

To address these limitations, we propose **MarkovTS**, combining a Markov-inspired multiscale state–transition encoder with a conditional flow-matching generator for zero-shot time series generation. The pretrained encoder uses a state codebook to quantize normalized multiscale patches into morphological states and a transition codebook to encode features between continuous patch-pairs and corresponding state-pairs. Reference-derived codebook vectors condition the pretrained generator to control temporal organization, while independent Gaussian sources provide stochastic variation. Together, local morphology and contextual transitions provide a reference-specific condition for controlling temporal organization during zero-shot transfer to unseen domains. Our contributions are as follows:

- We formalize single-reference zero-shot time series generation as transferring temporal organization from one unseen-domain reference without target-domain parameter updates.
- We propose MarkovTS, which couples a transferable multiscale state–transition representation of local morphology and contextual transitions with conditional flow matching to preserve stochastic variation across generated samples.
- Experiments across 12 unseen target datasets show strong distributional matching, while fixed-noise reference swaps show that generated outputs follow the selected joint condition. The generated data also support forecasting, classification, and anomaly detection.

2 Related Work and Background

2.1 Time-Series Generation

Time-series generators model temporal dependence through adversarial learning (Yoon et al., 2019; Xu et al., 2020; Jeon et al., 2022; Seyfi et al., 2022), continuous or discrete latent dynamics (Desai et al., 2021; Li et al., 2023; Zhou et al., 2023; Lee et al., 2023), and iterative denoising (Yuan & Qiao, 2024; Cao et al., 2024). TimeDP learns a multi-domain diffusion prior and derives a domain prompt from target examples without parameter updates (Huang et al., 2025); Winformer emphasizes window-level periodic alignment (Zhou et al., 2026). ImagenFew instead adapts pretrained diffusion models through target-domain few-shot fine-tuning (Gonen et al., 2025; Li et al., 2025).

Representation units also differ in temporal granularity. Sequence-level models use global latent codes (Fabius & van Amersfoort, 2014; Desai et al., 2021), whereas TimeGAN uses recurrent point-level states (Yoon et al., 2019). ImagenFew maps observations into invertible image representations (Gonen et al., 2025). Patch-level models aggregate local subseries and model inter-patch context;

Winformer aligns periodic windows (Zhou et al., 2026), while TimeVQVAE quantizes low- and high-frequency features into local time-frequency tokens with bidirectional priors (Lee et al., 2023).

Collapse toward morphology in time-series representations is subtle because in-domain tasks or training objectives place limited emphasis on complex evolution. VQShape and UniShape learn generalizable morphological representations for classification (Wen et al., 2024; Liu et al., 2026). Morphology-based forecasting improves performance (Yu et al., 2025; Lee et al., 2022; Shao et al., 2025; Wang et al., 2025) largely on smaller in-domain corpora, where other components can supply temporal context. Together, QUALS, ReCast, and LatentTSF highlight the risk that task-driven optimization favors a limited subset of temporal information, encouraging shortcut learning and representation degradation at the expense of transferable temporal structure (Li et al., 2026b; Ma et al., 2026; Yang et al., 2026).

2.2 Flow Matching

Flow matching trains a continuous normalizing flow by regressing a vector field that transports a simple source distribution to data distribution. Its simulation-free regression objective avoids integrating the flow during training and supports non-diffusion paths, including linear conditional paths that facilitate efficient sampling (Lipman et al., 2023). Let $X_0 \sim p_0 = \mathcal{N}(0, I)$ be a source sample, $X_1 \sim q_{\mathrm{data}}$ a real time series, $t \sim \mathcal{U}(0, 1)$. For the linear conditional path used in this work,

$$X_t = (1 - t)X_0 + tX_1, \qquad U_t = \frac{\mathrm{d}X_t}{\mathrm{d}t} = X_1 - X_0. \tag{1}$$

Given an optional condition C, a velocity network v_θ is optimized by

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{X_0, X_1, t}\left[\|v_\theta(X_t, t; C) - U_t\|_2^2\right]. \tag{2}$$

After training, generation starts from $\widetilde{X}_0 \sim p_0$ and solves

$$\frac{\mathrm{d}\widetilde{X}_t}{\mathrm{d}t} = v_\theta(\widetilde{X}_t, t; C), \qquad \widehat{X} = \widetilde{X}_1. \tag{3}$$

In MarkovTS, C is instantiated by the multiscale state and transition codes extracted from a reference time series; Section 4 specifies how these codes are learned and injected into the velocity network.

3 Problem Formulation

Let $\mathcal{D}_{\mathrm{pre}} = \bigcup_{i=1}^{M} \mathcal{D}_i$ be a pretraining corpus drawn from source-domain distributions $\mathcal{Q}_{\mathrm{pre}} = \{q_i\}_{i=1}^{M}$. Given a reference sequence $X_{\mathrm{ref}} \sim q_*$ from an unseen target distribution $q_* \notin \mathcal{Q}_{\mathrm{pre}}$, zero-shot time series generation uses fixed pretrained parameters Θ and independent Gaussian sources to generate new sequences:

$$Z_j \sim p_0 = \mathcal{N}(0, I), \qquad \widehat{X}^{(j)} = G_\Theta\big(Z_j; X_{\mathrm{ref}}\big), \quad j = 1, \ldots, N. \tag{4}$$

The resulting conditional distribution $p_\Theta(\widehat{X} \mid X_{\mathrm{ref}})$ should preserve the statistical properties and temporal organization expressed by X_{ref}, while Z_j provides variation among generated sequences.

4 Method

4.1 Overview

MarkovTS combines a pretrained multiscale state–transition encoder with a conditional flow-matching generator (Figure 2). At inference, the encoder extracts a fixed condition from a single reference, and the generator produces multiple sequences from independent Gaussian sources.

4.2 Multiscale State–Transition Representation Pretraining

Inspired by hidden Markov models, MarkovTS uses a state codebook to describe local morphology and a transition codebook to quantize transition queries derived from relations between adjacent

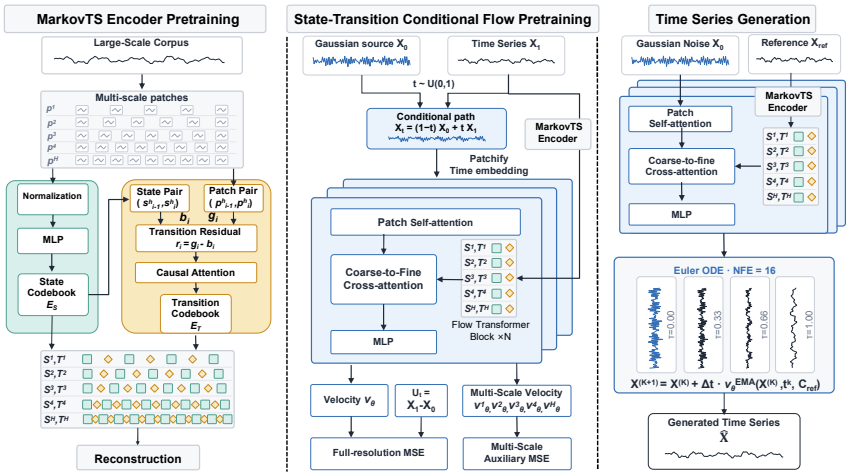


Figure 2: Overview of MarkovTS. **Left:** multiscale state–transition representation learning. **Middle:** conditional flow pretraining. **Right:** single-reference zero-shot generation.

patches and contextualized by causal attention over the full preceding context. This construction does not assume a first-order transition matrix.

MarkovTS processes each variable independently with shared parameters. For a length-L time series with D aligned variables, $X \in \mathbb{R}^{D\times L}$, standardization gives $\bar{X}$. At scale h, with $w_1 > \cdots > w_H$, each variable sequence (row) is partitioned into n_h non-overlapping intervals of span w_h, each represented by a P-dimensional patch p_i^h, $i = 0, \ldots, n_h - 1$. We omit the variable index d below.

The state branch isolates local morphology before assigning a discrete code. Let $\mu_{\mathrm{v}}(p_i^h)$ and $\sigma_{\mathrm{v}}(p_i^h)$ denote the mean and population standard deviation over nonempty bins only. It uses $\widetilde{p}_i^h = (p_i^h - \mu_{\mathrm{v}}(p_i^h))/\max(\sigma_{\mathrm{v}}(p_i^h), \epsilon)$; empty bins are filled with $\mu_{\mathrm{v}}(p_i^h)$ and therefore map to zero. A shared multilayer perceptron maps $\widetilde{p}_i^h$ to a unit-normalized query $q_i^{S,h}$, which is assigned to the most similar entry in the state codebook $E_S = \{e_k^S\}_{k=1}^{N_S}$:

$$k_i^{S,h} = \arg\max_k \cos(q_i^{S,h}, e_k^S), \qquad s_i^h = e^S_{k_i^{S,h}}. \tag{5}$$

Patch-independent encoding and a scale-shared codebook provide a common vocabulary for local morphologies at different temporal extents.

The transition branch complements normalized states with changes between adjacent intervals, indexed by $i = 1, \ldots, n_h - 1$. A separate encoder maps patches without local standardization to continuous features $a_i^h = f_{\mathrm{full}}(p_i^h)$, retaining relative level and amplitude. The patch-pair and state-pair encoders concatenate adjacent features and their differences to form the continuous patch-pair relation g_i^h, state-pair relation b_i^h, and transition residual r_i^h:

$$\begin{aligned} g_i^h &= f_P([a_{i-1}^h, a_i^h, a_i^h - a_{i-1}^h]), \\ b_i^h &= f_S([s_{i-1}^h, s_i^h, s_i^h - s_{i-1}^h]), \qquad r_i^h = g_i^h - b_i^h. \end{aligned} \tag{6}$$

Projected residual queries are interleaved with state vectors at interval boundaries and centers, respectively. Causal attention uses preceding states and residual queries to produce normalized transition queries $q_i^{T,h}$. Quantization selects from the separate, scale-shared transition codebook $E_T = \{e_k^T\}_{k=1}^{N_T}$:

$$k_i^{T,h} = \arg\max_k \cos(q_i^{T,h}, e_k^T), \qquad \tau_i^h = e^T_{k_i^{T,h}}. \tag{7}$$

Thus, identical local state pairs can receive different transition codes when their continuous relations or preceding contexts differ.

Pretraining first learns morphological states through normalized-patch reconstruction and code commitment (van den Oord et al., 2017). The state tokenizer and codebook are then frozen. State–transition learning reconstructs full-resolution and multiscale waveforms, while residual matching trains transition codes to retain relational direction and magnitude. Auxiliary regularization stabilizes the queries and discourages collapse.

For variable d, let $S_d^h = (s_{d,0}^h, \ldots, s_{d,n_h-1}^h)$ and $T_d^h = (\tau_{d,1}^h, \ldots, \tau_{d,n_h-1}^h)$ denote ordered sequences of assigned state and transition vectors. The complete representation supplied to the generator is

$$C(\bar{X}) = \big(C_d(\bar{X})\big)_{d=1}^{D}, \qquad C_d(\bar{X}) = \big((S_d^h, T_d^h)\big)_{h=1}^{H}. \tag{8}$$

4.3 State–Transition Conditional Flow Pretraining

The conditional flow model converts discrete state–transition representations into continuous waveforms by injecting temporally aligned conditions from coarse to fine. The encoder and codebooks remain fixed, while flow matching trains the generator from Gaussian sources.

Let X_1 denote a standardized training input and $X_{1,d} \in \mathbb{R}^L$ its variable-d sequence, paired with $C_d(X_1)$. An independent Gaussian source $X_{0,d}$ and time t define $X_{t,d}$ through Equation 1. Each forward pass uses one variable and its condition. Separate projections map the interleaved state and transition vectors into a common conditioning space while preserving their temporal coordinates. Each fine-scale element incorporates the average of overlapping projected elements from the nearest available coarser scale, and within-scale self-attention produces the scale-h condition memory C_d^h.

The flow Transformer embeds patches of $X_{t,d}$ with flow-time information. Each block applies waveform self-attention, then injects conditions through coarse-to-fine cross-attention. At each scale, waveform features are pooled over aligned intervals and attend to condition memory within overlapping temporal spans. The update is interpolated back and added before the next finer scale, allowing fine conditions to refine the coarse-conditioned waveform features. For waveform features H_ℓ in block ℓ, the scale-h update is

$$H_\ell \leftarrow H_\ell + \mathrm{Up}_h\big[\mathrm{CA}\big(\mathrm{Pool}_h(H_\ell), C_d^h\big)\big]. \tag{9}$$

The operators Pool_h and Up_h denote aligned pooling and interpolation, and CA denotes cross-attention. Pooled queries are recomputed after each scale update, so finer-scale conditioning builds on the preceding coarser-scale update. A feed-forward layer completes the block.

The output predicts the full-resolution velocity $v_\theta(X_{t,d}, t, C_d(X_1)) \in \mathbb{R}^L$ against $X_{1,d} - X_{0,d}$ under conditional flow matching (Lipman et al., 2023). Auxiliary heads predict pooled versions of the same target, linking multiscale conditions to the full-resolution field used for sampling.

4.4 Reference-Conditioned Zero-Shot Generation

At inference, the pretrained generator produces new sequences from a fixed reference-derived condition and fresh Gaussian sources. Each aligned variable uses its own condition; sources are shared across variables within a sample and independent across samples.

Reference statistics normalize X_{ref} to $\bar{X}_{\mathrm{ref}}$, which is encoded once into $C(\bar{X}_{\mathrm{ref}})$. For sample j, $\epsilon^{(j)} \sim \mathcal{N}(0, I_L)$ initializes each variable. Euler integration on $0 = t_0 < \cdots < t_K = 1$ uses the exponential-moving-average velocity field $v_{\bar{\theta}}$:

$$\begin{aligned} Z_{d,0}^{(j)} &= \epsilon^{(j)}, \\ Z_{d,k+1}^{(j)} &= Z_{d,k}^{(j)} + (t_{k+1} - t_k)\, v_{\bar{\theta}}\Big(Z_{d,k}^{(j)}, t_k, C_d(\bar{X}_{\mathrm{ref}})\Big). \end{aligned} \tag{10}$$

The reference mean μ_d and effective scale s_d restore the original units as $\widehat{X}_d^{(j)} = \mu_d + s_d Z_{d,K}^{(j)}$. Implementation details, training objectives, and theories are provided in Appendices B–C.

5 Experiments

We organize the evaluation around three research questions:

Table 1: Time-series generation quality on non-overlapping length-168 windows from length-512 outputs conditioned on length-512 references. Lower is better; values are five-seed mean ± sample standard deviation. Best and second-best methods are **bold** and underlined.

Metric	Dataset	MarkovTS	Winformer	ImagenFew	TimeDP	Diffusion-TS	TimeGAN	GT-GAN	TimeVAE	TimeVQVAE	TimeVQVAE-C
Maximum Mean Discrepancy	Electricity	**0.001**±0.001	**0.001**±0.002	0.002±0.003	**0.001**±0.001	0.003±0.002	0.367±0.255	0.254±0.166	0.577±0.006	0.152±0.024	0.002±0.001
	Solar	**0.033**±0.003	0.035±0.004	0.039±0.011	0.041±0.011	0.050±0.012	0.628±0.053	0.578±0.039	0.353±0.014	0.437±0.020	0.058±0.005
	Wind	**0.013**±0.009	0.034±0.014	0.071±0.050	0.025±0.017	0.035±0.006	0.213±0.017	0.170±0.040	0.170±0.004	0.131±0.014	0.018±0.007
	Traffic	0.087±0.002	**0.071**±0.005	0.080±0.009	0.083±0.034	0.111±0.031	0.567±0.057	0.538±0.078	0.218±0.007	0.213±0.016	0.089±0.001
	Taxi	0.110±0.008	**0.085**±0.010	0.108±0.042	0.095±0.023	0.131±0.014	0.275±0.054	0.319±0.032	0.139±0.007	0.128±0.004	0.109±0.014
	Pedestrian	**0.032**±0.006	0.040±0.008	0.041±0.012	0.044±0.020	0.071±0.019	0.090±0.030	0.112±0.019	0.065±0.002	0.067±0.007	0.058±0.002
	Air	0.015±0.002	**0.011**±0.002	0.023±0.013	**0.011**±0.003	0.022±0.011	0.120±0.045	0.211±0.041	0.089±0.016	0.028±0.002	0.041±0.008
	Temperature	**0.211**±0.002	0.230±0.021	0.275±0.059	0.219±0.022	0.241±0.049	0.926±0.042	0.809±0.081	1.002±0.014	0.323±0.008	0.259±0.043
	Rain	**0.031**±0.002	0.036±0.016	0.052±0.003	0.057±0.039	0.079±0.058	0.329±0.285	0.111±0.109	0.292±0.019	0.074±0.007	0.080±0.004
	NN5	**0.145**±0.000	0.147±0.008	0.164±0.005	0.164±0.010	0.186±0.043	0.874±0.088	0.632±0.074	0.821±0.061	0.327±0.012	0.243±0.041
	Fred-MD	**0.001**±0.000	0.002±0.001	0.005±0.002	0.002±0.001	0.006±0.002	0.043±0.021	0.133±0.102	0.059±0.008	0.008±0.002	0.005±0.002
	Exchange	**0.124**±0.008	0.137±0.012	0.160±0.027	0.151±0.024	0.206±0.035	0.530±0.154	0.475±0.116	0.543±0.149	0.342±0.050	0.233±0.107
K-L Divergence	Electricity	**0.001**±0.000	0.008±0.010	0.154±0.094	0.012±0.016	0.315±0.247	0.488±0.175	0.407±0.079	0.734±0.023	0.280±0.051	0.027±0.015
	Solar	**0.009**±0.002	0.013±0.005	0.079±0.010	0.016±0.005	0.066±0.055	0.612±0.447	0.120±0.041	0.260±0.016	0.865±0.108	0.234±0.062
	Wind	0.111±0.018	0.202±0.044	0.374±0.295	0.152±0.034	0.548±0.372	1.924±1.233	**0.107**±0.016	0.484±0.015	0.483±0.066	0.183±0.047
	Traffic	**0.007**±0.002	0.011±0.002	0.010±0.004	0.009±0.003	0.120±0.074	1.305±0.320	1.409±0.251	0.211±0.014	0.178±0.026	0.016±0.003
	Taxi	0.010±0.004	**0.005**±0.003	0.020±0.011	0.011±0.004	0.075±0.034	0.650±0.180	0.950±0.197	0.110±0.020	0.110±0.026	0.038±0.010
	Pedestrian	**0.007**±0.001	0.009±0.004	0.020±0.008	0.014±0.010	0.133±0.069	0.417±0.181	0.411±0.096	0.065±0.005	0.405±0.051	0.039±0.008
	Air	**0.026**±0.011	**0.026**±0.011	0.054±0.026	0.027±0.016	0.106±0.079	0.348±0.093	0.578±0.049	0.164±0.012	0.054±0.012	0.093±0.025
	Temperature	**0.137**±0.041	0.176±0.027	0.235±0.070	0.171±0.073	0.342±0.131	8.892±2.681	3.174±2.685	2.183±0.110	0.735±0.066	0.379±0.110
	Rain	**0.002**±0.001	0.011±0.003	0.012±0.002	0.013±0.012	0.061±0.072	0.506±0.174	0.432±0.099	0.160±0.022	0.047±0.018	0.065±0.018
	NN5	**0.045**±0.003	**0.045**±0.007	0.057±0.017	0.054±0.014	0.165±0.076	4.928±4.112	1.386±0.520	1.337±0.220	1.063±0.274	0.220±0.151
	Fred-MD	**0.176**±0.001	0.201±0.014	0.528±0.445	0.203±0.035	0.835±0.554	0.512±0.290	0.380±0.070	0.346±0.041	0.831±0.077	1.118±0.276
	Exchange	1.794±0.137	**1.621**±0.126	3.498±3.061	1.866±0.132	2.337±0.714	8.861±3.397	7.201±4.380	10.404±1.434	5.052±1.385	8.475±3.056
Wins / win rate		18 / 75.0%	8 / 33.3%	0 / 0.0%	2 / 8.3%	0 / 0.0%	0 / 0.0%	1 / 4.2%	0 / 0.0%	0 / 0.0%	0 / 0.0%

RQ1: Generation quality. How well does MarkovTS match unseen target distributions while preserving reference-specific temporal organization and stochastic diversity?

RQ2: Representation analysis. What do state and transition codes represent, and how do representation design and code assignment affect generation quality?

RQ3: Downstream utility. Can MarkovTS-generated data improve downstream tasks?

5.1 Experimental Setup

Data and generation settings. We evaluate generation on 12 datasets covering energy, transportation, environmental, and economic data. The MarkovTS encoder and generator are pretrained on GIFT-Eval Pretrain (Aksu et al., 2024),[1] excluding the 12 evaluation datasets. We follow the evaluation settings of TimeDP and Winformer, using the target training split only to provide references, without fine-tuning. In the main comparison, deterministic nearest-code state and transition assignments from each length-512 reference condition length-512 outputs generated with independent Gaussian sources and 16-step Euler integration. Each output is partitioned into three non-overlapping length-168 evaluation windows, with the final eight points discarded. Results report means and sample standard deviations over seeds 2021–2025.

Baselines. We compare with nine generators: TimeGAN and GT-GAN (Yoon et al., 2019; Jeon et al., 2022); TimeVAE, TimeVQVAE, and TimeVQVAE-C (Desai et al., 2021; Lee et al., 2023; Huang et al., 2025); and Diffusion-TS, TimeDP, Winformer, and ImagenFew (Yuan & Qiao, 2024; Huang et al., 2025; Zhou et al., 2026; Gonen et al., 2025). All main-table baselines are trained or adapted with target-domain training data under their reported protocols; notably, TimeDP and its reproduced baselines are trained jointly on all 12 evaluation datasets, with unseen-domain generation evaluated only in a separate experiment (Huang et al., 2025).

Evaluation metrics. We use Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) and a histogram-based flattened Kullback–Leibler divergence (flat-KL) (Kullback & Leibler, 1951) for marginal target-distribution agreement; lower values are better. Fixed-noise reference swaps assess conditional fidelity using shape-normalized TS2Vec, autocorrelation, spectral distance, and reference retrieval.

Appendices B.5 and D.1–D.3 provide training configurations and evaluation details covering datasets, protocols, metrics, and baselines.

[1] https://huggingface.co/datasets/Salesforce/GiftEvalPretrain

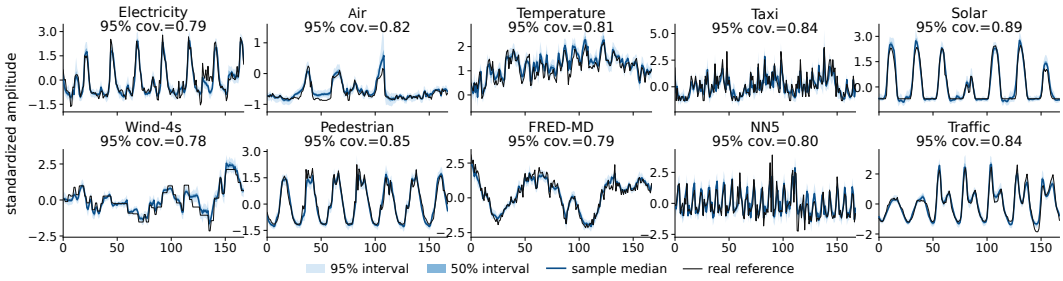


Figure 3: Conditional distributions across ten held-out datasets. Panels show references, medians, and 50%/95% empirical intervals; inclusion rate denotes reference coverage by the 95% interval.

5.2 Generation Quality (RQ1)

Distributional fidelity. MarkovTS is best or tied-best in 18 of 24 MMD and flat-KL comparisons in Table 1, leading on 9 of 12 datasets per metric. Representative results are MMD 0.013 versus 0.018 on Wind and flat-KL 0.002 versus 0.011 on Rain.

MMD and flat-KL support agreement with target length-168 window and pooled-value distributions. The same pretrained model serves 12 datasets without target updates; Appendix E.1 gives complementary controls.

Reference-specific conditional fidelity. MarkovTS aims to generate diverse samples faithful to a selected reference, which marginal MMD and flat-KL cannot test. Table 2 fixes Gaussian noise and reference i's affine restoration, then replaces only the joint state–transition condition with that of a deranged reference $j \neq i$; a condition-agnostic generator would be unaffected. On every dataset, swaps increase distance from i in shape-normalized TS2Vec, ACF, and PSD. Active-condition retrieval reaches 93.7% in the binary task and about 63% in the 100-way task, versus chance levels of 50% and 1%; nearly equal correct/swapped 100-way accuracy indicates redirection rather than information loss. Thus, the intervention directly supports the reference-specific use of the joint condition required by our single-reference claim, without isolating states from transitions.

Table 2: Reference-specific fidelity under fixed-noise condition swaps. Entries are five-seed means; Appendix E.1.1 reports standard deviations.

Metric	Elec.	Solar	Wind	Traffic	Taxi	Ped.	Air	Temp.	Rain	NN5	Fred-MD	Exchange	Mean
Δ Shape	0.016	0.017	0.093	0.056	0.041	0.127	0.074	0.036	0.040	0.018	0.032	0.057	0.051
Δ ACF	0.230	0.085	0.234	0.124	0.152	0.177	0.237	0.078	0.056	0.093	0.233	0.242	0.162
Δ PSD	0.250	0.122	0.242	0.160	0.214	0.218	0.300	0.148	0.220	0.113	0.171	0.236	0.199
Binary retrieval	95.5	95.7	93.8	98.8	99.4	92.0	95.8	92.3	96.7	75.9	92.2	96.0	93.7
100-way correct	58.9	51.6	71.2	84.1	89.1	45.8	86.6	45.1	78.3	26.0	46.3	77.3	63.4
100-way swapped	59.2	51.9	69.9	84.2	89.4	46.7	84.0	45.0	78.0	25.6	46.5	76.6	63.1

Reference fidelity and stochastic diversity. Figure 3 summarizes 50 generated samples per reference across ten held-out datasets. Recurring median traces and nonzero 50%/95% intervals show reference alignment with stochastic variation; Appendix F gives details and samples.

5.3 State–Transition Analysis (RQ2)

Code and model replacement. Table 3 compares deterministic assignment with uniform code replacement (Random) and reference-dependent probability sampling (Stochastic), applied to states, transitions, or both with model parameters frozen. This paired intervention uses a deterministic control distinct from the three-window main estimate in Table 1. Randomizing both streams causes greater degradation than either single-stream intervention in most comparisons, indicating that both provide useful reference information. Stochastic sampling instead maintains scores close to the deterministic control, suggesting that reference-dependent code selection preserves useful conditioning information despite assignment changes.

Table 3: Generation quality under state–transition code interventions and patch-based comparison.

Variant	Metric	Elec.	Solar	Wind	Traffic	Taxi	Ped.	Air	Temp.	Rain	NN5	Fred-MD	Exchange
MarkovTS	MMD	0.001	0.032	0.019	0.087	0.101	0.032	0.009	0.142	0.031	0.180	0.002	0.124
	flat-KL	0.001	0.009	0.147	0.003	0.002	0.007	0.019	0.161	0.007	0.043	0.244	1.631
Random states	MMD	0.002	0.236	0.023	0.168	0.118	0.049	0.016	0.134	0.046	0.205	0.003	0.148
	flat-KL	0.059	0.689	0.189	0.144	0.075	0.088	0.039	0.134	0.010	0.106	0.364	5.030
Random transitions	MMD	0.002	0.190	0.023	0.130	0.119	0.054	0.017	0.152	0.053	0.231	0.002	0.147
	flat-KL	0.061	0.842	0.165	0.131	0.146	0.148	0.046	0.109	0.015	0.193	0.200	3.068
Random both	MMD	0.003	0.516	0.023	0.302	0.148	0.082	0.022	0.213	0.051	0.292	0.002	0.148
	flat-KL	0.068	1.099	0.178	0.247	0.184	0.192	0.057	0.253	0.050	0.301	0.236	3.351
Stochastic states	MMD	0.001	0.034	0.023	0.080	0.091	0.034	0.013	0.134	0.036	0.166	0.003	0.138
	flat-KL	0.005	0.015	0.177	0.007	0.016	0.010	0.030	0.098	0.004	0.047	0.256	1.760
Stochastic transitions	MMD	0.002	0.034	0.023	0.080	0.091	0.035	0.013	0.134	0.036	0.186	0.003	0.142
	flat-KL	0.012	0.014	0.177	0.007	0.017	0.015	0.030	0.098	0.004	0.051	0.291	1.842
Stochastic both	MMD	0.002	0.034	0.023	0.080	0.091	0.037	0.013	0.134	0.036	0.186	0.003	0.143
	flat-KL	0.015	0.020	0.177	0.007	0.016	0.020	0.030	0.098	0.004	0.051	0.288	1.883
Patch-Flow	MMD	0.003	0.045	0.033	0.084	0.105	0.057	0.011	0.204	0.040	0.236	0.004	0.142
	flat-KL	0.054	0.173	0.203	0.003	0.012	0.163	0.021	0.224	0.008	0.215	0.369	2.013

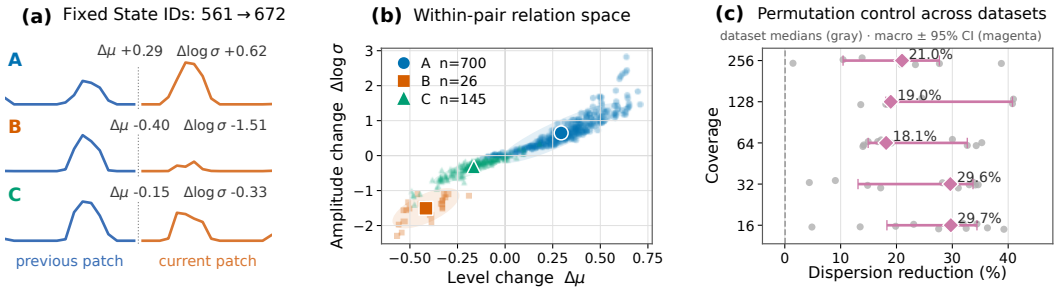


Figure 4: State–transition complementarity on Solar at span 32. (a) Real patch pairs sharing state IDs but assigned distinct transition codes. (b) Occurrence centroids and covariance ellipses. (c) Scale-level within-code dispersion reduction against count-preserving permutations across datasets; gray points are dataset medians, and magenta diamonds are macro-medians with 95% bootstrap CIs.

Patch-Flow replaces the state–transition representation with a standard patch Transformer and single codebook while retaining the remaining model and training settings. Under the paired intervention protocol, MarkovTS has lower MMD and flat-KL on eleven of twelve datasets, supporting state–transition over single-codebook patch conditioning.

State–transition complementarity. Using real training references and the frozen encoder, we test whether transition codes distinguish changes within the same ordered state pair. Figure 4(a–b) illustrates distinct level and amplitude changes associated with different transition codes. Across 3,589 groups defined by dataset, scale, and state pair, we compare within-code dispersion in level, amplitude, slope, and boundary changes against count-preserving label permutations. The scale-level macro-median reduction is 18.1–29.7%, with all five 95% bootstrap confidence intervals excluding zero (Figure 4(c)). This indicates that context-conditioned transition codes capture variation not resolved by the state pair, complementing the morphological states.

5.4 Downstream Utility (RQ3)

We use MarkovTS-generated data for downstream training. *Real + MarkovTS* pairs each real training example with one generated counterpart and replaces exactly half of every mini-batch: complete sequences for classification and position-matched windows for forecasting and anomaly detection. Batch size and updates per epoch match *Real only*; Arsenal uses a fixed 50% replacement set of equal cardinality. All comparisons share maximum training budgets, model selection, real validation indices, and real test data.

Forecasting. Following the BasicTS protocol (Shao et al., 2024; Li et al., 2026a)[2], we evaluate five backbones on six datasets with lookback 336 and prediction horizons $\{96, 192, 336, 720\}$. Table 4 reports mean absolute error (MAE) and mean squared error (MSE), averaged over the four horizons.

[2] https://github.com/GestaltCogTeam/BasicTS

Table 4: Forecasting performance averaged over four horizons. Cells show *Real only*/*Real + MarkovTS*; improvements are bold. Imp. counts improved dataset–metric pairs out of 12.

	ECL		ETTh1		ETTh2		Exchange		Traffic		Weather		
Backbone	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	Imp.
CATS	.258/**.245**	.165/**.161**	.443/**.423**	.448/**.425**	.425/**.384**	.392/**.355**	.420/.421	.388/**.387**	.284/**.252**	.421/**.402**	.258/**.249**	.221/.225	10/12
iTrans.	.259/.261	.164/.169	.473/**.433**	.485/**.421**	.439/**.427**	.421/**.409**	.478/**.447**	.464/**.419**	.289/.297	.412/.429	.267/**.258**	.233/**.230**	8/12
SparseTSF	.265/**.261**	.173/**.171**	.429/**.417**	.427/**.412**	.427/**.414**	.399/**.389**	.481/**.425**	.478/**.390**	.295/**.279**	.445/.458	.283/**.276**	.252/.255	10/12
TQNet	.246/.247	.158/.158	.425/.428	.422/.429	.417/**.416**	.391/.393	.454/**.427**	.428/**.387**	.265/**.259**	.410/**.398**	.249/.252	.222/.223	5/12
OLinear	.298/**.297**	.188/**.186**	.428/.429	.414/.415	.416/**.412**	.389/**.386**	.456/**.439**	.435/**.402**	.328/**.321**	.465/**.453**	.250/.254	.221/.223	8/12

Table 5: Classification on 23 UEA datasets: *Real only*/*Real + MarkovTS*. Bold: improvements; Imp.: improved metrics. Percentages except LL.

Classifier	Acc.↑	BAcc.↑	F1↑	AUROC↑	LL↓	Sens.↑	Spec.↑	Imp.
ResNet1D	68.15/**68.94**	67.94/**68.86**	65.05/**67.12**	83.71/**85.05**	1.13/**1.07**	66.02/**69.33**	69.95/69.21	6/7
TimesNet	65.84/**68.36**	65.03/**66.94**	63.86/**65.26**	83.32/**84.13**	1.16/**1.10**	66.28/**66.73**	65.03/**68.30**	7/7
ConvTran	69.31/**70.23**	68.01/**69.16**	67.93/67.49	85.52/**85.65**	0.82/**0.79**	70.49/69.10	67.36/**70.65**	5/7
Arsenal	73.21/73.17	71.34/71.32	71.16/**71.31**	84.39/**84.85**	5.56/**5.12**	71.19/**71.53**	74.17/73.77	4/7

Table 6: Anomaly-detection on 26 datasets. Cells show *Real only*/*Real + MarkovTS*; improvements are bold. Higher is better; Imp. counts improved metrics. A-P/A-R: pointwise PR-/ROC-AUC; R-A-P/R-A-R: range-aware AUCs; V-PR/V-ROC: PR/ROC volumes under surface.

Detector	A-P↑	A-R↑	R-A-P↑	R-A-R↑	V-PR↑	V-ROC↑	Imp.
DLinear	.233/**.253**	.672/**.679**	.255/**.273**	.713/**.715**	.254/**.271**	.710/**.714**	6/6
TranAD	.278/.278	.674/**.676**	.267/.261	.676/**.677**	.264/.259	.673/**.675**	3/6
CATCH	.248/**.254**	.675/**.683**	.278/**.284**	.720/**.726**	.275/**.281**	.717/**.723**	6/6
ModernTCN	.237/**.241**	.660/**.664**	.259/**.262**	.704/**.708**	.257/**.261**	.702/**.705**	6/6
TimesNet	.268/.267	.689/.685	.286/**.290**	.731/.730	.285/**.289**	.729/.727	2/6

Augmentation reduces both errors for every backbone on multiple datasets; ETTh1 MSE drops from 0.485 to 0.421 for iTransformer and Exchange MSE from 0.478 to 0.390 for SparseTSF.

For forecasting only, *Online* regenerates one counterpart per real training window each epoch, trains only on generated windows, and matches the per-epoch batch and update counts of *Real only* (Appendix G).

Classification. On 23 UEA datasets (Bagnall et al., 2018) from Multiverse (Middlehurst et al., 2026)[3], four classifiers are evaluated using seven metrics; generated samples inherit their references' class labels. *Real + MarkovTS* improves most aggregates in Table 5, including all seven for TimesNet; improvements average 2.69%, with no decline exceeding 2%.

Anomaly detection. Following TAB/OpenTS (Qiu et al., 2025)[4], five detectors are evaluated on 215 series from 26 datasets; generated counterparts match their references' lengths and index grids and inherit their pointwise labels. *Real + MarkovTS* improves all six aggregates for DLinear, CATCH, and ModernTCN; improvements average 1.89%, with mixed results for TranAD and TimesNet.

Appendix D provides protocols; Appendices G–I report complete results.

6 Conclusion

We introduced MarkovTS for reference-conditioned zero-shot time series generation. Its pretrained multiscale state–transition encoder separates locally normalized morphology from contextual transitions to condition a flow-matching generator for stochastic generation from one reference without target-domain tuning. Across 12 unseen datasets, MarkovTS is best or tied-best in most MMD and flat-KL comparisons. Fixed-noise swaps show outputs follow the selected joint condition; code interventions and within-state-pair analyses support complementary state and transition components. MarkovTS-generated data improve forecasting, classification, and anomaly detection on average under matched training budgets, with gains varying across models and metrics. Together, the results support reference-specific generation beyond marginal distribution matching.

[3] https://github.com/aeon-toolkit/multiverse

[4] https://decisionintelligence.github.io/OpenTS/leaderboards/

AI USE STATEMENT

In this work, we use generative AI tools, including GPT-5.6 Sol and GPT-6 Astra, in part as rigorous reviewers, to assist with literature review and consolidation, suggestions for experimental ablations and analyses, performance aggregation, Python visualization, and checks of experimental consistency. These tools also assist with translation, grammar correction, language polishing, and consistency checking throughout the manuscript. All AI-assisted content, code, analyses, and claims are reviewed and verified by the authors, who take full responsibility for the final work.

ETHICS STATEMENT

This computational study uses previously released, publicly available research datasets and benchmark implementations and entails no new data collection or participant recruitment. Pretraining uses the GIFT-Eval Pretrain corpus. The generation datasets and experimental protocol follow the TimeDP repository, and downstream evaluation uses BasicTS for forecasting, Multiverse for classification, and TAB/OpenTS for anomaly detection. Dataset provenance and preprocessing are documented in Appendix D. We identify no study-specific concerns under the ICLR Code of Ethics beyond responsible use and attribution of these public resources.

REPRODUCIBILITY STATEMENT

Sections 3–5 provide the reproducible core of the study: Section 3 formalizes the single-reference zero-shot setting and its data-access assumptions; Section 4 specifies representation pretraining, conditional flow learning, and inference; and Section 5 defines the evaluation and reports the principal evidence for generation quality, reference-specific conditioning, representation behavior, and downstream utility. Appendix A supports the formulation with unified notation and tensor dimensions. Appendices B–C expand the method: Appendix B specifies preprocessing, patchification, architecture, objectives, codebook updates, optimization, sampling, implementation settings, and throughput, while Appendix C establishes the stated invariance, stability, and transition-residual properties. Appendices D–K support the evaluation. Appendix D records data provenance, splits, baselines, metrics, aggregation, random seeds, and task-specific training settings; Appendix E reports additional generation results and controlled reference, representation, and length analyses; Appendix F examines code structure, state–transition complementarity, multiscale conditioning, distributional diagnostics, flow trajectories, and sample stability; Appendices G–I provide horizon-wise forecasting and per-dataset classification and anomaly-detection results; Appendix J visualizes partial code randomization; and Appendix K states the study's limitations and scope. All external datasets and benchmark implementations named above are publicly available.

An anonymized code package containing the necessary core implementation is included with the submission. It provides the multiscale state–transition representation model, codebook learning and lookup, the conditional flow-matching generator, reference-conditioned sampling, and interfaces for forecasting, classification, and anomaly-detection benchmarks.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024. URL https://arxiv.org/abs/2410.10393.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018. URL https://arxiv.org/abs/1811.00075.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. URL https://arxiv.org/abs/2105.04906.

Defu Cao, Wen Ye, Yizhou Zhang, and Yan Liu. TimeDiT: General-purpose diffusion transformers for time series foundation model. *arXiv preprint arXiv:2409.02322*, 2024. URL https://arxiv.org/abs/2409.02322.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. TimeVAE: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021. URL https://arxiv.org/abs/2111.08095.

Otto Fabius and Joost R. van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014. URL https://arxiv.org/abs/1412.6581.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I. Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, 38:22–48, 2024. doi: 10.1007/s10618-023-00948-2. URL https://doi.org/10.1007/s10618-023-00948-2.

Tal Gonen, Itai Pemper, Ilan Naiman, Nimrod Berman, and Omri Azencot. Time series generation under data scarcity: A unified generative modeling approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5750. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fc1767dae81344d5b9290abfdceed417-Abstract-Conference.html.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://www.jmlr.org/papers/v13/gretton12a.html.

Yu-Hao Huang, Chang Xu, Yueying Wu, Wu-Jun Li, and Jiang Bian. TimeDP: Learning to generate multi-domain time series with domain prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17520–17527, 2025. doi: 10.1609/aaai.v39i17.33926. URL https://ojs.aaai.org/index.php/AAAI/article/view/33926.

Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. GT-GAN: General purpose time series synthesis with generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36999–37010, 2022. doi: 10.52202/068431-2681. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/f03ce573aa8bce26f77b76f1cb9ee979-Abstract-Conference.html.

Dongbin Kim, Jinseong Park, Jaewook Lee, and Hoki Kim. Are self-attentions effective for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/cf66f995883298c4db2f0dcba28fb211-Abstract-Conference.html.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. doi: 10.1214/aoms/1177729694.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7665–7693. PMLR, 2023. URL https://proceedings.mlr.press/v206/lee23d.html.

Hyunwook Lee, Seungmin Jin, Hyeshin Chu, Hongkyu Lim, and Sungahn Ko. Learning to remember patterns: Pattern matching memory networks for traffic forecasting. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.10380.

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. doi: 10.1609/aaai.v37i7.26031. URL https://ojs.aaai.org/index.php/AAAI/article/view/26031.

Yujie Li, Zezhi Shao, Chengqing Yu, Tangwen Qian, Zhao Zhang, Yifan Du, Shaoming He, Fei Wang, and Yongjun Xu. STA-GANN: A valid and generalizable spatio-temporal kriging approach. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1726–1736, 2025.

Yujie Li, Zezhi Shao, Chengqing Yu, Yisong Fu, Tao Sun, Yongjun Xu, and Fei Wang. APT: Affine prototype-timestamp for time series forecasting under distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15180–15188, 2026a. doi: 10.1609/aaai.v40i18.38542. URL https://doi.org/10.1609/aaai.v40i18.38542.

Yujie Li, Zezhi Shao, Chengqing Yu, Yisong Fu, Weijie Zhu, Yifan Du, Jilin Hu, Bin Yang, Yongjun Xu, and Fei Wang. Quals: Corpus equilibrium for universal forecasting via pattern quantization and learnability synchronization. *arXiv preprint arXiv:2609.20156*, 2026b.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with 1k parameters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30211–30226. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24n.html.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814. PMLR, 2025. URL https://proceedings.mlr.press/v267/lin25e.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Zhen Liu, Yucheng Wang, Boyuan Li, Junhao Zheng, Emadeldeen Eldele, Min Wu, and Qianli Ma. A unified shape-aware foundation model for time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23972–23980, 2026. doi: 10.1609/aaai.v40i28.39574.

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Xiang Ma, Taihua Chen, Pengcheng Wang, Xuemei Li, and Caiming Zhang. ReCast: Reliability-aware codebook-assisted lightweight time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24299–24307, 2026. doi: 10.1609/aaai.v40i29.39610.

Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. HIVE-COTE 2.0: A new meta ensemble for time series classification. *Machine Learning*, 110:3211–3243, 2021. doi: 10.1007/s10994-021-06057-9. URL https://doi.org/10.1007/s10994-021-06057-9.

Matthew Middlehurst, Aiden Rushbrooke, Ali Ismail-Fawaz, Maxime Devanne, Germain Forestier, Angus Dempster, Geoffrey I. Webb, Christopher Holder, and Anthony Bagnall. The multiverse of time series machine learning: An archive for multivariate time series classification. *arXiv preprint arXiv:2603.20352*, 2026. URL https://arxiv.org/abs/2603.20352.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron J. Elmore, and Michael J. Franklin. Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, 2022. doi: 10.14778/3551793.3551830. URL https://www.vldb.org/pvldb/vol15/p2774-paparrizos.pdf.

Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. TAB: Unified benchmarking of time series anomaly detection methods. *Proceedings of the VLDB Endowment*, 18(9):2775–2789, 2025. doi: 10.14778/3746405.3746407. URL https://www.vldb.org/pvldb/vol18/p2775-hu.pdf.

Ali Seyfi, Jean-Francois Rajotte, and Raymond Ng. Generating multivariate time series with COmmon Source CoordInated GAN (COSCI-GAN). In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2375. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/d3408794e41dd23e34634344d662f5e9-Abstract-Conference.html.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2024.

Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. BLAST: Balanced sampling time series corpus for universal forecasting models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2502–2513, 2025.

Sabera J Talukder, Yisong Yue, and Georgia Gkioxari. TOTEM: TOkenized time series EMbeddings for general time series analysis. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QlTLkH6xRC.

Xiaoyu Tao, Shilong Zhang, Mingyue Cheng, Daoyu Wang, Tingyue Pan, Bokai Pan, Changqing Zhang, and Shijin Wang. From values to tokens: An LLM-driven framework for context-aware time series forecasting via symbolic discretization. *arXiv preprint arXiv:2508.09191*, 2025. doi: 10.48550/arXiv.2508.09191. URL https://arxiv.org/abs/2508.09191.

Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, 2022. doi: 10.14778/3514061.3514067. URL https://www.vldb.org/pvldb/vol15/p1201-tuli.pdf.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Fei Wang, Yujie Li, Zezhi Shao, Chengqing Yu, Yisong Fu, Zhulin An, Yongjun Xu, and Xueqi Cheng. ARIES: Relation assessment and model recommendation for deep time series forecasting. *arXiv preprint arXiv:2509.06060*, 2025.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks*, pp. 1578–1585. IEEE, 2017. doi: 10.1109/IJCNN.2017.7966039. URL https://doi.org/10.1109/IJCNN.2017.7966039.

Yunshi Wen, Tengfei Ma, Tsui-Wei Weng, Lam M. Nguyen, and Anak Agung Julius. Abstracted shapes as tokens - a generalizable and interpretable model for time-series classification. In *Advances in Neural Information Processing Systems*, volume 37, pp. 92246–92272, 2024. doi: 10.52202/079017-2929. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a7c57c8942b7d2adba1ac615ba0e60b9-Abstract-Conference.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.12261.

Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. COT-GAN: Generating sequential data via causal optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/641d77dd5271fca28764612a028d9c8e-Abstract.html.

Jie Yang, Yifan Hu, Yuante Li, Kexin Zhang, Kaize Ding, and Philip S. Yu. From observations to states: Latent time series forecasting, 2026. URL https://arxiv.org/abs/2602.00297.

Zihao Yao, Qi Zheng, Jiankai Zuo, and Yaying Zhang. Towards a unified generative model for scarce time series with domain experts. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=iXh2dmdt8v.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5509–5519, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html.

Mingyang Yu, Xiahui Guo, Peng Chen, Zhenkai Li, and Yang Shu. Towards measuring and modeling geometric structures in time series forecasting via image modality. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2390–2398, 2025. doi: 10.1145/3746027.3754794.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

Wenzhen Yue, Yong Liu, Hao Wang, Haoxuan Li, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. OLinear: A linear model for time series forecasting in orthogonally transformed domain. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/571c7e164fb1ffbcf2f84a63784451ec-Abstract-Conference.html.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022. doi: 10.1609/aaai.v36i8.20881. URL https://ojs.aaai.org/index.php/AAAI/article/view/20881.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Haoyi Zhou, Xin Xue, Tianyu Chen, Lanhao Li, Lijun Sun, and Jianxin Li. Winformer: Transcending pairwise similarity for time-series generation. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://icml.cc/virtual/2026/poster/63196.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42625–42643. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhou23i.html.

A NOTATION

Table 7 summarizes the notation. Indices h, i, and d index scales, intervals, and variables.

Table 7: Notation for the MarkovTS formulation, representation, and generator.

Symbol	Meaning
$\mathcal{D}_{\mathrm{pre}}, \mathcal{Q}_{\mathrm{pre}}$	Pretraining corpus and its set of source-domain distributions.
q_*, R_*, T_*	Unseen target distribution, evaluation reference pool, and disjoint target test set. Each generated sample uses one member of R_*.
$X_{\mathrm{ref}}, K_{\mathrm{eval}}$	Inference reference and number of references used in repeated single-reference evaluation.
$X \in \mathbb{R}^{D\times L}$	A length-L time series comprising D aligned variable sequences, each modeled independently.
$\bar{X}, P, n_h$	Standardized input, canonical patch dimension, and number of padded intervals at scale h.
$\mathcal{H}, H, h$	Set, number, and index of temporal scales.
w_h	Temporal span at scale h, ordered from coarse to fine.
$p_i^h, \widetilde{p}_i^h, \mathcal{V}_i^h$	Fixed-dimensional interval patch, its locally standardized form, and its nonempty-bin index set.
$q_i^{S,h}, E_S, e_k^S, s_i^h$	State query, state codebook, its k-th entry, and assigned state codebook vector.
N_S, N_T	State and transition codebook sizes; denoted K_S, K_T in the implementation configuration.
g_i^h, b_i^h, r_i^h	Patch-pair relation, state-pair relation, and transition residual.
$q_i^{T,h}, E_T, e_k^T, \tau_i^h$	Transition query, transition codebook, its k-th entry, and assigned transition codebook vector.
a_i^h, R	Continuous patch feature and common pair-relation output dimension.
S_d^h, T_d^h	Ordered sequences of state and transition vectors for variable d at scale h.
$C(\bar{X}), C_d(\bar{X})$	Condition collection and single-variable condition.
$C_{\mathrm{seq},d}^h, C_d^h$	Ordered scale-h condition sequence and processed condition memory.
X_0, X_1, X_t, U_t	Gaussian source, clean sample, interpolation state, and target velocity $X_1 - X_0$.
$v_\theta, v_{\bar{\theta}}$	Velocity fields with learned and exponential-moving-average parameters.
M, M^h, P_h	Full-resolution mask, scale-h mask, and masked pooling operator.
$j, Z_{d,k}^{(j)}, t_k$	Sample index, variable-d integration state, and time at step k.
μ_d, σ_d, s_d	Reference mean, standard deviation, and effective rescaling factor.
N_{FE}	Number of Euler function evaluations for sampling.

B ADDITIONAL METHOD DETAILS

We specify preprocessing, representation objectives, and conditional flow training and sampling.

B.1 PREPROCESSING AND ALIGNED PATCHIFICATION

For each variable d, let Ω_d index the finite observations in the reference window. The reference mean and population standard deviation are

$$\mu_d = \frac{1}{|\Omega_d|}\sum_{t\in\Omega_d} x_{t,d}, \qquad \sigma_d = \sqrt{\frac{1}{|\Omega_d|}\sum_{t\in\Omega_d}(x_{t,d}-\mu_d)^2}, \tag{11}$$

For a nondegenerate reference, $s_d = \sigma_d$. Normalization uses $\bar{x}_{t,d} = (x_{t,d} - \mu_d)/(s_d + \epsilon)$ with stabilizer ϵ; output rescaling uses $\hat{x}_{t,d} = s_d\hat{\bar{x}}_{t,d} + \mu_d$. Flow pretraining instead computes statistics over each finite training row before window extraction, without further standardization.

For padded length L_{pad}, scale h contains $n_h = L_{\mathrm{pad}}/w_h$ intervals. Each interval i is partitioned into P consecutive bins $B_{i,b}^h$ of w_h/P positions. With validity mask M_t, let $c_{i,b}^h = \sum_{t\in B_{i,b}^h} M_t$,

$\check{p}^h_{i,b} = \sum_{t \in B^h_{i,b}} M_t \bar{x}_t / \max(1, c^h_{i,b})$, and $\mathcal{V}^h_i = \{b : c^h_{i,b} > 0\}$. The canonical patch is

$$p^h_{i,b} = \begin{cases} \check{p}^h_{i,b}, & b \in \mathcal{V}^h_i, \\ \dfrac{\sum_{b' \in \mathcal{V}^h_i} \check{p}^h_{i,b'}}{\max(1, |\mathcal{V}^h_i|)}, & b \notin \mathcal{V}^h_i. \end{cases} \tag{12}$$

When $\mathcal{V}^h_i$ is nonempty, the second case fills an empty bin with the mean of the nonempty-bin means in its interval; a fully empty interval evaluates to zero and is masked as inactive. The state branch computes local mean and variance from $\mathcal{V}^h_i$ only, so filled entries become exactly zero after local standardization and remain in normalized-shape reconstruction. Raw and multiscale waveform reconstruction instead remove them with bin-level masks. Patch-pair features retain source-standardized levels and amplitudes without local standardization.

At scales $h \in \mathcal{H}$, aligned intervals have temporal spans $w_h \in \{256, 128, 64, 32, 16\}$ and use a base patch size of 16. An interval and its state are active when at least half its samples are valid. Transitions require both adjacent states to be valid; invalid intervals break the transition sequence. Aligned variables are separately encoded on shared interval boundaries.

B.2 Representation learning and codebook optimization

The patch-pair and state-pair networks f_P and f_S in Eq. 6 have independent parameters and a shared R-dimensional output space. State-code inputs to f_S are detached, but f_S remains trainable.

State warm-up learns local morphology before transition learning freezes the state tokenizer and codebook. Both codebooks span all scales; decoders serve only representation pretraining.

B.2.1 State warm-up

Let m^h_{ui} indicate patch validity for window u, interval i, and scale h, with $N_h = \max(1, \sum_{u,i} m^h_{ui})$ and query dimension $d_q = 64$. State warm-up minimizes

$$\begin{aligned} \mathcal{L}_S = \mathcal{L}_{\text{shape}} + \lambda_S \mathcal{L}^S_{\text{commit}} + \lambda_{\text{iso}} \mathcal{L}_{\text{iso}} \\ + \lambda_{\text{loc}} \mathcal{L}_{\text{loc}} + \lambda_- \rho_S(s) \mathcal{L}_-. \end{aligned} \tag{13}$$

A shared linear decoder maps states to $\widehat{p}^h_{ui} \in \mathbb{R}^P$, with reconstruction and commitment losses

$$\begin{aligned} \mathcal{L}_{\text{shape}} &= \frac{1}{H} \sum_{h=1}^{H} \frac{\sum_{u,i} m^h_{ui} \|\widehat{p}^h_{ui} - \widetilde{p}^h_{ui}\|_1}{P N_h}, \\ \mathcal{L}^S_{\text{commit}} &= \frac{1}{H} \sum_{h=1}^{H} \frac{\sum_{u,i} m^h_{ui} \|q^{S,h}_{ui} - \text{sg}(s^h_{ui})\|_2^2}{d_q N_h}. \end{aligned} \tag{14}$$

The operator sg stops gradients. The denominator PN_h reflects that normalized-shape reconstruction averages all P bins of each active patch, including empty-bin targets that equal zero after local standardization based only on nonempty bins. Commitment updates queries, while codebooks use exponential moving averages (EMA).

Query isotropy discourages concentration in a small number of embedding directions. For $n \geq 2$ sampled valid queries at one scale, let $z_a = \text{norm}(q^{S,h}_a)$, $\mu = n^{-1} \sum_a z_a$, and $\Sigma = n^{-1} \sum_a (z_a - \mu)(z_a - \mu)^\top$. Define

$$\delta = \max\left(\text{sg}\left(\frac{\text{tr}\, \Sigma}{d_q} \right), 10^{-6} \right), \qquad A_{jk} = \frac{\Sigma_{jk}}{\sqrt{\max(\Sigma_{jj}, 10^{-6}) \max(\Sigma_{kk}, 10^{-6})}}. \tag{15}$$

The per-scale penalty is

$$I_h = \|\mu\|_2^2 + \frac{1}{d_q} \sum_{j=1}^{d_q} \left(\frac{\Sigma_{jj}}{\delta} - 1 \right)^2 + \frac{1}{d_q} \sum_{j \neq k} A^2_{jk}. \tag{16}$$

$\mathcal{L}_{\text{iso}}$ averages I_h equally over scales with at least two valid queries, with at most 2,048 queries each.

Local consistency regularizes queries and assignments under shape perturbations. For a normalized patch $\widetilde{p}$, construct $\widetilde{p}' = \mathrm{StdNorm}(\widetilde{p} + 0.03\epsilon)$, where $\epsilon \sim \mathcal{N}(0, I_P)$ and StdNorm subtracts the patch mean and divides by its population standard deviation clamped below at 10^{-6}. Let q, q' be their state queries. With detached, unit-normalized prototypes and temperature $\eta = 0.1$, define

$$p_k(q) = \frac{\exp(q^\top \mathrm{sg}(e_k^S)/\eta)}{\sum_l \exp(q^\top \mathrm{sg}(e_l^S)/\eta)}, \qquad \ell_{\mathrm{loc}} = 1 - \cos(q, q') + \mathrm{JS}(p(q), p(q')). \tag{17}$$

Jensen–Shannon divergence JS clamps probabilities at 10^{-8} before logarithms. Both queries receive gradients. $\mathcal{L}_{\mathrm{loc}}$ averages up to 256 valid patches per scale, then nonempty scales equally.

Delayed negative separation distinguishes dissimilar shapes after warm-up. From at most 128 valid state queries per scale, construct the set $\mathcal{P}_h$ of unordered pairs that satisfy all four criteria: different training windows, different selected code IDs, prototype cosine similarity at most 0.5, and normalized-shape cosine similarity at most 0.5. The loss is

$$\mathcal{L}_- = \frac{1}{|\mathcal{H}_-|} \sum_{h \in \mathcal{H}_-} \frac{1}{|\mathcal{P}_h|} \sum_{(a,b) \in \mathcal{P}_h} [\cos(q_a^{S,h}, q_b^{S,h}) - 0.2]_+^2, \tag{18}$$

where $\mathcal{H}_-$ contains scales with eligible pairs and $[x]_+ = \max(x, 0)$. Pair selection is detached; gradients pass only through query similarities. At optimizer step s, the weight is

$$\rho_S(s) = \min\left(1, \max\left(0, \frac{s - 5000}{5000}\right)\right). \tag{19}$$

All state-query subsampling is uniform without replacement within each device's batch. A regularizer contributes zero when no scale supplies enough valid state queries or pairs.

B.2.2 Transition learning

Transition learning optimizes the continuous patch-pair encoder, state-pair network, causal context encoder, and decoder using waveform reconstruction, residual matching, and query regularization:

$$\mathcal{L}_{\mathrm{enc}} = \mathcal{L}_{\mathrm{raw}} + \lambda_{\mathrm{ms}} \mathcal{L}_{\mathrm{multiscale}} + \lambda_T \mathcal{L}_{\mathrm{match}}^T + \lambda_{\mathrm{vc}} \mathcal{L}_{\mathrm{varcov}}^T. \tag{20}$$

The decoder reconstructs waveforms coarse to fine using learned queries at the coarsest scale, an interpolation bridge with a residual multilayer perceptron at finer scales, and scale-specific linear heads producing P values per interval. Each window follows one route across all scales: joint, state, or transition-driven reconstruction, with probabilities 0.30, 0.50, and 0.20. The state route uses only the current state without the cross-scale bridge. The transition-driven route uses preceding states and transitions through the current interval, initializing from a state when preceding information is unavailable; the joint route additionally accesses the current state.

Let v_{uib}^h indicate that bin b and its enclosing patch are valid. For route predictions $\widehat{p}_{uib}^h$,

$$\begin{aligned} R_h &= \frac{\sum_{u,i,b} v_{uib}^h \left(|\widehat{p}_{uib}^h - p_{uib}^h| + (\widehat{p}_{uib}^h - p_{uib}^h)^2 \right)}{2 \max(1, \sum_{u,i,b} v_{uib}^h)}, \\ \mathcal{L}_{\mathrm{raw}} &= R_H, \qquad \mathcal{L}_{\mathrm{multiscale}} = \frac{1}{H-1} \sum_{h=1}^{H-1} R_h. \end{aligned} \tag{21}$$

Scale H is finest. Because v_{uib}^h combines bin and patch validity, both waveform losses exclude empty bins and normalize by the number of valid bins. This differs from normalized-shape reconstruction, which retains their zero-valued targets. Each loss is estimated over the sampled decoding route.

For each valid adjacent pair, the layer-normalized residual r_i^h (Eq. 6) and $\log(1 + \|r_i^h\|_2)$ are projected to a local query, contextualized by causal attention before quantization. The selected vector predicts $\widehat{r}_i^h = W_R \tau_i^h$, matched to the detached target $r_* = \mathrm{sg}(r_i^h)$ by

$$\ell_T(r_*, \widehat{r}) = 2 - 2\cos(r_*, \widehat{r}) + 0.1 \left[\log(1 + \|r_*\|_2) - \log(1 + \|\widehat{r}\|_2)\right]^2. \tag{22}$$

Table 8: Objective weights used in state warm-up and transition learning. The negative-separation weight follows the ramp $\rho_S(s)$.

Stage	Objective	Weight
State	Normalized-shape reconstruction	1.00
	Code commitment	1.00
	Query isotropy	0.10
	Perturbation consistency	0.05
	Delayed negative separation	$0.02\rho_S(s)$
Transition	Full-resolution reconstruction	1.00
	Multiscale reconstruction	1.00
	Direction–magnitude matching	1.00
	Query variance–covariance	0.05

Cosine similarity uses $\epsilon = 10^{-6}$; residual norms are clamped below at 10^{-6} before $\log(1+\cdot)$. $\mathcal{L}^T_{\text{match}}$ averages over valid transitions within each scale, then over scales. Gradients pass through the straight-through transition vector and prediction head; only this loss detaches its target.

Variance–covariance regularization follows the spread and decorrelation principles of VICReg (Bardes et al., 2022) and acts on both local and contextual transition queries. For either query set at scale h, sample $n \geq 2$ valid queries and set $z_a = \sqrt{d_q}\,\text{norm}(q_a)$, $\bar{z} = n^{-1}\sum_a z_a$. Define

$$\sigma_j = \sqrt{\frac{1}{n}\sum_a (z_{aj} - \bar{z}_j)^2 + 10^{-4}}, \qquad V = \frac{1}{n-1}\sum_a (z_a - \bar{z})(z_a - \bar{z})^\top, \tag{23}$$

and

$$J_h(q) = \frac{1}{d_q}\sum_j [1-\sigma_j]_+ + \frac{1}{d_q}\sum_{j\neq k} V_{jk}^2, \qquad \mathcal{L}^T_{\text{varcov}} = \frac{1}{2}\left(\langle J_h(q^{T,\text{local}})\rangle_h + \langle J_h(q^{T,\text{context}})\rangle_h\right). \tag{24}$$

Each average uses 2–2,048 valid queries per eligible scale and query set.

B.2.3 Codebook optimization

Both branches use unit-normalized queries and codebook entries with deterministic nearest-code assignment (Eqs. 5 and 7) and straight-through gradients (van den Oord et al., 2017), e.g., $q_i^{S,h} + \text{sg}(s_i^h - q_i^{S,h})$. EMA statistics instead use Sinkhorn-balanced hard assignments a_{ik} over equal numbers of valid queries sampled from each nonempty scale. With decay $\beta = 0.99$,

$$\begin{aligned} n_k &\leftarrow \beta n_k + (1-\beta)\sum_i a_{ik}, \\ m_k &\leftarrow \beta m_k + (1-\beta)\sum_i a_{ik} q_i, \\ e_k &\leftarrow \text{norm}\left(\frac{m_k}{n_k + \epsilon}\right), \end{aligned} \tag{25}$$

where $\text{norm}(z) = z/\max(\|z\|_2, 10^{-6})$. EMA uses nearest-code assignments when sampled queries are fewer than codebook entries and reinitializes persistently inactive entries from current queries. Neighbor smoothing, periodic whitening and repulsion, and covariance correction modify EMA codebook updates separately from gradient-based objectives. Objective weights appear in Table 8; training schedules and optimizer settings are in Appendix B.5.

B.3 Multiscale state–transition condition

After representation pretraining, the encoder and both codebooks are frozen. For variable d at scale h, the ordered condition sequence interleaves states and transitions in temporal order:

$$C^h_{\text{seq},d} = [s^h_{d,0}, \tau^h_{d,1}, s^h_{d,1}, \ldots, \tau^h_{d,n_h-1}, s^h_{d,n_h-1}]. \tag{26}$$

State $s^h_{d,i}$ lies at interval center $(i+\frac{1}{2})w_h$ and transition $\tau^h_{d,i}$ at boundary iw_h. Masks retain active states and transitions within valid segments.

Separate state and transition projections map the elements of $C^h_{\mathrm{seq},d}$ into a common conditioning space. For any active hierarchy suffix, each fine-scale element incorporates the mean of overlapping projected elements from its nearest active coarser scale $\pi(h)$. A learned layer fuses this summary with the local element, and within-scale self-attention produces the scale-h condition memory C^h_d.

Condition injection follows Eq. 9 with masked interval pooling. Active scales update coarse to fine with query repooling. Attention respects validity, segment identity, and temporal overlap.

B.4 Conditional flow optimization and zero-shot sampling

For single-variable inputs $X_0, X_1, X_t \in \mathbb{R}^L$, condition C, validity mask M, and target $U_t = X_1 - X_0$ (Eq. 1), the full-resolution loss averages over valid positions per sequence, then over the batch:

$$\mathcal{L}_{\mathrm{full}} = \mathbb{E}\left[\frac{\|M \odot (v_\theta(X_t, t, C) - U_t)\|_2^2}{\|M\|_1 + \epsilon}\right]. \tag{27}$$

For every active scale h, masked pooling defines the target $U^h_t = \mathrm{Pool}_h(U_t)$ for an auxiliary velocity head $v_{\theta,h}$. The complete objective is

$$\mathcal{L}_{\mathrm{flow}} = \mathcal{L}_{\mathrm{full}} + \sum_{h \in \mathcal{H}_{\mathrm{act}}} \lambda_h \mathbb{E}\left[\frac{\|M^h \odot (v_{\theta,h}(X_t, t, C) - U^h_t)\|_2^2}{\|M^h\|_1 + \epsilon}\right]. \tag{28}$$

Training uses all five scales or a suffix starting at span 128, 64, 32, or 16; inactive scales contribute neither loss nor gradient. For $Y = (X_t, t, C)$ and fixed linear masked pooling P_h, squared-loss population targets satisfy $v^\star(Y) = \mathbb{E}[U_t \mid Y]$ and $v^\star_h(Y) = \mathbb{E}[P_h U_t \mid Y] = P_h v^\star(Y)$ by linearity. This consistency applies on the training-condition support (Lipman et al., 2023).

Sampling follows Section 4.4, with reference normalization and rescaling defined in Appendix B.1.

B.5 Training and implementation settings

Tables 9 and 10 summarize training and model settings with $P = 16$ canonical values. State warm-up starts randomly; transition learning resets the optimizer and scheduler, following Appendix B.2. Flow optimization freezes the encoder and both codebooks.

Flow training uses univariate windows of length 512–4096 in increments of 256, with one length per batch. The final phase assigns probability 0.5 to length 4096 and total probability 0.7 to lengths at least 3072. The full-resolution loss has unit weight; auxiliary weights appear in Table 10.

The final flow phase initializes from a step-2,000 generator EMA checkpoint pretrained on the same GIFT-Eval Pretrain corpus, resetting the optimizer, scheduler, and EMA state. Linear warm-up precedes cosine decay over a planned 4,000-step schedule to a minimum learning-rate ratio of 0.1. Evaluation uses the checkpoint after 2,400 additional updates. Sampling uses EMA parameters, nearest-code assignments, and the integration settings in Table 10; the main length-512-to-168 evaluation uses seeds 2021–2025.

Pretraining uses NVIDIA A800 80GB GPUs; downstream experiments and generation use NVIDIA RTX 4090 24GB GPUs. Before stage freezing, the representation model, including its reconstruction decoder, has 22.87M parameters; the flow generator has 24.57M. The transition context encoder has six causal Transformer layers of width 384 with eight heads; the reconstruction decoder has three layers per scale. Encoder training standardizes a finite segment of at least 512 points selected from each 4096-point row, whereas flow training standardizes the complete training row before cropping. Both use population standard deviations and a constant-scale fallback of one.

AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, and $\epsilon = 10^{-8}$ throughout. Encoder schedules use linear warm-up and zero final rates. State–transition training resumes the same schedule at steps 120,000–160,000 with covariance-regularized codebook updates. Flow training uses activation checkpointing.

B.6 Generation throughput

Table 11 reports throughput for the selected EMA model with 16 Euler steps on four RTX 4090 24GB GPUs. Values are medians over 12 datasets and five seeds, aggregated across GPUs and

Table 9: Pretraining settings by stage. All stages use AdamW, bfloat16, and gradient clipping at norm 1.

Stage	Updates	Learning rate	LR warm-up	Effective batch
State warm-up	20,000	10^{-4}	200	84 (4×21)
State–transition	160,000	10^{-4}	1,600	84 (4×21)
Flow, final phase	2,400	10^{-5}	100	10,240 (8×1280)

Table 10: Representation and conditional flow configurations used by MarkovTS.

Component	Configuration
Representation configuration	
Base patch / temporal spans	16 / $\{16, 32, 64, 128, 256\}$
State codebook	$K_S = 1024$, dimension 64, EMA 0.99
Transition codebook	$K_T = 1024$, dimension 64, EMA 0.99
Conditional flow: final optimization phase	
Flow backbone	width 288, depth 8, heads 8, MLP ratio 4
Auxiliary flow losses	$\lambda_h = 0.1$ for each active scale
Flow parameter EMA	0.9995
Euler sampling	16 function evaluations

counting complete sequences before evaluation-window extraction. Flow-only timing covers numerical integration; end-to-end timing also includes condition preparation and output processing, including CPU-side checks and file writing.

C Theoretical Properties of MarkovTS

We analyze state-identity stability and reconstruction error under transition residual matching.

C.1 Invariant and stable state identities

Proposition 1 (Affine-invariant state assignment). Let $p \in \mathbb{R}^P$ be a canonical patch with nonempty-bin set $\mathcal{V}$ and $\sigma_{\mathrm{v}}(p) > 0$, where μ_{v} and σ_{v} use only indices in $\mathcal{V}$. Define $N_{\mathrm{v}}(p) = (p - \mu_{\mathrm{v}}(p)\mathbf{1})/\sigma_{\mathrm{v}}(p)$. For any $a > 0$ and $b \in \mathbb{R}$,

$$N_{\mathrm{v}}(ap + b\mathbf{1}) = N_{\mathrm{v}}(p). \tag{29}$$

Thus, a state query computed solely from $N_{\mathrm{v}}(p)$ retains its deterministic code assignment under positive rescaling and additive offsets. For a unit-normalized query q and unit-normalized code vectors e_k, maximizing cosine similarity is equivalent to nearest-neighbor assignment on the unit sphere. If

$$\gamma = q^\top e_{k^\star} - \max_{\ell \neq k^\star} q^\top e_\ell > 0, \tag{30}$$

then every unit query q' satisfying $\|q' - q\|_2 < \gamma/2$ retains assignment $k^\star$.

Proof. The validity mask, and hence $\mathcal{V}$, is unchanged by the affine transformation. For $a > 0$, $\mu_{\mathrm{v}}(ap + b\mathbf{1}) = a\mu_{\mathrm{v}}(p) + b$ and $\sigma_{\mathrm{v}}(ap + b\mathbf{1}) = a\sigma_{\mathrm{v}}(p)$, which proves Eq. 29. For unit vectors, $\|q - e_k\|_2^2 = 2 - 2q^\top e_k$, so maximizing cosine similarity minimizes Euclidean distance. For every $\ell \neq k^\star$,

$$(q')^\top (e_{k^\star} - e_\ell) \geq \gamma - 2\|q' - q\|_2 > 0, \tag{31}$$

by Cauchy–Schwarz and $\|e_k\|_2 = 1$. Thus $k^\star$ remains the unique maximizer. □

Invariance requires an inactive standard-deviation floor for both patches. With the assignment margin, it supports morphological-code reuse under level and amplitude changes.

C.2 Transition residual and relation fidelity

Proposition 2 (Transition-residual control). Let g be the patch-pair relation, b the corresponding state-pair relation, $r = g - b$, and $\hat{g} = b + \hat{r}$ the relation reconstructed from an assigned transition

Table 11: Generation throughput aggregated across four RTX 4090 24GB GPUs. Rates count complete sequences per second; batch sizes are per GPU.

Length	Batch/GPU	Flow-only	End-to-end
512	128	323.02	128.89
1024	96	221.13	74.40
2048	64	166.73	43.75

vector. Then

$$\|g - \hat{g}\|_2^2 = \|r - \hat{r}\|_2^2. \tag{32}$$

Assume $0 < m \le \|r\|_2, \|\hat{r}\|_2 \le M$ and $\alpha > 0$. For

$$\ell_T^\alpha(r, \hat{r}) = 2 - 2\cos(r, \hat{r}) + \alpha\big[\log(1 + \|r\|_2) - \log(1 + \|\hat{r}\|_2)\big]^2, \tag{33}$$

the residual error satisfies

$$\|r - \hat{r}\|_2^2 \le \kappa \ell_T^\alpha(r, \hat{r}), \qquad \kappa = \max\left\{M^2, \frac{(1+M)^2}{\alpha}\right\}. \tag{34}$$

Proof. The first identity follows from $g - (b + \hat{r}) = r - \hat{r}$. Writing $a = \|r\|_2$, $\hat{a} = \|\hat{r}\|_2$, and $\vartheta = \cos(r, \hat{r})$ gives

$$\|r - \hat{r}\|_2^2 = a\hat{a}(2 - 2\vartheta) + (a - \hat{a})^2. \tag{35}$$

The first term is at most $M^2(2 - 2\vartheta)$. By the mean-value theorem for $z \mapsto \log(1 + z)$ on $[0, M]$, $|a - \hat{a}| \le (1 + M)|\log(1 + a) - \log(1 + \hat{a})|$. Combining the bounds proves Eq. 34. □

Under the stated norm bounds, the direction–magnitude objective in Eq. 22 bounds the error of patch-pair relations reconstructed from state-pair relations and predicted transition residuals.

D Experimental Details

D.1 Pretraining corpus and generation datasets

We pretrain on GIFT-Eval Pretrain (Aksu et al., 2024), excluding the 12 generation datasets in Table 12 before training-window extraction. The pretraining corpus follows the official GIFT-Eval Pretrain dataset listing linked in Section 5.1.

The 12 held-out public GluonTS datasets (Huang et al., 2025) cover energy, transportation, environmental measurements, and economic activity. Table 12 summarizes their source data, sampling frequencies, and temporal coverage before window extraction. The main comparison evaluates non-overlapping length-168 windows extracted from length-512 references and generated outputs.

Generation uses target training references and frozen MarkovTS parameters. Test data are excluded from generation and downstream fitting; validation follows the task-specific protocols below.

D.2 Generation evaluation protocol and baselines

For the main evaluation, we select $K_{\text{eval}} = 100$ valid length-512 univariate reference windows extracted from the channels of the target training split. The frozen MarkovTS encoder assigns deterministic nearest-code states and transitions, and one fixed flow checkpoint generates length-512 outputs from independent Gaussian sources. Each output yields three fixed, non-overlapping length-168 windows in temporal order; the final eight points are discarded. For each reference, we generate outputs until at least 50 windows are available, order the candidates first by Gaussian draw and then by temporal position, and retain the first 50 (thus discarding the final candidate window). This gives $N = 5000$ evaluation windows per dataset and seed. Table 1 evaluates these windows against the test split over seeds 2021–2025. GraphFormer and reference-dependent codebook sampling are not used in this main protocol.

TimeVQVAE-C uses dataset identity as the class label during multi-domain training. ImagenFew results report five-seed means and sample standard deviations under the same length-168 evaluation (Gonen et al., 2025). The main table compares RBF-MMD and flat-KL. Wins are counted over the 24 MMD/flat-KL comparisons after rounding means to three decimals, crediting all tied methods.

Table 12: Held-out datasets for generation evaluation.

Dataset	Application	Frequency	Scale	Content and temporal extent
Electricity	Energy	Hourly	321 clients	Electricity consumption, 2012–2014
Solar	Energy	Hourly	137 series	Solar power production in Alabama, 2006
Wind	Energy	4 seconds	1 long series	AEMO wind production, beginning August 2019
Traffic	Transportation	10 minutes	963 sensors	Freeway-lane occupancy in the San Francisco Bay Area over 440 days
Taxi	Transportation	30 minutes	1,214 locations	New York City taxi rides in January 2015 and January 2016
Pedestrian	Transportation	Hourly	66 sensors	Pedestrian counts in Melbourne, May 2009–April 2020
Air Quality	Environment	Hourly	270 series	Pollutants from 59 Beijing/London stations, 2017–2018
Temperature	Environment	Daily	422 stations	Mean temperature measurements across Australia, 2015–2017
Rain	Environment	Daily	422 stations	Rainfall measurements from the same Australian station collection
NN5	Banking	Daily	111 series	Cash withdrawals from automated teller machines
Fred-MD	Macroeconomics	Monthly	107 series	Transformed U.S. macroeconomic indicators
Exchange	Finance	Daily	8 series	Exchange rates among eight currencies

D.3 Evaluation metrics

Generation fidelity. Let $\mathcal{X}_r = \{\mathbf{x}_i\}_{i=1}^{n_r}$ and $\mathcal{X}_g = \{\widetilde{\mathbf{x}}_j\}_{j=1}^{n_g}$ denote the real and generated evaluation windows, where each $\mathbf{x} \in \mathbb{R}^{L\times C}$. In the main length-512-to-168 benchmark, channels are evaluated as univariate windows ($C = 1$). Before computing MMD and flat-KL, both sets are standardized by the pooled mean μ_r and standard deviation σ_r of the real evaluation set:

$$\mathbf{x}' = \frac{\mathbf{x} - \mu_r}{\sigma_r + 10^{-7}}. \tag{36}$$

Both metrics are lower-is-better.

Maximum Mean Discrepancy (MMD) compares complete vectorized windows. With $\mathbf{z}_i = \mathrm{vec}(\mathbf{x}'_i)$, $\widetilde{\mathbf{z}}_j = \mathrm{vec}(\widetilde{\mathbf{x}}'_j)$, and the radial-basis-function kernel $k(\mathbf{u}, \mathbf{v}) = \exp(-\gamma\|\mathbf{u} - \mathbf{v}\|_2^2)$, the evaluator reports the biased squared estimate

$$\begin{aligned}\widehat{\mathrm{MMD}}^2 = &\frac{1}{n_r^2}\sum_{i,i'} k(\mathbf{z}_i, \mathbf{z}_{i'}) + \frac{1}{n_g^2}\sum_{j,j'} k(\widetilde{\mathbf{z}}_j, \widetilde{\mathbf{z}}_{j'}) \\ &- \frac{2}{n_r n_g}\sum_{i,j} k(\mathbf{z}_i, \widetilde{\mathbf{z}}_j),\end{aligned} \tag{37}$$

The kernel uses $\gamma = 1/(LC)$, and tables retain the conventional label MMD.

Flattened Kullback–Leibler divergence (flat-KL) discards the sample, time, and channel indices and pools all scalar observations. The range of the real set defines $B = 50$ equal-width histogram bins. Let p_b and q_b be the normalized real and generated masses in bin b, and let $\bar{q}_b = (q_b + \epsilon)/\sum_{a=1}^{B}(q_a + \epsilon)$ with $\epsilon = 10^{-9}$. The reported divergence is

$$\text{flat-KL}(\mathcal{X}_r \| \mathcal{X}_g) = \sum_{b=1}^{B} p_b \log \frac{p_b}{\bar{q}_b}. \tag{38}$$

Context-Fréchet Inception Distance (Context-FID) supplies the complementary representation-space comparison with ImagenFew (Gonen et al., 2025). For each dataset d, a separately trained TS2Vec encoder E_d (Yue et al., 2022) maps a sequence to a contextual embedding. If $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ and $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ are the empirical means and covariances of real and generated embeddings, respectively, then

$$\mathrm{ContextFID} = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|_2^2 + \mathrm{Tr}\Big(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\Big). \tag{39}$$

Lower Context-FID is better. TS2Vec only encodes evaluation samples; MarkovTS remains fixed.

Forecasting utility. For N test windows, forecast horizon H, and C variables, let y_{ihc} and $\widehat{y}_{ihc}$ denote a target and its prediction. We report

$$\mathrm{MAE} = \frac{1}{NHC}\sum_{i,h,c}|y_{ihc} - \widehat{y}_{ihc}|, \qquad \mathrm{MSE} = \frac{1}{NHC}\sum_{i,h,c}(y_{ihc} - \widehat{y}_{ihc})^2. \tag{40}$$

Both metrics are lower-is-better. Table 4 averages the scores over four forecast horizons.

Classification utility. Let n_c be the number of test instances in class c, $w_c = n_c/N$, and P_c, R_c denote one-vs-rest precision and recall. With $F_c = 2P_cR_c/(P_c + R_c)$, the reported metrics are

$$\begin{aligned} \mathrm{Accuracy} &= \frac{1}{N}\sum_i \mathbb{I}[\widehat{y}_i = y_i], & \mathrm{BAcc} &= \frac{1}{C}\sum_c R_c, \\ \mathrm{F1} &= \begin{cases} F_{c_{\min}}, & C = 2, \\ \sum_c w_c F_c, & C > 2, \end{cases} & \mathrm{AUROC} &= \begin{cases} \mathrm{AUROC}_1, & C = 2, \\ \sum_c w_c\,\mathrm{AUROC}_c, & C > 2. \end{cases} \end{aligned} \tag{41}$$

The indices $c_{\min}$ and $c_{\max}$ identify the least and most frequent test classes, with ties resolved by the smallest and largest encoded labels, respectively. Binary F1 treats $c_{\min}$ as positive, whereas binary AUROC uses the probability of encoded class 1. Multiclass AUROC uses prevalence-weighted one-vs-rest scores. Undefined precision, recall, and F1 contributions are set to zero.

Sensitivity and specificity use $R_{c_{\min}}$ and $R_{c_{\max}}$ for binary tasks, respectively. Following Multiverse, both use prevalence-weighted recall $\sum_c w_c R_c$ for multiclass tasks.

For predicted class probabilities $\widehat{p}_{ic}$, log loss is

$$\mathrm{LL} = -\frac{1}{N}\sum_i \log \widehat{p}_{i,y_i}. \tag{42}$$

All measures except log loss are higher-is-better. Dataset-level results follow the binary or multiclass convention before equal-weight averaging over the 23 datasets.

Anomaly-detection utility. Let s_t be an anomaly score, $y_t \in \{0, 1\}$ the pointwise label, and $\widehat{y}_t(\tau) = \mathbb{I}[s_t \geq \tau]$. At threshold τ, precision, recall, and false-positive rate are

$$P(\tau) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad R(\tau) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad F(\tau) = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{43}$$

Pointwise PR-AUC (A-P, implemented as average precision) and pointwise ROC-AUC (A-R) summarize the threshold sweep as

$$\text{A-P} = \sum_k (R_k - R_{k-1})P_k, \qquad \text{A-R} = \int_0^1 R(F)\,dF. \tag{44}$$

Following TAB/OpenTS (Qiu et al., 2025), the range-aware operator softly expands each ground-truth anomaly interval by a tolerance window and combines overlap with an existence reward. Let $P_w(\tau)$, $R_w(\tau)$, and $F_w(\tau)$ be the resulting range-aware rates for tolerance w, and let ℓ be the median ground-truth anomaly-range length. Range-aware PR-AUC and ROC-AUC are

$$\text{R-A-P} = \int P_\ell\,dR_\ell, \qquad \text{R-A-R} = \int R_\ell\,dF_\ell. \tag{45}$$

The volumes under the PR and ROC surfaces average range-aware areas over tolerance widths (Paparrizos et al., 2022):

$$\text{V-PR} = \frac{1}{2\ell+1}\sum_{w=0}^{2\ell}\int P_w\,dR_w, \qquad \text{V-ROC} = \frac{1}{2\ell+1}\sum_{w=0}^{2\ell}\int R_w\,dF_w. \tag{46}$$

All six anomaly-detection metrics are threshold-free and higher-is-better.

Table 13: Forecasting datasets and chronological splits. Mean length counts training time steps; training values count scalar observations.

Dataset	Application	Frequency	Ch.	Mean length	Training values	Split
ECL	Energy	Hourly	321	18,412	5,910,252	7:1:2
ETTh1	Energy	Hourly	7	8,640	60,480	6:2:2
ETTh2	Energy	Hourly	7	8,640	60,480	6:2:2
Exchange	Economy	Daily	8	5,311	42,488	7:1:2
Traffic	Transportation	Hourly	862	12,280	10,585,360	7:1:2
Weather	Environment	10 minutes	21	36,887	774,627	7:1:2

D.4 Forecasting datasets and evaluation protocol

The forecasting evaluation uses six datasets (Li et al., 2026a); Table 13 lists training statistics and chronological splits. ETTh1 and ETTh2 contain two-year transformer load and oil-temperature records.

Each forecasting dataset contains one continuous multivariate training sequence. Dataset size counts its length times its channel count before augmentation or sliding-window extraction.

Forecasting uses CATS, iTransformer, and TQNet (Kim et al., 2024; Liu et al., 2024; Lin et al., 2025), SparseTSF (Lin et al., 2024), and OLinear (Yue et al., 2025). Table 14 lists configurations.

Table 14: Forecasting-backbone configurations.

Backbone	Architecture	Regularization	Learning rate
CATS	$d = 128$; FFN 256; 16 heads; 3 layers; patch/stride 24/24	dropout 0.20; RevIN	3×10^{-4}
iTransformer	$d = 256$; FFN 512; 1 head; 1 layer	dropout 0.25; RevIN	5×10^{-5}
SparseTSF	period 4	—	6×10^{-4}
TQNet	$d = 512$; 4 heads; dataset cycle length	dropout 0; RevIN	7.5×10^{-5}
OLinear	$d = 256$; FFN 256; 2 layers; patch/stride 16/8	dropout 0.20; embedding 8	3.75×10^{-5}

Forecasting batch sizes are 64, except 16 for ECL and 32 for Weather; TQNet uses 16 throughout. Weight decay is 10^{-4} for iTransformer and zero otherwise.

Forecasting uses lookback 336 and horizons $\{96, 192, 336, 720\}$, with channel-wise normalization and seed 42. Models minimize MAE for at most 100 epochs, using a MultiStep schedule (milestones 1 and 25, factor 0.5), gradient-norm clipping at 5, and early-stopping patience 15. Validation is performed each epoch; the selected checkpoint is evaluated once on the real test split.

For forecasting augmentation, reference standard deviations below 10^{-6} are replaced by the corresponding training-split standard deviations, or by one if the training-split values are also below 10^{-6}.

For forecasting, a real training sequence is partitioned into consecutive, non-overlapping reference segments: full segments contain exactly 4096 time steps, followed by one optional remainder segment. A remainder of length $L_s < 4096$ is masked and right-padded to $G_s = \max(512, 256\lceil L_s/256 \rceil)$, and its generated output is cropped back to L_s. Each reference segment conditions exactly one generated segment. We concatenate the generated segments in their original order, producing one generated-only sequence with exactly the same length as the real sequence; no real values are inserted into this sequence. Sliding windows are extracted from the complete real and generated sequences. Windows may cross a join between two adjacent generated segments, but no window contains both real and generated values. A real window and its generated counterpart use the same start and end indices, so the two window pools have equal cardinality.

Under *Real + MarkovTS*, we first sample the real-window indices for a batch, then uniformly choose exactly half of those indices and substitute the position-matched generated windows. Batch size and the number of updates per epoch are therefore identical to *Real only*. Maximum epochs, the learning-rate schedule, validation indices, the test set, and early-stopping rules are also shared. *Online* uses the same segmentation, generated-only concatenation, and window extraction. Its only difference is that fresh Gaussian sources regenerate all generated counterparts before every epoch; the epoch then uses only these generated windows while retaining the *Real only* number of batches and updates.

D.5 CLASSIFICATION DATASETS AND EVALUATION PROTOCOL

The classification study uses the 23 datasets shared by all four selected classifiers in the fixed Multiverse archive snapshot (Middlehurst et al., 2026). Table 15 reports official train/test sizes, task metadata, and mean training lengths and scalar counts after length equalization; parentheses show original variable-length ranges. The original test split is reserved for final evaluation.

Table 15: Classification datasets used for downstream evaluation.

Dataset	Train	Test	Ch.	Mean length	Training values	Cls.	Type
ArticularyWordRecognition	275	300	9	144	356,400	25	Motion
AtrialFibrillation	15	15	2	640	19,200	3	Biosignal
CharacterTrajectories	1422	1436	3	119 (60–182)	507,654	20	Motion
Cricket	108	72	6	1,197	775,656	12	Motion
DuckDuckGeese	50	50	1345	270	18,157,500	5	Audio
ERing	30	270	4	65	7,800	6	Motion
Epilepsy	137	138	3	206	84,666	4	Motion
EthanolConcentration	261	263	3	1,751	1,371,033	4	Spectroscopy
HandMovementDirection	160	74	10	400	640,000	4	Biosignal
Handwriting	150	850	3	152	68,400	26	Motion
Heartbeat	204	205	61	405	5,039,820	2	Audio
JapaneseVowels	270	370	12	25 (7–29)	81,000	9	Audio
LSST	2459	2466	6	36	531,144	14	Sensor
Libras	180	180	2	45	16,200	15	Motion
MotorImagery	278	100	64	3,000	53,376,000	2	Biosignal
NATOPS	180	180	24	51	220,320	6	Motion
PEMS-SF	267	173	963	144	37,025,424	7	Sensor
PhonemeSpectra	3315	3353	11	217	7,912,905	39	Audio
RacketSports	151	152	6	30	27,180	4	Motion
SelfRegulationSCP1	268	293	6	896	1,440,768	2	Biosignal
SpokenArabicDigits	6599	2199	13	65 (4–93)	5,576,155	10	Audio
StandWalkJump	12	15	4	2,500	120,000	3	Biosignal
UWaveGestureLibrary	120	320	3	315	113,400	8	Motion

Selected classifiers. We use ResNet1D (Wang et al., 2017), TimesNet (Wu et al., 2023), ConvTran (Foumani et al., 2024), and Arsenal (Middlehurst et al., 2021).

Augmentation and evaluation. Each real classification sequence X_i has one generated counterpart $\widehat{X}_i$ with the same length and class label. For a reference of length $L \leq 4096$, generation uses reference-only normalization and masked right-padding to $G = \max(512, 256\lceil L/256 \rceil)$, followed by cropping to L and inverse normalization. Thus the real and generated pools have the same cardinality and class counts, without temporal rescaling.

For each neural-classifier batch, we first sample real-sequence indices and then uniformly choose exactly half of them, replacing X_i by $\widehat{X}_i$ at the selected indices. The batch size, number of updates per epoch, maximum training budget, and model-selection rule match *Real only*. Arsenal does not use mini-batches: before fitting, we uniformly select exactly half of the original training indices once and replace those real sequences by their generated counterparts. Its mixed training set therefore has the same cardinality as the *Real only* set.

The two settings reuse exactly the same real validation indices. A sequence assigned to validation and its generated counterpart are both excluded from checkpoint-selection training. ResNet1D uses validation performance to select the training duration and then refits on all original training indices; for *Real + MarkovTS*, the same fixed-cardinality 50% replacement rule is applied during this refit. TimesNet and ConvTran select checkpoints on the shared real validation subset. Arsenal fits once on all original training indices, using either all-real sequences or its fixed 50% mixed set. Table 16 summarizes the training configurations.

Metrics follow the benchmark definitions in Appendix D.3.

Table 16: Classification training configurations.

Model	Optimizer / learning rate	Batch size	Maximum epochs
ResNet1D	Adam / 10^{-3}	64	100
TimesNet	RAdam / 10^{-3}	16	30
ConvTran	RAdam / 10^{-3}	16	100
Arsenal	25 members; 2,000 kernels each	—	—

Table 17: Anomaly-detection dataset statistics for the original training segments.

Dataset	Series	Mean length	Ch.	Training values
ASD	12	8,527.6	19	1,944,289
CalIt2	1	2,520.0	2	5,040
CATSv2	4	28,471.8	17	1,936,079
CICIDS	1	85,115.0	72	6,128,280
Creditcard	1	142,403.0	29	4,129,687
DAPHNET	7	29,241.0	9	1,842,183
DLR	1	11,565.0	9	104,085
ECG	1	56,104.0	32	1,795,328
Exathlon	30	15,804.4	7–31	10,022,209
GECCO	1	69,260.0	9	623,340
GHL	25	49,347.5	19	23,440,072
GutenTAG	48	10,000.0	2–20	3,940,000
LTDB	5	12,715.2	2–3	147,152
MITDB	6	35,416.7	2	425,000
MSL	1	58,317.0	55	3,207,435
NYC	1	13,104.0	3	39,312
PSM	1	132,481.0	25	3,312,025
PUMP	1	76,901.0	44	3,383,644
SKAB	34	9,405.0	8	2,558,160
SMAP	1	135,183.0	25	3,379,575
SMD	1	708,405.0	38	26,919,390
SVDB	6	23,760.0	2	285,120
SWAN	1	60,000.0	38	2,280,000
SWAT	1	495,000.0	51	25,245,000
TAO	12	2,000.0	3	72,000
TODS	12	20,000.0	5	1,200,000

D.6 Anomaly-detection datasets and evaluation protocol

The anomaly-detection study uses 26 multivariate datasets from TAB/OpenTS (Qiu et al., 2025), comprising 215 series, listed in Table 26. Genesis, KDDcup99, and OPP are excluded from evaluation and aggregation because their official results are incomplete.

Table 17 summarizes the real training segments of these 215 series. Mean length is averaged over series within each dataset; channel ranges reflect differences between series. Training values sum scalar observations across all real training segments.

Selected detectors. The detectors include DLinear (Zeng et al., 2023), TranAD (Tuli et al., 2022), CATCH (Wu et al., 2025), ModernTCN (Luo & Wang, 2024), and TimesNet (Wu et al., 2023).

Augmentation and evaluation. Each anomaly-detection training sequence is partitioned into consecutive, non-overlapping reference segments: full segments contain exactly 4096 time steps, followed by one optional remainder segment. A remainder of length $L_s < 4096$ is masked and right-padded to $G_s = \max(512, 256\lceil L_s/256 \rceil)$, and its generated output is cropped back to L_s. We generate every segment once and concatenate only generated segments in their original order. The resulting sequence has the same length and index grid as its real counterpart; no real values are inserted. The anomaly label at each real index is copied to the generated sequence at the same index.

Stride-1 windows are extracted from the complete real and generated sequences, including across joins between adjacent generated segments. Each generated window uses the same start and end indices, and therefore the same label slice, as its real counterpart. Under *Real + MarkovTS*, we first sample the real-window indices for a batch and then uniformly choose exactly half of those indices

Table 18: Anomaly-detection training configurations. CATCH uses separate optimizers for its main and mask networks.

Model	Window	Optimizer / learning rate	Batch size	Maximum epochs
DLinear	100	Adam / 10^{-4}	64	10
TranAD	100	Adam / 10^{-4}	64	5
CATCH	96	Adam / 10^{-4}, 10^{-5}	64	3
ModernTCN	100	Adam / 10^{-3}	64	1
TimesNet	100	Adam / 10^{-4}	32	10

for replacement by their generated counterparts. The batch size, number of updates per epoch, maximum training budget, and model-selection rules match *Real only*. Generated samples are used only for training; both settings reuse the same real validation indices and real test data. Table 18 lists their shared detector configurations.

The six metrics follow Appendix D.3. Series-level scores are averaged within each dataset, followed by an equal-weight average over the 26 datasets.

E Additional Generation Analyses

E.1 Additional generation-quality evaluation

We complement the main MMD and flat-KL results with TS2Vec-based Context-Fréchet Inception Distance (Context-FID). It compares temporal embedding distributions (Appendix D.3).

Table 19: TS2Vec-based Context-FID at sequence length 168. Lower is better; bold marks the better method on each dataset.

Dataset	ImagenFew	MarkovTS
Electricity	0.5226	**0.3845**
Solar	**0.1857**	0.4350
Wind	2.3173	**0.6314**
Traffic	**0.6135**	0.6471
Taxi	0.6176	**0.4889**
Pedestrian	**0.4465**	2.2981
Air	1.0353	**0.4145**
Temperature	3.1750	**1.4524**
Rain	3.7823	**1.0151**
NN5	2.2606	**1.4191**
Fred-MD	1.2467	**1.2070**
Exchange	8.8181	**3.3709**
Wins (all 12 datasets)	3	**9**

MarkovTS achieves lower Context-FID on 9 of 12 datasets, while ImagenFew leads on Solar, Traffic, and Pedestrian. These complement Table 1's window-level and pooled-value results.

E.1.1 Fixed-noise reference-swap analysis

MMD and flat-KL compare distributions aggregated over references, so they cannot determine whether an output follows its selected reference. We therefore pair every generation with a fixed-noise intervention. For reference i, frozen deterministic state–transition condition C_i, and Gaussian source z, the correct output is $Y_i = G(z, C_i)$. A seed-specific derangement selects $j \neq i$, and the swapped output is $Y_{i \leftarrow j} = G(z, C_j)$. The pair shares z, model parameters, and reference i's affine restoration; only the complete multiscale state–transition condition changes. This controls noise and prevents affine scale from revealing the swapped reference.

The experiment uses the main flow checkpoint, 16 Euler steps, and seeds 2021–2025. Per dataset and seed, 100 length-512 training references each produce 50 correct/swapped pairs. Each output is evaluated through its three non-overlapping length-168 windows, with the final eight points ignored. GraphFormer and probability sampling remain disabled.

Table 20: Fixed-noise reference-swap analysis. Distance columns are swapped-minus-correct distances to the original reference; positive values indicate that changing only the condition moves the output away from its original owner. Retrieval targets the active condition. Values are five-seed mean $\pm$ sample standard deviation; chance is 50% for binary and 1% for 100-way retrieval.

Dataset	Δ Shape-TS2Vec	Δ ACF	Δ PSD	Binary (%)	100-way correct/swapped (%)
Electricity	0.0157 ± 0.0017	0.2299 ± 0.0195	0.2500 ± 0.0243	95.5 ± 1.1	58.9 ± 3.8 / 59.2 ± 4.3
Solar	0.0165 ± 0.0025	0.0852 ± 0.0120	0.1218 ± 0.0102	95.7 ± 1.1	51.6 ± 2.7 / 51.9 ± 2.9
Wind	0.0934 ± 0.0034	0.2341 ± 0.0109	0.2424 ± 0.0095	93.8 ± 0.8	71.2 ± 1.6 / 69.9 ± 2.1
Traffic	0.0559 ± 0.0005	0.1235 ± 0.0107	0.1595 ± 0.0116	98.8 ± 0.3	84.1 ± 0.7 / 84.2 ± 1.2
Taxi	0.0414 ± 0.0016	0.1523 ± 0.0063	0.2138 ± 0.0058	99.4 ± 0.2	89.1 ± 0.8 / 89.4 ± 0.9
Pedestrian	0.1271 ± 0.0167	0.1774 ± 0.0073	0.2178 ± 0.0066	92.0 ± 1.4	45.8 ± 1.1 / 46.7 ± 1.7
Air	0.0735 ± 0.0045	0.2368 ± 0.0128	0.2995 ± 0.0087	95.8 ± 1.3	86.6 ± 0.8 / 84.0 ± 2.6
Temperature	0.0364 ± 0.0013	0.0780 ± 0.0022	0.1482 ± 0.0029	92.3 ± 0.4	45.1 ± 2.9 / 45.0 ± 2.8
Rain	0.0404 ± 0.0037	0.0555 ± 0.0012	0.2198 ± 0.0028	96.7 ± 0.6	78.3 ± 1.1 / 78.0 ± 0.2
NN5	0.0180 ± 0.0011	0.0929 ± 0.0062	0.1133 ± 0.0038	75.9 ± 0.8	26.0 ± 0.2 / 25.6 ± 0.7
Fred-MD	0.0324 ± 0.0017	0.2325 ± 0.0205	0.1707 ± 0.0094	92.2 ± 0.3	46.3 ± 0.6 / 46.5 ± 0.6
Exchange	0.0569 ± 0.0027	0.2417 ± 0.0090	0.2361 ± 0.0109	96.0 ± 0.8	77.3 ± 1.5 / 76.6 ± 1.5
Mean	0.0506	0.1616	0.1994	93.67	63.36 / 63.09

For shape retrieval, the dataset-specific 320-dimensional TS2Vec encoders used in the Imagen-Few comparison remain frozen. Each length-168 reference and output window is independently z-standardized before TS2Vec embedding. For each representation or structural statistic, Table 20 reports the swapped-minus-correct distance to original reference i: cosine distance between TS2Vec embeddings, ACF RMS over lags 1–40, and Hellinger distance between normalized non-DC periodograms. Binary retrieval chooses the active condition from references i and j; the 100-way task searches among the 100 reference windows at the same temporal offset within their length-512 references, targeting i for the correct output and j after swapping.

All 12 datasets have positive distance changes under all three measures. Across dataset means, the changes are 0.0506 for Shape-TS2Vec, 0.1616 for ACF, and 0.1994 for PSD. Mean binary retrieval is 93.67%, compared with 50% chance. Correct and swapped 100-way retrieval are 63.36% and 63.09%, respectively, compared with 1% chance. Their near equality shows that the swap preserves approximately the same amount of identifiable condition information while redirecting it to a different reference. Correct and swapped marginal MMD remain similar because derangement preserves the empirical condition pool; these small differences are treated as a sanity check rather than conditional-fidelity evidence. Because the complete state–transition condition is swapped jointly, the intervention establishes sensitivity to the joint condition but does not separate the semantic roles of states and transitions.

Reference replication and perturbation. We compare generation with reference replication and Gaussian perturbation, neither requiring an encoder or generator. For each seed in 2021–2025, 100 length-512 references are sampled with replacement from the training pool. Replication uses the complete reference as a length-512 control output, whereas perturbation adds $\sigma \operatorname{std}(X_{\text{ref}}^{512})\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\sigma \in \{0.05, 0.1, 0.2, 0.5, 1\}$; the same noise realization is reused across strengths within each seed. As in the main protocol, every control output is split into three non-overlapping length-168 windows, its final eight points are discarded, and outputs are repeated or sampled until the first 50 windows per reference can be retained in output-then-temporal order. Thus, each control also contributes 5,000 length-168 evaluation windows per dataset and seed. Table 21 shows that MarkovTS improves over replication under both metrics on Solar, Wind, Pedestrian, Temperature, Rain, Fred-MD, and Exchange. The gains are not universal: replication retains lower MMD on Traffic, Taxi, and Air, and lower flat-KL on NN5. Among the simple controls, replication gives the lowest Traffic MMD, whereas Gaussian perturbation with $\sigma = 0.1$ gives the lowest NN5 flat-KL.

Gaussian perturbation can improve distributional matching, but its effects vary across datasets and metrics. MarkovTS retains lower MMD than every tested noise strength on Solar, Wind, Traffic, Pedestrian, Rain, NN5, Fred-MD, and Exchange, whereas noise achieves lower MMD on Taxi, Air, and Temperature. Noise achieves lower flat-KL on Traffic, Air, Temperature, NN5, and Exchange. Together, these controls show that MarkovTS can improve test-distribution matching beyond reference replication and independent Gaussian perturbation on many, but not all, datasets and metrics.

Table 21: Generation versus reference replication and Gaussian perturbation. Values are five-seed mean ± standard deviation; lower is better. Bold marks the best means at displayed precision.

Dataset	Replication	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 1$	MarkovTS
				MMD↓			
Electricity	**0.001**±0.001	**0.001**±0.001	**0.001**±0.002	**0.001**±0.001	**0.001**±0.002	0.002±0.002	**0.001**±0.001
Solar	0.035±0.005	0.037±0.005	0.038±0.006	0.040±0.006	0.049±0.009	0.193±0.011	**0.033**±0.003
Wind	0.026±0.015	0.028±0.020	0.024±0.018	0.023±0.016	0.022±0.007	0.021±0.005	**0.013**±0.009
Traffic	**0.070**±0.006	0.092±0.003	0.092±0.003	0.092±0.003	0.098±0.004	0.165±0.009	0.087±0.002
Taxi	0.094±0.011	0.099±0.000	0.099±0.000	0.098±0.000	**0.093**±0.000	0.106±0.000	0.110±0.008
Pedestrian	0.050±0.016	0.053±0.014	0.053±0.017	0.057±0.024	0.060±0.021	0.061±0.024	**0.032**±0.006
Air	**0.012**±0.003	**0.012**±0.004	0.013±0.002	0.014±0.002	0.013±0.002	0.017±0.002	0.015±0.002
Temperature	0.222±0.025	0.146±0.000	0.146±0.000	0.143±0.000	**0.128**±0.000	0.134±0.000	0.211±0.002
Rain	0.033±0.003	0.035±0.003	0.035±0.003	0.034±0.003	0.032±0.003	0.039±0.003	**0.031**±0.002
NN5	0.153±0.004	0.156±0.005	0.167±0.005	0.179±0.007	0.201±0.008	0.193±0.007	**0.145**±0.000
Fred-MD	0.003±0.002	0.003±0.002	0.003±0.002	0.003±0.002	0.003±0.001	0.003±0.001	**0.001**±0.000
Exchange	0.132±0.020	0.127±0.015	0.137±0.024	0.142±0.025	0.145±0.028	0.149±0.024	**0.124**±0.008
				flat-KL↓			
Electricity	0.106±0.048	0.110±0.000	0.102±0.000	0.100±0.000	0.098±0.000	0.084±0.000	**0.001**±0.000
Solar	0.037±0.037	0.073±0.000	0.105±0.000	0.260±0.001	0.595±0.002	0.863±0.002	**0.009**±0.002
Wind	0.223±0.063	0.224±0.065	0.219±0.060	0.201±0.047	0.177±0.024	0.151±0.008	**0.111**±0.018
Traffic	0.012±0.003	0.017±0.002	0.007±0.002	**0.004**±0.002	0.017±0.005	0.065±0.008	0.007±0.002
Taxi	0.030±0.025	0.021±0.000	0.018±0.001	0.018±0.000	0.025±0.000	0.049±0.000	**0.010**±0.004
Pedestrian	0.020±0.008	0.022±0.008	0.027±0.009	0.032±0.012	0.079±0.017	0.153±0.075	**0.007**±0.001
Air	0.029±0.005	0.029±0.005	0.027±0.004	0.029±0.005	**0.016**±0.003	0.037±0.008	0.026±0.011
Temperature	0.164±0.022	0.170±0.002	0.158±0.001	0.141±0.000	0.090±0.000	**0.071**±0.000	0.137±0.041
Rain	0.007±0.002	0.007±0.003	0.008±0.004	0.007±0.003	0.011±0.006	0.030±0.008	**0.002**±0.001
NN5	0.034±0.011	0.035±0.013	**0.033**±0.011	0.041±0.014	0.047±0.016	0.038±0.013	0.045±0.003
Fred-MD	0.517±0.259	0.346±0.142	0.340±0.135	0.308±0.124	0.243±0.008	0.187±0.004	**0.176**±0.001
Exchange	1.934±0.372	**1.766**±0.028	2.168±0.122	2.490±0.142	2.957±0.238	3.454±0.393	1.794±0.137

E.2 State–transition ablations

We examine code-assignment interventions and a single-codebook patch-based alternative. Uniform randomization and probability sampling modify the state stream, transition stream, or both while keeping the pretrained encoder, codebooks, and flow generator frozen. Patch-Flow instead uses a separately trained patch encoder with a single codebook. Table 22 summarizes the results.

Joint randomization produces higher mean MMD and flat-KL than either single-stream intervention on most datasets, including Solar, Traffic, and NN5, supporting both reference-derived streams. Effects vary: joint randomization does not yield the highest MMD on Rain or the highest flat-KL on Wind. The three probability-sampling variants yield similar distributional scores at the selected temperature. Under this paired intervention protocol, the deterministic MarkovTS control has lower MMD and flat-KL than Patch-Flow on eleven of twelve datasets.

Both code-assignment experiments use seeds 2021–2025 on all 12 datasets. Per dataset and seed, 100 length-512 references each condition 50 Gaussian sources, yielding 5,000 sequences per variant; one sampled code sequence is shared across each reference's 50 sources. Generation uses 16 Euler steps, restores reference normalization statistics, and extracts the length-168 crop $[168, 336)$ for evaluation against the target test split. Within each experiment and seed, variants share references, Gaussian sources, checkpoints, data files, and evaluation controls. The deterministic control in this paired experiment is therefore not the three-window main estimate in Table 1. Seeds vary reference selection, Gaussian sources, and code sampling; standard deviations measure cross-seed variation.

E.2.1 Uniform code randomization

Random states replaces every valid state index with an independent uniform draw from all 1,024 entries, retaining reference-derived transition indices; Random transitions does the converse, and Random both replaces both streams. Draws may coincide with original indices; invalid positions and codebook vectors remain unchanged. Random both shares its randomized state and transition streams with the respective single-stream variants. All interventions follow deterministic encoding, without recomputing transitions after state replacement.

Table 22: Generation quality under code interventions and patch-based comparison. Values are five-seed mean ± sample standard deviation. Lower is better.

			Random			Stochastic			
Dataset	Metric	MarkovTS	States	Transitions	Both	States	Transitions	Both	Patch-Flow
Elec.	MMD	0.001±0.000	0.002±0.002	0.002±0.002	0.003±0.002	0.001±0.001	0.002±0.002	0.002±0.002	0.003±0.002
	flat-KL	0.001±0.000	0.059±0.044	0.061±0.041	0.068±0.039	0.005±0.003	0.012±0.008	0.015±0.013	0.054±0.031
Solar	MMD	0.032±0.003	0.236±0.008	0.190±0.011	0.516±0.008	0.034±0.004	0.034±0.003	0.034±0.003	0.045±0.006
	flat-KL	0.009±0.000	0.689±0.036	0.842±0.027	1.099±0.046	0.015±0.008	0.014±0.007	0.020±0.010	0.173±0.018
Wind	MMD	0.019±0.001	0.023±0.026	0.023±0.025	0.023±0.026	0.023±0.026	0.023±0.026	0.023±0.026	0.033±0.030
	flat-KL	0.147±0.006	0.189±0.061	0.165±0.051	0.178±0.056	0.177±0.059	0.177±0.057	0.177±0.057	0.203±0.068
Traffic	MMD	0.087±0.002	0.168±0.010	0.130±0.008	0.302±0.009	0.080±0.007	0.080±0.007	0.080±0.007	0.084±0.003
	flat-KL	0.003±0.000	0.144±0.016	0.131±0.040	0.247±0.085	0.007±0.004	0.007±0.004	0.007±0.004	0.003±0.001
Taxi	MMD	0.101±0.001	0.118±0.007	0.119±0.007	0.148±0.007	0.091±0.011	0.091±0.012	0.091±0.011	0.105±0.003
	flat-KL	0.002±0.000	0.075±0.029	0.146±0.055	0.184±0.053	0.016±0.011	0.017±0.012	0.016±0.011	0.012±0.004
Ped.	MMD	0.032±0.001	0.049±0.003	0.054±0.006	0.082±0.003	0.034±0.003	0.035±0.004	0.037±0.010	0.057±0.007
	flat-KL	0.007±0.001	0.088±0.017	0.148±0.017	0.192±0.011	0.010±0.004	0.015±0.006	0.020±0.007	0.163±0.015
Air	MMD	0.009±0.001	0.016±0.002	0.017±0.004	0.022±0.005	0.013±0.002	0.013±0.002	0.013±0.002	0.011±0.002
	flat-KL	0.019±0.002	0.039±0.020	0.046±0.040	0.057±0.054	0.030±0.013	0.030±0.013	0.030±0.013	0.021±0.002
Temp.	MMD	0.142±0.001	0.134±0.006	0.152±0.010	0.213±0.018	0.134±0.007	0.134±0.007	0.134±0.007	0.204±0.015
	flat-KL	0.161±0.004	0.134±0.024	0.109±0.021	0.253±0.045	0.098±0.016	0.098±0.017	0.098±0.018	0.224±0.039
Rain	MMD	0.031±0.001	0.046±0.002	0.053±0.002	0.051±0.004	0.036±0.002	0.036±0.002	0.036±0.002	0.040±0.003
	flat-KL	0.007±0.000	0.010±0.002	0.015±0.004	0.050±0.031	0.004±0.003	0.004±0.003	0.004±0.003	0.008±0.002
NN5	MMD	0.180±0.002	0.205±0.008	0.231±0.007	0.292±0.010	0.166±0.024	0.186±0.007	0.186±0.007	0.236±0.007
	flat-KL	0.043±0.001	0.106±0.036	0.193±0.050	0.301±0.084	0.047±0.004	0.051±0.013	0.051±0.013	0.215±0.043
Fred-MD	MMD	0.002±0.000	0.003±0.001	0.002±0.001	0.002±0.001	0.003±0.001	0.003±0.001	0.003±0.001	0.004±0.002
	flat-KL	0.244±0.006	0.364±0.258	0.200±0.026	0.236±0.025	0.256±0.018	0.291±0.106	0.288±0.105	0.369±0.139
Exchange	MMD	0.124±0.003	0.148±0.020	0.147±0.021	0.148±0.022	0.138±0.012	0.142±0.010	0.143±0.017	0.142±0.012
	flat-KL	1.631±0.026	5.030±2.159	3.068±0.668	3.351±0.374	1.760±0.108	1.842±0.189	1.883±0.203	2.013±0.091

E.2.2 Reference-dependent code sampling

Probability sampling draws from all 1,024 entries using the frozen query's cosine energies E_k:

$$p(k \mid q) = \frac{\exp(-E_k(q)/T)}{\sum_{\ell=1}^{1024} \exp(-E_\ell(q)/T)}, \qquad T = 0.005. \tag{47}$$

Stochastic states, Stochastic transitions, and Stochastic both sample every valid position in the corresponding streams, without top-k truncation. Unsampled streams use nearest-code assignment. State assignments are obtained first; their selected vectors are then used to recompute transition queries and energies. Thus, Stochastic states can change transition indices even when transitions use nearest-code assignment, preserving the encoder's state-dependent transition construction rather than fixing the other stream as in uniform randomization.

E.2.3 Single-codebook patch representation

Patch-Flow uses a standard patch Transformer with a single codebook to condition a flow-matching generator. The patch representation is trained using codebook optimization and reconstruction losses, with the remaining model and training configurations matching MarkovTS.

E.3 Generation across sequence lengths

The length analysis uses the same pretrained MarkovTS encoder, codebooks, flow model, and metric implementations, with $N_{\mathrm{FE}} = 16$ Euler steps. For each dataset and seed, $K_{\mathrm{eval}} = 100$ references each contribute 50 retained evaluation windows, giving $N = 5000$ windows in total. We use seeds 2021–2025.

Reference and generation lengths match, $L_{\mathrm{gen}} = L_{\mathrm{ref}} \in \{512, 1024, 2048\}$. Each output yields 3, 6, or 12 fixed, non-overlapping $L_{\mathrm{eval}} = 168$ windows, discarding the final 8, 16, or 32 points, respectively. For each reference, we generate outputs until at least 50 windows are available, order

candidates first by Gaussian draw and then by temporal position, and retain the first 50. The comparison therefore measures length-168 window fidelity under references and outputs of different lengths, not full-sequence fidelity at lengths 512, 1024, or 2048.

Fred-MD, NN5, Rain, and Temperature contain 504 released training points per channel. Lengths 512, 1024, and 2048 therefore require 8, 520, and 1544 reflection-padded points, respectively.

Table 23 compares reference and generation lengths 512, 1024, and 2048 over five seeds; † marks datasets requiring reflection padding.

Table 23: Generation quality across reference and output lengths. Lower is better; values are five-seed mean ± sample standard deviation. All outputs are evaluated through fixed, non-overlapping length-168 windows.

Metric	Dataset	512	1024	2048
MMD	Electricity	0.001 ± 0.001	0.002 ± 0.001	0.002 ± 0.001
	Solar	0.033 ± 0.003	0.038 ± 0.005	0.036 ± 0.007
	Wind	0.013 ± 0.009	0.017 ± 0.013	0.021 ± 0.017
	Traffic	0.087 ± 0.002	0.222 ± 0.022	0.194 ± 0.011
	Taxi	0.110 ± 0.008	0.102 ± 0.003	0.098 ± 0.001
	Pedestrian	0.032 ± 0.006	0.037 ± 0.005	0.036 ± 0.006
	Air	0.015 ± 0.002	0.014 ± 0.003	0.012 ± 0.004
	Temperature†	0.211 ± 0.002	0.183 ± 0.007	0.202 ± 0.015
	Rain†	0.031 ± 0.002	0.029 ± 0.002	0.027 ± 0.002
	NN5†	0.145 ± 0.000	0.168 ± 0.001	0.215 ± 0.000
	Fred-MD†	0.001 ± 0.000	0.001 ± 0.000	0.001 ± 0.000
	Exchange	0.124 ± 0.008	0.130 ± 0.008	0.135 ± 0.004
flat-KL	Electricity	0.001 ± 0.000	0.064 ± 0.044	0.061 ± 0.046
	Solar	0.009 ± 0.002	0.012 ± 0.003	0.013 ± 0.002
	Wind	0.111 ± 0.018	0.103 ± 0.035	0.114 ± 0.055
	Traffic	0.007 ± 0.002	0.007 ± 0.002	0.005 ± 0.002
	Taxi	0.010 ± 0.004	0.009 ± 0.005	0.007 ± 0.003
	Pedestrian	0.007 ± 0.001	0.018 ± 0.011	0.016 ± 0.009
	Air	0.026 ± 0.011	0.022 ± 0.004	0.017 ± 0.007
	Temperature†	0.137 ± 0.041	0.139 ± 0.041	0.141 ± 0.042
	Rain†	0.002 ± 0.001	0.002 ± 0.001	0.002 ± 0.001
	NN5†	0.045 ± 0.003	0.044 ± 0.002	0.051 ± 0.002
	Fred-MD†	0.176 ± 0.001	0.175 ± 0.002	0.175 ± 0.003
	Exchange	1.794 ± 0.137	1.855 ± 0.144	1.786 ± 0.054

F Qualitative and Structural Generation Analyses

We complement generation metrics with analyses of the learned representation and frozen sampler. Representation analyses use real sequences and deterministic nearest-code assignments. Generation examples use exponential-moving-average flow parameters and independent Gaussian sources unless specified otherwise, showing variation under a fixed reference-derived condition.

F.1 State and transition code structure

F.1.1 Shared codebook entries

Figure 5 illustrates shared local morphology and contextual transitions across held-out datasets. State examples show locally normalized patches assigned to frequent codes; transition examples show previous–current patch pairs sharing a transition code. Deterministic selection prioritizes usage and representation across Electricity, Solar, Traffic, and Exchange. Recurring shapes and ordered changes suggest shared temporal structure across these datasets.

F.1.2 State–transition complementarity

Shared morphological states need not specify changes between their continuous patches. Using real length-512 training references and the frozen encoder, we group transition occurrences by dataset d, temporal span w_h, and ordered state pair. Each reference is standardized once by its mean and

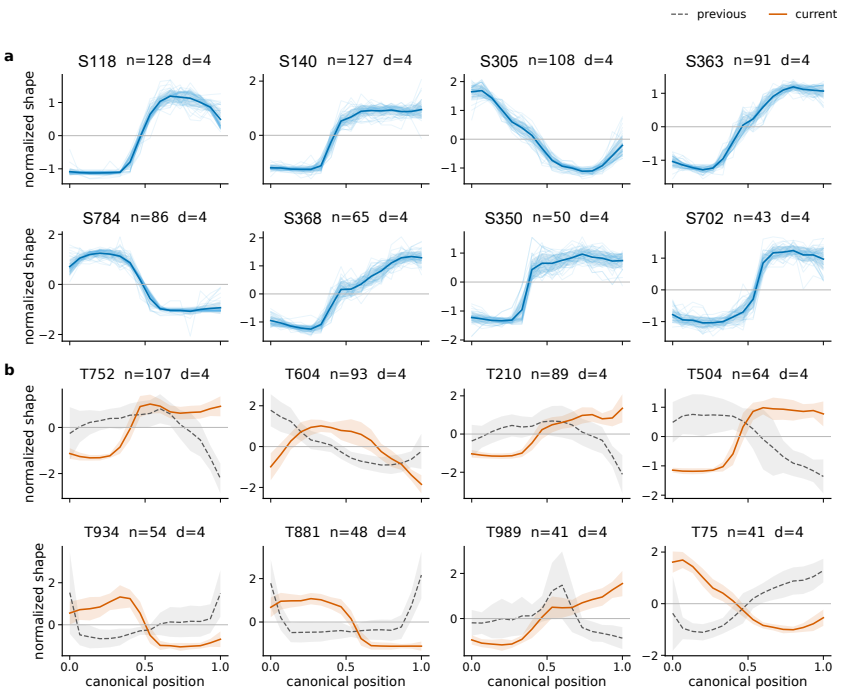


Figure 5: Representative state and transition codes at temporal span 16. **(a)** Assigned normalized patches and their mean. **(b)** Mean previous/current patches and one-standard-deviation bands. n and d denote the number of assignments and represented datasets, respectively.

population standard deviation, with denominator stabilizer 10^{-8}. Adjacent intervals are reduced to canonical patches $p, q \in \mathbb{R}^P$, $P = 16$, without further patchwise normalization, retaining relative levels and amplitudes.

Let $m(a)$ and $s(a)$ denote the mean and population standard deviation of a canonical patch a, and let $u_b = b - (P-1)/2$. The least-squares slope $\beta(a)$ and continuous-relation feature $\phi(p, q)$ are

$$\begin{aligned} \beta(a) &= \frac{\sum_{b=0}^{P-1} u_b a_b}{\sum_{b=0}^{P-1} u_b^2}, \\ \phi(p,q) &= \big[m(q) - m(p),\ \log\max(s(q), 10^{-6}) - \log\max(s(p), 10^{-6}), \\ &\qquad \beta(q) - \beta(p),\ q_0 - p_{P-1}\big]^{\mathsf{T}}. \end{aligned} \tag{48}$$

The four coordinates measure level shift, log-amplitude ratio, slope change, and boundary jump. Before group filtering or subsampling, each coordinate is centered and divided by its population standard deviation over all valid transition occurrences in the same dataset–span cell; standard deviations below 10^{-8} are replaced by one. Occurrence j has standardized feature $\widetilde{\phi}_j$.

A state-pair group is eligible when at least two transition codes each occur at least three times. We retain at most 100 eligible groups per dataset–span cell, sampled uniformly without replacement; within each group, we retain up to eight codes ranked by decreasing frequency, breaking ties by increasing code ID, and up to 64 occurrences per code, sampled uniformly without replacement. Sampling uses base seed 20260922. For a retained group g, let $I_{g,k}$ contain the retained occurrences of transition code k, $n_{g,k} = |I_{g,k}|$, and $\bar{\phi}_{g,k}$ be their mean standardized feature. Pooled within-code dispersion is

$$W_g^{\mathrm{obs}} = \frac{\sum_k \sum_{j \in I_{g,k}} \|\widetilde{\phi}_j - \bar{\phi}_{g,k}\|_2^2}{\sum_k (n_{g,k} - 1)}. \tag{49}$$

Thus, each code contributes according to its residual degrees of freedom. Within each group, we permute transition labels 200 times while preserving the occurrence set and every code count. Writing

$W_g^{\pi_b}$ for the dispersion under permutation b, the observed-to-permuted ratio is

$$R_g = \frac{W_g^{\mathrm{obs}}}{\max\left(\frac{1}{200}\sum_{b=1}^{200} W_g^{\pi_b}, 10^{-12}\right)}. \tag{50}$$

Ratios below one indicate greater within-code coherence than count-matched random labels.

Aggregation gives each eligible dataset equal weight. We take median ratios over retained groups $G_{d,h}$ per dataset–span cell, then over datasets with at least five groups:

$$M_{d,h} = \mathrm{median}_{g \in G_{d,h}} R_g, \qquad M_h = \mathrm{median}_{d:\,|G_{d,h}| \geq 5} M_{d,h}. \tag{51}$$

Figure 4(c) reports the reduction $100(1 - M_h)\%$. At each span, we resample the eligible dataset medians with replacement 5,000 times, retaining the original number of datasets, and recompute the macro-median. The 2.5th and 97.5th percentiles of the resulting reductions define the 95% interval. Spans $256, 128, 64, 32, 16$ include $8, 8, 10, 10, 10$ datasets and $547, 710, 700, 772, 860$ groups, respectively, totaling 3,589 groups in the macro statistics. The reductions of 18.1%–29.7%, with all intervals excluding zero, support transition-code information beyond the morphological state pair.

The examples in Figures 6 and 7 make this complementarity visible across the 12 datasets. Within a fixed state pair, the Solar examples differ in amplitude change, while the Wind examples distinguish upward and downward level shifts. Each displayed sequence is a feature-space medoid for one transition code, with normalization applied once to the complete length-512 reference so that between-patch level and amplitude differences remain visible. The preceding context is shown in gray, followed by the previous patch in blue and the current patch in orange.

Examples are selected by a support-filtered centroid-diversity rule requiring three transition codes with at least five occurrences each, independently of the aggregate analysis. For Fred-MD, no group meets this threshold; the marked example uses state pair $934 \to 934$ at scale 64, with code supports of 7, 3, and 5. This exception covers all datasets qualitatively without altering quantitative results.

F.2 Multiscale conditioning

Figure 8 removes coarse conditions while holding the reference, Gaussian source, generator, and integration settings fixed. In the three examples, removing coarse states and transitions alters global organization, while the remaining fine-scale codes retain local waveform detail.

F.3 Distributional diagnostics

Figure 9 compares generated and real validation sequences through marginal value densities, autocorrelation, and power spectra, assessing amplitude statistics, temporal dependence, and periodic content. Using up to 3,000 finite-valued sequences per distribution, these diagnostics complement Table 1.

F.4 Conditional flow path

Figure 10 shows intermediate Euler states under a fixed state–transition condition. The flow gradually suppresses unstructured Gaussian variation while establishing the reference's periodicity, harmonic content, trend, and smooth level shift.

F.5 Conditional sample stability

For Figure 3, we draw 100 length-512 references with replacement from eligible training windows in each dataset and generate 50 samples per reference draw using fixed discrete conditions and independent Gaussian sources. Statistics use the crop $[168, 336)$, with the reference and generated samples standardized using the complete length-512 reference. Pointwise empirical quantiles define the median and central 50% and 95% intervals; inclusion rate is the fraction of reference values within the 95% interval. We display the reference draw whose inclusion rate is closest to the dataset's 10th percentile, resolving ties by draw order, to illustrate a lower-coverage case.

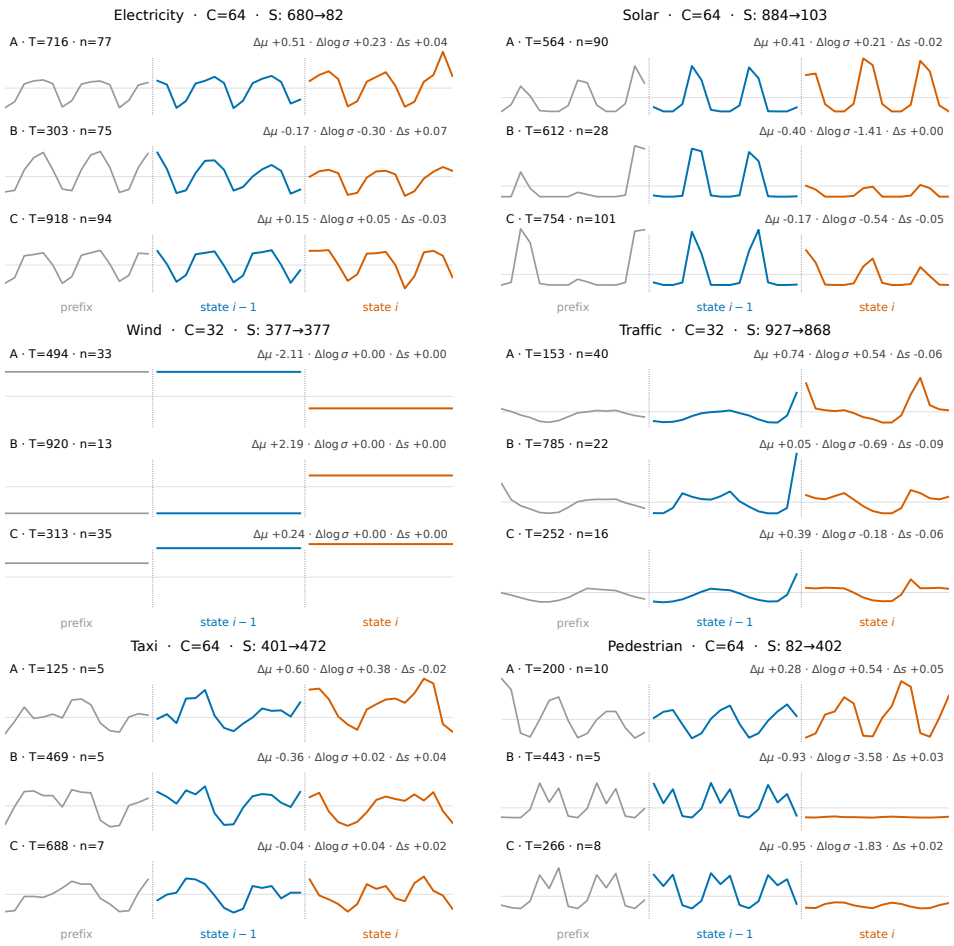


Figure 6: Transition-code examples within fixed state pairs on Electricity, Solar, Wind, Traffic, Taxi, and Pedestrian. Each panel shows three real patch-pair medoids. C denotes temporal scale, S the ordered state pair, T the transition code, and n its occurrence count. Annotations report changes in level, log-amplitude, and slope.

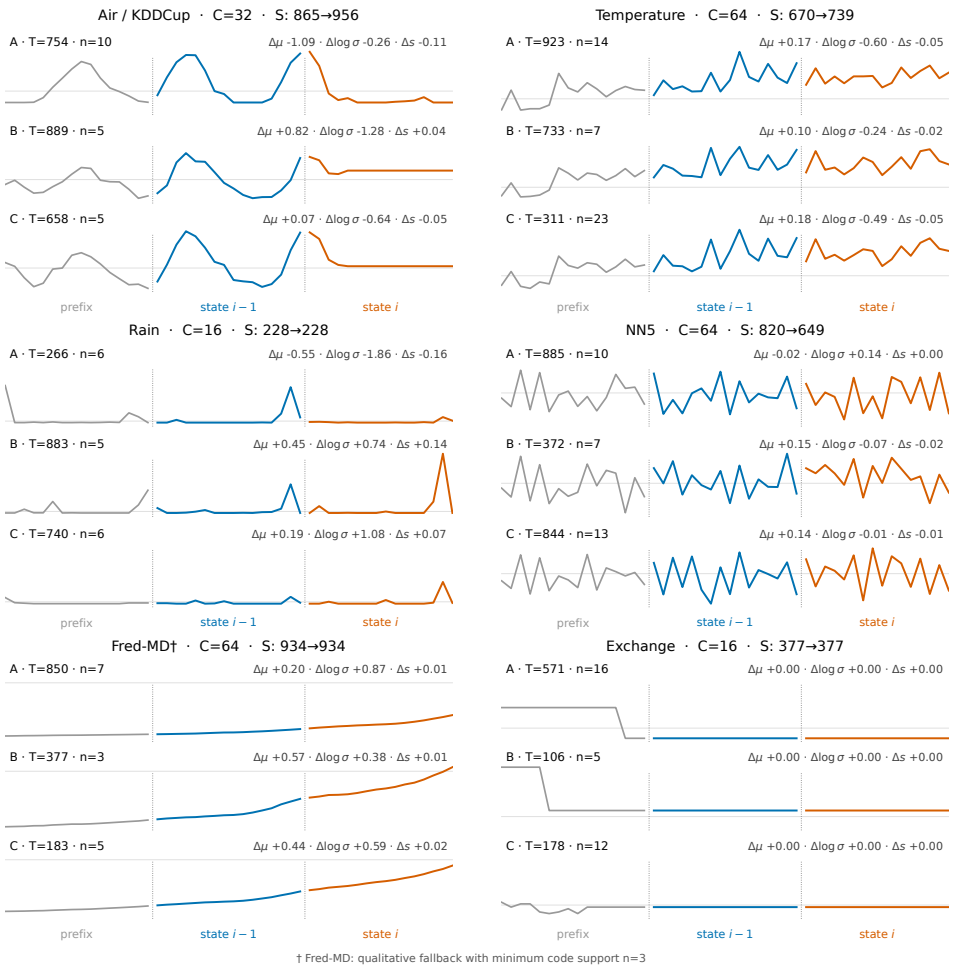


Figure 7: Transition-code examples within fixed state pairs on Air, Temperature, Rain, NN5, Fred-MD, and Exchange, using the notation of Figure 6. The dagger marks Fred-MD's lower-support qualitative example.

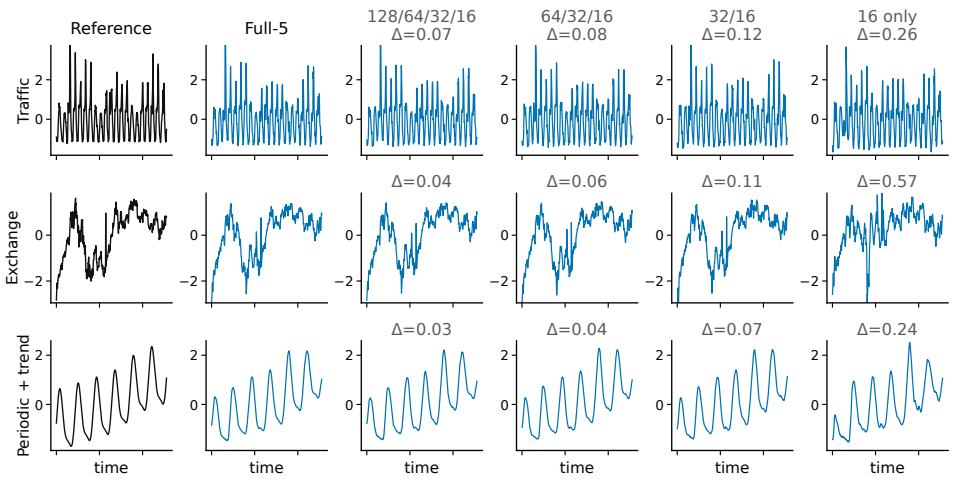


Figure 8: Coarse-to-fine conditioning ablation with a shared Gaussian source. Columns progressively remove the coarsest condition; Δ is the standardized RMS distance from the full five-scale output.

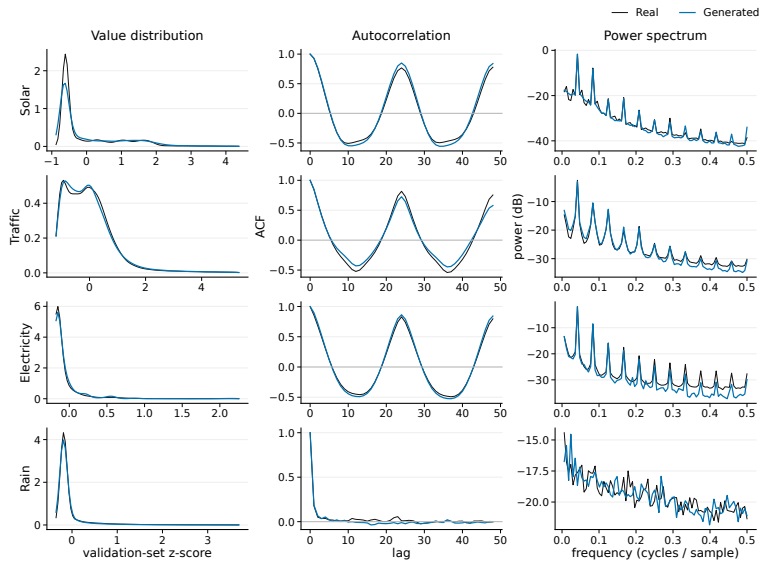


Figure 9: Distributional diagnostics on four held-out datasets. Columns compare value density, autocorrelation, and normalized power spectrum for real validation data (black) and MarkovTS samples (blue).

G Horizon-Wise Forecasting Results

Table 24 compares *Real only*, *Real + MarkovTS* using a fixed generated counterpart pool and 50/50 counterpart-replacement sampling, and online generated-data replacement (*Online*) across forecasting horizons. Training protocols are detailed in Appendix D.4. Bold Imp. rows report relative MAE and MSE reductions (%) from *Real only*.

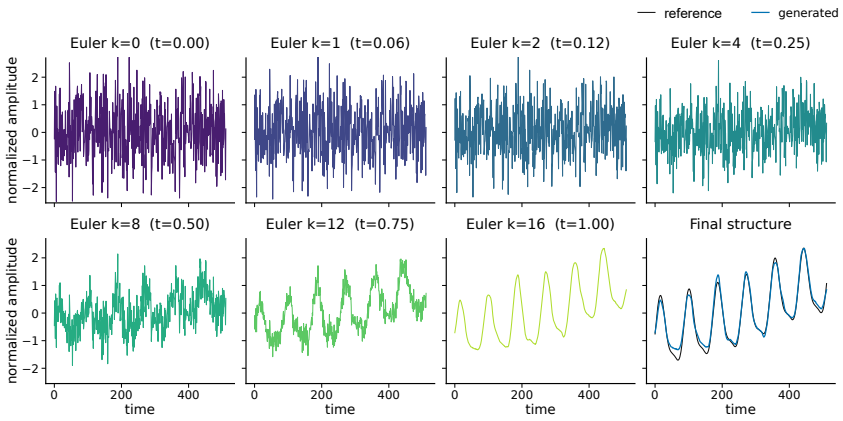


Figure 10: Intermediate states along a conditional flow path. The final panel compares the reference with the generated realization.

Table 24: Forecasting MAE and MSE by horizon (lower is better). R: Real only; R+M: Real + MarkovTS; O: Online. Imp.: relative error reduction (%) from R.

Dataset	H	CATS R	CATS R+M	CATS O	iTransformer R	iTransformer R+M	iTransformer O	SparseTSF R	SparseTSF R+M	SparseTSF O	TQNet R	TQNet R+M	TQNet O	OLinear R	OLinear R+M	OLinear O
							MAE ↓									
ECL	96	.232	.217	.223	.230	.233	.235	.244	.237	.238	.220	.218	.221	.274	.273	.274
	192	.250	.233	.239	.252	.248	.252	.252	.248	.249	.239	.237	.240	.287	.288	.289
	336	.260	.250	.256	.264	.264	.268	.266	.264	.264	.248	.252	.256	.301	.303	.304
	720	.291	.281	.290	.289	.299	.302	.297	.295	.296	.277	.280	.287	.329	.323	.338
	Mean	.258	.245	.252	.259	.261	.264	.265	.261	.262	.246	.247	.251	.298	.297	.301
	Imp.		**5.0**	**2.3**		**-0.8**	**-1.9**		**1.5**	**1.1**		**-0.4**	**-2.0**		**0.3**	**-1.0**
ETTh1	96	.411	.387	.387	.420	.395	.397	.391	.388	.390	.388	.389	.392	.395	.392	.394
	192	.432	.413	.410	.463	.420	.422	.414	.410	.414	.414	.414	.419	.417	.418	.420
	336	.447	.432	.430	.462	.439	.441	.440	.425	.427	.432	.436	.438	.437	.436	.435
	720	.483	.462	.458	.546	.479	.475	.471	.443	.445	.466	.471	.468	.463	.470	.477
	Mean	.443	.423	.421	.473	.433	.434	.429	.417	.419	.425	.428	.429	.428	.429	.432
	Imp.		**4.5**	**5.0**		**8.5**	**8.2**		**2.8**	**2.3**		**-0.7**	**-0.9**		**-0.2**	**-0.9**
ETTh2	96	.368	.331	.330	.382	.365	.362	.370	.355	.357	.358	.357	.360	.358	.355	.356
	192	.417	.382	.378	.440	.414	.414	.413	.406	.409	.408	.408	.406	.406	.405	.402
	336	.443	.400	.403	.451	.446	.444	.438	.436	.443	.437	.436	.429	.433	.429	.428
	720	.472	.422	.425	.484	.481	.483	.486	.460	.462	.467	.463	.457	.468	.460	.460
	Mean	.425	.384	.384	.439	.427	.426	.427	.414	.418	.417	.416	.413	.416	.412	.412
	Imp.		**9.6**	**9.6**		**2.7**	**3.0**		**3.0**	**2.1**		**0.2**	**1.0**		**1.0**	**1.0**
Exchange	96	.215	.208	.205	.235	.211	.215	.235	.202	.206	.209	.205	.207	.219	.213	.216
	192	.306	.310	.304	.330	.315	.323	.344	.295	.298	.338	.328	.304	.321	.317	.313
	336	.431	.429	.435	.493	.443	.447	.499	.417	.420	.472	.436	.434	.467	.448	.431
	720	.730	.739	.752	.852	.821	.838	.844	.783	.827	.795	.739	.787	.817	.777	.831
	Mean	.420	.421	.424	.478	.447	.456	.481	.425	.438	.454	.427	.433	.456	.439	.448
	Imp.		**-0.2**	**-1.0**		**6.5**	**4.6**		**11.6**	**8.9**		**5.9**	**4.6**		**3.7**	**1.8**
Traffic	96	.263	.233	.240	.270	.281	.281	.285	.270	.272	.249	.243	.254	.316	.308	.313
	192	.283	.242	.249	.281	.292	.291	.290	.273	.274	.261	.254	.262	.323	.317	.320
	336	.287	.248	.255	.291	.299	.298	.296	.278	.279	.267	.261	.269	.329	.322	.325
	720	.305	.283	.282	.311	.315	.314	.311	.294	.294	.284	.279	.285	.343	.335	.339
	Mean	.284	.252	.257	.289	.297	.296	.295	.279	.280	.265	.259	.268	.328	.321	.324
	Imp.		**11.6**	**9.8**		**-2.9**	**-2.6**		**5.7**	**5.3**		**2.2**	**-0.9**		**2.2**	**1.1**
Weather	96	.193	.182	.184	.206	.194	.192	.235	.217	.217	.182	.183	.185	.188	.188	.190
	192	.233	.225	.228	.245	.235	.235	.265	.254	.254	.227	.229	.228	.227	.230	.230
	336	.274	.266	.271	.284	.276	.276	.294	.291	.291	.265	.271	.273	.264	.271	.272
	720	.331	.323	.324	.333	.328	.328	.339	.341	.340	.320	.324	.329	.320	.326	.330
	Mean	.258	.249	.252	.267	.258	.258	.283	.276	.276	.249	.252	.254	.250	.254	.255
	Imp.		**3.5**	**2.3**		**3.4**	**3.4**		**2.5**	**2.5**		**-1.2**	**-2.0**		**-1.6**	**-2.0**
							MSE ↓									
ECL	96	.137	.130	.134	.135	.138	.140	.150	.144	.144	.130	.127	.130	.159	.157	.158
	192	.154	.149	.151	.160	.155	.158	.159	.157	.157	.149	.148	.149	.175	.176	.176
	336	.165	.165	.169	.169	.171	.175	.172	.173	.173	.158	.162	.166	.190	.193	.194
	720	.203	.201	.211	.194	.212	.214	.212	.211	.211	.193	.195	.204	.231	.219	.240
	Mean	.165	.161	.166	.164	.169	.172	.173	.171	.171	.158	.158	.162	.188	.186	.192
	Imp.		**2.4**	**-0.6**		**-3.0**	**-4.9**		**1.2**	**1.2**		**0.0**	**-2.5**		**1.1**	**-2.1**
ETTh1	96	.396	.376	.378	.410	.373	.373	.374	.371	.373	.368	.370	.371	.369	.366	.365
	192	.437	.418	.411	.480	.414	.412	.414	.410	.412	.413	.415	.417	.409	.410	.411
	336	.456	.444	.439	.471	.435	.436	.445	.438	.437	.444	.453	.453	.436	.435	.431
	720	.503	.464	.457	.581	.460	.458	.473	.428	.430	.462	.477	.469	.442	.451	.463
	Mean	.448	.425	.421	.485	.421	.420	.427	.412	.413	.422	.429	.428	.414	.415	.417
	Imp.		**5.1**	**6.0**		**13.2**	**13.4**		**3.5**	**3.3**		**-1.7**	**-1.4**		**-0.2**	**-0.7**

Table 24: Horizon-wise forecasting results (continued).

Dataset	H	CATS R	CATS R+M	CATS O	iTransformer R	iTransformer R+M	iTransformer O	SparseTSF R	SparseTSF R+M	SparseTSF O	TQNet R	TQNet R+M	TQNet O	OLinear R	OLinear R+M	OLinear O
ETTh2	96	.316	.284	.283	.334	.319	.317	.323	.309	.314	.311	.313	.317	.305	.304	.304
	192	.384	.364	.350	.431	.395	.395	.385	.384	.396	.388	.387	.380	.380	.379	.375
	336	.415	.376	.388	.440	.434	.430	.413	.421	.436	.419	.421	.411	.415	.412	.404
	720	.455	.397	.400	.480	.488	.475	.477	.444	.447	.447	.452	.443	.456	.449	.449
	Mean	.392	.355	.355	.421	.409	.404	.399	.389	.398	.391	.393	.388	.389	.386	.383
	Imp.		**9.4**	**9.4**		**2.9**	**4.0**		**2.5**	**0.3**		**-0.5**	**0.8**		**0.8**	**1.5**
Exchange	96	.094	.084	.083	.103	.084	.088	.105	.081	.083	.086	.081	.083	.092	.087	.090
	192	.185	.183	.176	.207	.182	.194	.221	.166	.169	.215	.202	.179	.192	.187	.183
	336	.353	.350	.355	.423	.359	.358	.434	.325	.331	.411	.360	.356	.398	.366	.340
	720	.921	.932	.968	1.124	1.051	1.085	1.155	.988	1.062	1.001	.903	1.025	1.059	.971	1.091
	Mean	.388	.387	.395	.464	.419	.431	.478	.390	.411	.428	.387	.411	.435	.402	.426
	Imp.		**0.3**	**-1.8**		**9.7**	**7.1**		**18.4**	**14.0**		**9.6**	**4.0**		**7.6**	**2.1**
Traffic	96	.382	.368	.373	.375	.401	.399	.424	.438	.434	.377	.365	.379	.436	.421	.430
	192	.418	.389	.393	.403	.420	.418	.437	.448	.444	.403	.388	.397	.453	.443	.446
	336	.431	.402	.406	.420	.434	.433	.448	.460	.457	.417	.403	.412	.468	.455	.461
	720	.454	.450	.445	.449	.463	.462	.470	.484	.479	.442	.434	.437	.502	.492	.496
	Mean	.421	.402	.404	.412	.429	.428	.445	.458	.453	.410	.398	.406	.465	.453	.458
	Imp.		**4.5**	**4.0**		**-4.3**	**-3.9**		**-2.9**	**-1.9**		**3.0**	**0.8**		**2.5**	**1.3**
Weather	96	.145	.151	.152	.158	.153	.153	.185	.183	.183	.144	.144	.145	.145	.146	.145
	192	.184	.190	.193	.200	.197	.198	.224	.224	.225	.188	.188	.186	.186	.187	.186
	336	.236	.238	.242	.251	.249	.249	.267	.272	.272	.237	.241	.243	.233	.238	.238
	720	.316	.321	.323	.323	.322	.319	.333	.341	.339	.317	.318	.324	.318	.321	.325
	Mean	.221	.225	.228	.233	.230	.230	.252	.255	.255	.222	.223	.225	.221	.223	.224
	Imp.		**-1.8**	**-3.2**		**1.3**	**1.3**		**-1.2**	**-1.2**		**-0.5**	**-1.4**		**-0.9**	**-1.4**

H Per-Dataset Classification-Augmentation Results

Table 25 reports *Real only*/*Real + MarkovTS* for every dataset–classifier pair. All metrics except log loss are percentages and higher-is-better; lower log loss is better. Appendices D.5 and D.3 define training and metrics.

Table 25: Per-dataset classification: Real only/Real + MarkovTS. Arrows indicate better performance. For multiclass datasets, = Acc. denotes Sens. and Spec. equal to Acc. under the benchmark convention.

Dataset	Acc.↑	BAcc.↑	F1↑	AUROC↑	LL↓	Sens.↑	Spec.↑
ResNet1D							
ArticularyWordRecognition	95.78/88.78	95.78/88.78	95.78/87.48	99.94/99.71	0.16/0.42	= Acc.	
AtrialFibrillation	11.11/13.33	11.11/13.33	8.85/10.73	37.56/40.22	4.18/4.13	= Acc.	
CharacterTrajectories	98.84/99.28	98.75/99.21	98.84/99.28	99.98/99.99	0.05/0.03	= Acc.	
Cricket	98.15/98.15	98.15/98.15	98.14/98.14	99.97/99.99	0.06/0.04	= Acc.	
DuckDuckGeese	56.67/55.33	56.67/55.33	55.66/56.01	82.37/81.68	1.62/2.17	= Acc.	
ERing	85.68/86.54	85.68/86.54	85.50/86.36	98.78/98.95	0.40/0.36	= Acc.	
Epilepsy	95.65/96.14	95.81/96.22	95.67/96.12	99.13/99.53	0.20/0.13	= Acc.	
EthanolConcentration	26.74/25.48	26.72/25.48	25.55/23.80	53.90/53.04	1.69/1.93	= Acc.	
HandMovementDirection	35.13/34.69	36.15/36.83	32.78/33.00	61.40/63.20	1.49/1.60	= Acc.	
Handwriting	52.59/55.41	52.15/55.10	49.54/52.43	93.31/93.16	1.76/1.67	= Acc.	
Heartbeat	70.57/62.44	66.85/70.03	50.85/56.35	76.23/76.58	0.59/0.88	58.48/87.14	75.22/52.93
JapaneseVowels	98.29/96.85	98.23/96.91	98.29/96.87	99.98/99.92	0.06/0.10	= Acc.	
LSST	42.92/60.16	39.87/48.02	43.51/58.84	80.08/88.31	3.32/1.90	= Acc.	
Libras	92.78/91.85	92.78/91.85	92.69/91.82	99.63/99.65	0.26/0.33	= Acc.	
MotorImagery	49.33/49.00	49.33/49.00	34.91/38.57	49.11/47.84	1.87/1.27	27.33/32.67	71.33/65.33
NATOPS	94.63/94.81	94.63/94.81	94.46/94.76	99.77/99.64	0.17/0.14	= Acc.	
PEMS-SF	70.14/77.65	70.70/77.38	69.46/77.48	94.70/96.64	0.99/0.73	= Acc.	
PhonemeSpectra	29.14/28.53	29.14/28.54	28.27/27.98	89.02/88.40	3.30/3.78	= Acc.	
RacketSports	78.73/88.82	79.64/89.69	78.52/88.83	94.31/97.65	1.06/0.34	= Acc.	
SelfRegulationSCP1	72.47/72.01	72.42/72.02	56.14/68.22	67.51/87.80	0.98/0.90	57.53/72.60	87.30/71.43
SpokenArabicDigits	98.51/98.85	98.52/98.85	98.51/98.85	99.98/100.00	0.05/0.03	= Acc.	
StandWalkJump	31.11/28.89	31.11/28.89	22.20/19.54	51.56/47.11	1.13/1.17	= Acc.	
UWaveGestureLibrary	82.50/82.71	82.50/82.71	82.11/82.26	97.09/97.12	0.58/0.59	= Acc.	
TimesNet							
ArticularyWordRecognition	94.67/97.33	94.67/97.33	94.47/97.34	99.68/99.93	0.85/0.18	= Acc.	
AtrialFibrillation	26.67/40.00	26.67/40.00	21.67/38.89	34.67/39.33	1.20/1.19	= Acc.	
CharacterTrajectories	98.12/97.56	97.98/97.37	98.12/97.56	99.90/99.96	0.09/0.09	= Acc.	
Cricket	94.44/93.06	94.44/93.06	94.36/92.17	99.96/99.94	0.19/0.25	= Acc.	
DuckDuckGeese	42.00/52.00	42.00/52.00	41.05/48.69	72.05/80.20	1.49/1.35	= Acc.	
ERing	75.93/90.74	75.93/90.74	74.04/90.79	96.86/99.03	1.18/0.60	= Acc.	
Epilepsy	89.86/90.58	89.83/90.62	89.25/90.24	98.63/98.80	0.28/0.29	= Acc.	
EthanolConcentration	29.66/25.48	29.55/25.39	19.86/22.66	57.46/55.09	1.84/1.78	= Acc.	
HandMovementDirection	56.76/60.81	58.93/62.14	56.19/59.41	79.85/79.96	1.15/1.16	= Acc.	
Handwriting	17.41/22.71	16.67/22.04	13.30/19.81	68.58/71.25	3.06/3.13	= Acc.	
Heartbeat	68.29/72.19	64.02/52.70	48.82/14.93	73.19/73.90	0.59/0.57	54.39/8.77	73.65/96.62
JapaneseVowels	97.84/97.84	97.87/98.15	97.84/97.85	99.94/99.91	0.09/0.12	= Acc.	

Continued on next page

Table 25 (continued)

Dataset	Acc.↑	BAcc.↑	F1↑	AUROC↑	LL↓	Sens.↑	Spec.↑
LSST	57.62/53.73	40.62/38.73	53.95/51.53	85.33/83.98	1.43/1.55	= Acc.	
Libras	75.00/73.33	75.00/73.33	74.20/73.20	97.19/97.35	0.81/0.74	= Acc.	
MotorImagery	52.00/52.00	52.00/52.00	56.36/57.90	48.88/50.76	1.05/1.20	62.00/66.00	42.00/38.00
NATOPS	83.33/92.78	83.33/92.78	83.13/92.80	96.90/99.39	0.36/0.18	= Acc.	
PEMS-SF	72.83/72.83	73.23/72.52	72.36/72.57	94.88/95.32	0.75/0.78	= Acc.	
PhonemeSpectra	12.59/11.90	12.59/11.90	12.04/11.48	73.78/74.08	7.44/7.03	= Acc.	
RacketSports	83.55/81.58	84.68/82.94	83.10/81.24	94.46/92.42	0.44/0.51	= Acc.	
SelfRegulationSCP1	81.91/83.96	81.96/84.00	84.08/85.63	95.35/95.27	0.39/0.47	95.89/95.89	68.03/72.11
SpokenArabicDigits	98.14/97.45	98.14/97.45	98.14/97.46	99.97/99.95	0.09/0.11	= Acc.	
StandWalkJump	26.67/33.33	26.67/33.33	25.45/29.10	52.00/52.00	1.12/1.12	= Acc.	
UWaveGestureLibrary	79.06/79.06	79.06/79.06	77.04/77.72	96.89/97.12	0.87/0.86	= Acc.	
ConvTran							
ArticularyWordRecognition	96.33/97.00	96.33/97.00	96.11/96.94	99.92/99.98	0.19/0.10	= Acc.	
AtrialFibrillation	33.33/33.33	33.33/33.33	27.32/26.50	52.67/55.33	1.13/1.11	= Acc.	
CharacterTrajectories	99.16/98.89	99.12/98.80	99.17/98.88	100.00/99.99	0.03/0.04	= Acc.	
Cricket	97.22/98.61	97.22/98.61	97.21/98.60	99.98/99.96	0.23/0.07	= Acc.	
DuckDuckGeese	52.00/52.00	52.00/52.00	51.48/51.10	83.95/80.40	1.23/1.58	= Acc.	
ERing	87.04/87.41	87.04/87.41	86.66/87.31	98.10/98.10	0.82/0.63	= Acc.	
Epilepsy	95.65/96.38	95.77/96.50	95.64/96.38	99.51/99.56	0.16/0.18	= Acc.	
EthanolConcentration	31.56/26.62	31.53/26.57	31.33/25.67	58.38/56.90	1.45/1.50	= Acc.	
HandMovementDirection	35.13/33.78	36.07/36.43	33.93/33.23	62.80/61.49	1.34/1.35	= Acc.	
Handwriting	22.59/23.88	22.07/24.09	17.13/19.83	79.36/80.32	2.69/2.74	= Acc.	
Heartbeat	74.15/75.61	65.38/67.47	49.52/52.83	74.55/76.90	0.52/0.52	45.61/49.12	85.14/85.81
JapaneseVowels	99.19/97.84	99.30/98.45	99.19/97.84	99.98/99.99	0.05/0.08	= Acc.	
LSST	62.13/62.57	38.89/40.68	58.21/58.28	88.08/88.80	1.19/1.16	= Acc.	
Libras	83.89/88.33	83.89/88.33	83.81/88.34	99.00/99.33	0.62/0.37	= Acc.	
MotorImagery	50.00/60.00	50.00/60.00	66.67/55.56	51.36/64.84	0.80/0.66	100.00/50.00	0.00/70.00
NATOPS	95.00/95.00	95.00/95.00	95.07/94.92	99.84/99.77	0.14/0.18	= Acc.	
PEMS-SF	68.21/68.79	68.37/69.36	67.33/68.15	90.31/92.45	1.06/0.99	= Acc.	
PhonemeSpectra	26.36/27.35	26.37/27.35	25.69/25.94	90.57/90.65	2.38/2.37	= Acc.	
RacketSports	82.89/83.55	84.27/84.98	82.78/83.18	92.57/92.53	0.58/0.66	= Acc.	
SelfRegulationSCP1	90.10/86.69	90.12/86.72	90.61/87.93	97.36/96.63	0.28/0.34	95.89/97.26	84.35/76.19
SpokenArabicDigits	98.27/98.95	98.27/98.95	98.27/98.95	99.98/100.00	0.07/0.04	= Acc.	
StandWalkJump	26.67/33.33	26.67/33.33	22.22/16.67	51.33/38.00	1.25/1.11	= Acc.	
UWaveGestureLibrary	87.19/89.38	87.19/89.38	87.08/89.19	97.43/98.08	0.53/0.43	= Acc.	
Arsenal							
ArticularyWordRecognition	99.33/99.33	99.33/99.33	99.33/99.33	99.83/99.65	0.13/0.23	= Acc.	
AtrialFibrillation	6.67/6.67	6.67/6.67	5.56/5.56	27.00/27.33	33.64/32.24	= Acc.	
CharacterTrajectories	99.30/98.95	99.28/98.88	99.30/98.95	99.78/99.71	0.16/0.21	= Acc.	
Cricket	100.00/100.00	100.00/100.00	100.00/100.00	100.00/100.00	0.00/0.00	= Acc.	
DuckDuckGeese	48.00/56.00	48.00/56.00	48.47/56.10	78.17/81.50	7.87/6.21	= Acc.	
ERing	98.89/97.41	98.89/97.41	98.89/97.42	99.55/99.14	0.28/0.53	= Acc.	
Epilepsy	98.55/99.28	98.53/99.27	98.54/99.28	100.00/99.99	0.01/0.02	= Acc.	
EthanolConcentration	42.59/35.36	42.68/35.41	42.34/35.41	67.51/61.05	9.38/12.35	= Acc.	
HandMovementDirection	50.00/52.70	48.69/53.21	49.89/52.91	72.63/76.03	6.59/4.56	= Acc.	
Handwriting	59.18/58.94	58.50/58.34	56.58/56.41	85.06/85.31	10.36/9.68	= Acc.	
Heartbeat	76.10/75.61	64.03/63.69	46.15/45.65	71.40/72.39	3.99/3.07	36.84/36.84	91.22/90.54
JapaneseVowels	83.24/82.43	82.82/81.97	82.68/81.74	95.99/95.99	2.44/2.31	= Acc.	
LSST	65.69/63.34	37.06/33.39	60.32/57.40	84.86/84.48	6.66/6.16	= Acc.	
Libras	90.00/90.00	90.00/90.00	90.04/90.04	96.43/95.55	2.47/2.90	= Acc.	
MotorImagery	50.00/49.00	50.00/49.00	47.92/51.43	48.04/50.18	2.00/1.61	46.00/54.00	54.00/44.00
NATOPS	88.33/90.00	88.33/90.00	87.99/89.79	97.40/97.54	1.34/1.27	= Acc.	
PEMS-SF	82.66/82.66	82.01/82.01	82.14/82.14	94.42/96.20	2.94/1.56	= Acc.	
PhonemeSpectra	29.88/29.56	29.89/29.56	29.09/28.91	75.84/76.47	15.67/14.37	= Acc.	
RacketSports	90.13/90.79	90.93/91.47	90.13/90.75	96.06/96.05	1.76/1.69	= Acc.	
SelfRegulationSCP1	85.32/85.32	85.31/85.31	84.81/84.70	92.12/93.36	1.59/1.07	82.19/81.51	88.44/89.12
SpokenArabicDigits	99.14/99.09	99.14/99.09	99.13/99.09	99.88/99.91	0.10/0.08	= Acc.	
StandWalkJump	46.67/46.67	46.67/46.67	43.37/43.39	61.33/66.00	17.02/14.22	= Acc.	
UWaveGestureLibrary	94.06/93.75	94.06/93.75	94.04/93.72	97.65/97.66	1.52/1.44	= Acc.	

I PER-DATASET ANOMALY-DETECTION AUGMENTATION RESULTS

Table 26 reports *Real only*/*Real + MarkovTS* per dataset–detector pair to three decimals; all six metrics are higher-is-better. Appendices D.6 and D.3 define protocols and metrics.

Table 26: Per-dataset anomaly detection: Real only/Real + MarkovTS. Higher is better.

Dataset	A-P↑	A-R↑	R-A-P↑	R-A-R↑	V-PR↑	V-ROC↑
DLinear						
ASD	0.179/0.172	0.762/0.762	0.208/0.204	0.815/0.814	0.205/0.200	0.806/0.806
CATSv2	0.064/0.063	0.567/0.564	0.078/0.077	0.654/0.650	0.079/0.078	0.648/0.645
CICIDS	0.002/0.002	0.748/0.749	0.002/0.002	0.683/0.680	0.002/0.002	0.699/0.700
CalIt2	0.100/0.108	0.776/0.800	0.101/0.107	0.808/0.826	0.099/0.105	0.799/0.818
Creditcard	0.070/0.055	0.919/0.748	0.036/0.006	0.847/0.589	0.035/0.010	0.851/0.622
DAPHNET	0.196/0.198	0.735/0.746	0.316/0.320	0.816/0.825	0.307/0.310	0.809/0.818

Continued on next page

Table 26 (continued)

Dataset	A-P↑	A-R↑	R-A-P↑	R-A-R↑	V-PR↑	V-ROC↑
DLR	0.160/0.134	0.916/0.909	0.141/0.123	0.904/0.903	0.144/0.125	0.905/0.904
ECG	0.550/0.550	0.848/0.837	0.475/0.468	0.790/0.774	0.522/0.523	0.822/0.811
Exathlon	0.798/0.794	0.900/0.889	0.733/0.714	0.871/0.861	0.740/0.722	0.872/0.862
GECCO	0.116/0.110	0.791/0.786	0.163/0.163	0.891/0.889	0.153/0.151	0.875/0.870
GHL	0.010/0.010	0.322/0.305	0.018/0.017	0.492/0.480	0.018/0.017	0.478/0.465
GutenTAG	0.103/0.151	0.682/0.688	0.145/0.198	0.763/0.786	0.136/0.191	0.751/0.772
LTDB	0.204/0.189	0.599/0.563	0.310/0.306	0.710/0.695	0.300/0.292	0.695/0.674
MITDB	0.054/0.059	0.583/0.572	0.078/0.082	0.669/0.660	0.078/0.082	0.661/0.651
MSL	0.139/0.134	0.601/0.584	0.208/0.203	0.678/0.660	0.206/0.200	0.671/0.653
NYC	0.038/0.032	0.699/0.631	0.053/0.049	0.714/0.682	0.053/0.049	0.712/0.675
PSM	0.352/0.342	0.559/0.550	0.353/0.343	0.559/0.550	0.353/0.343	0.557/0.546
PUMP	0.120/0.120	0.562/0.565	0.178/0.175	0.653/0.653	0.177/0.174	0.647/0.648
SKAB	0.420/0.416	0.570/0.564	0.658/0.649	0.733/0.726	0.642/0.635	0.725/0.719
SMAP	0.104/0.104	0.391/0.394	0.118/0.118	0.427/0.430	0.119/0.119	0.427/0.430
SMD	0.133/0.135	0.708/0.706	0.139/0.139	0.733/0.729	0.138/0.139	0.731/0.728
SVDB	0.052/0.048	0.610/0.579	0.091/0.087	0.710/0.688	0.085/0.081	0.696/0.673
SWAN	0.446/0.454	0.494/0.507	0.369/0.378	0.392/0.404	0.401/0.410	0.436/0.447
SWAT	0.085/0.630	0.304/0.819	0.116/0.589	0.397/0.813	0.114/0.576	0.390/0.804
TAO	1.000/1.000	1.000/1.000	0.999/0.999	1.000/1.000	0.955/0.960	0.987/0.990
TODS	0.576/0.573	0.828/0.824	0.549/0.579	0.832/0.834	0.535/0.559	0.822/0.824
TranAD						
ASD	0.149/0.151	0.619/0.607	0.133/0.138	0.640/0.620	0.133/0.139	0.635/0.616
CATSv2	0.126/0.127	0.594/0.603	0.128/0.130	0.625/0.630	0.127/0.129	0.617/0.623
CICIDS	0.001/0.001	0.698/0.680	0.002/0.002	0.627/0.604	0.002/0.002	0.647/0.629
CalIt2	0.053/0.061	0.520/0.577	0.062/0.070	0.554/0.615	0.060/0.069	0.544/0.604
Creditcard	0.107/0.109	0.952/0.954	0.060/0.061	0.907/0.911	0.057/0.058	0.902/0.906
DAPHNET	0.154/0.135	0.571/0.502	0.262/0.224	0.683/0.597	0.253/0.216	0.671/0.587
DLR	0.241/0.253	0.895/0.903	0.266/0.204	0.952/0.955	0.268/0.203	0.945/0.950
ECG	0.524/0.484	0.872/0.859	0.476/0.443	0.807/0.793	0.493/0.460	0.817/0.803
Exathlon	0.920/0.920	0.947/0.947	0.844/0.846	0.920/0.920	0.847/0.847	0.919/0.918
GECCO	0.269/0.300	0.719/0.789	0.052/0.063	0.586/0.673	0.052/0.063	0.587/0.674
GHL	0.013/0.015	0.303/0.301	0.017/0.019	0.339/0.349	0.017/0.019	0.336/0.346
GutenTAG	0.025/0.035	0.569/0.585	0.033/0.038	0.591/0.607	0.033/0.038	0.589/0.605
LTDB	0.217/0.219	0.590/0.596	0.330/0.321	0.719/0.710	0.321/0.314	0.705/0.699
MITDB	0.108/0.108	0.691/0.687	0.126/0.125	0.782/0.777	0.127/0.126	0.773/0.768
MSL	0.128/0.126	0.478/0.487	0.180/0.180	0.562/0.571	0.179/0.179	0.556/0.564
NYC	0.034/0.032	0.676/0.675	0.048/0.047	0.696/0.698	0.047/0.046	0.688/0.688
PSM	0.502/0.451	0.635/0.624	0.438/0.407	0.573/0.557	0.438/0.407	0.574/0.558
PUMP	0.210/0.211	0.802/0.802	0.251/0.254	0.796/0.796	0.249/0.252	0.794/0.794
SKAB	0.750/0.767	0.827/0.836	0.852/0.860	0.871/0.877	0.843/0.852	0.872/0.878
SMAP	0.097/0.097	0.369/0.374	0.112/0.112	0.378/0.387	0.112/0.112	0.376/0.386
SMD	0.112/0.107	0.663/0.654	0.105/0.100	0.670/0.663	0.105/0.101	0.669/0.662
SVDB	0.064/0.062	0.577/0.567	0.088/0.086	0.657/0.650	0.086/0.083	0.653/0.645
SWAN	0.530/0.556	0.603/0.632	0.408/0.432	0.423/0.456	0.437/0.460	0.466/0.496
SWAT	0.728/0.722	0.818/0.817	0.527/0.503	0.718/0.700	0.498/0.478	0.691/0.675
TAO	1.000/1.000	1.000/1.000	0.999/0.999	1.000/1.000	0.948/0.948	0.980/0.980
TODS	0.174/0.173	0.525/0.525	0.134/0.134	0.491/0.491	0.134/0.134	0.487/0.487
CATCH						
ASD	0.170/0.186	0.738/0.790	0.193/0.226	0.790/0.839	0.191/0.221	0.782/0.830
CATSv2	0.090/0.084	0.604/0.601	0.104/0.100	0.682/0.675	0.105/0.101	0.680/0.674
CICIDS	0.001/0.001	0.669/0.684	0.001/0.002	0.582/0.593	0.002/0.002	0.598/0.610
CalIt2	0.089/0.101	0.772/0.790	0.104/0.109	0.811/0.818	0.101/0.107	0.796/0.808
Creditcard	0.102/0.100	0.953/0.951	0.052/0.052	0.912/0.908	0.050/0.050	0.909/0.906
DAPHNET	0.193/0.195	0.730/0.739	0.310/0.313	0.809/0.818	0.301/0.303	0.801/0.810
DLR	0.171/0.143	0.922/0.904	0.162/0.145	0.927/0.921	0.161/0.144	0.921/0.914
ECG	0.526/0.534	0.860/0.860	0.458/0.462	0.803/0.802	0.499/0.505	0.828/0.828
Exathlon	0.782/0.798	0.882/0.882	0.738/0.730	0.864/0.862	0.744/0.738	0.864/0.862
GECCO	0.198/0.227	0.912/0.941	0.275/0.320	0.963/0.976	0.260/0.300	0.953/0.969
GHL	0.009/0.009	0.233/0.246	0.016/0.017	0.442/0.451	0.016/0.017	0.428/0.437
GutenTAG	0.246/0.288	0.791/0.828	0.428/0.466	0.839/0.870	0.406/0.444	0.832/0.863
LTDB	0.213/0.207	0.596/0.578	0.321/0.321	0.713/0.709	0.310/0.306	0.695/0.685
MITDB	0.096/0.135	0.708/0.727	0.121/0.152	0.756/0.768	0.121/0.152	0.750/0.763
MSL	0.154/0.155	0.627/0.632	0.233/0.228	0.709/0.711	0.229/0.224	0.700/0.702
NYC	0.021/0.020	0.469/0.452	0.040/0.040	0.606/0.605	0.040/0.039	0.593/0.591
PSM	0.385/0.386	0.600/0.600	0.388/0.389	0.600/0.599	0.389/0.390	0.600/0.600
PUMP	0.104/0.104	0.466/0.476	0.169/0.170	0.591/0.598	0.168/0.169	0.583/0.590
SKAB	0.441/0.426	0.589/0.579	0.661/0.653	0.735/0.729	0.649/0.641	0.733/0.727
SMAP	0.115/0.116	0.431/0.430	0.129/0.129	0.459/0.456	0.129/0.129	0.457/0.455
SMD	0.164/0.172	0.782/0.790	0.167/0.174	0.804/0.810	0.166/0.173	0.802/0.808
SVDB	0.062/0.079	0.602/0.623	0.091/0.103	0.703/0.723	0.086/0.098	0.686/0.706
SWAN	0.425/0.437	0.485/0.501	0.346/0.357	0.373/0.386	0.376/0.387	0.415/0.428
SWAT	0.082/0.083	0.230/0.232	0.120/0.122	0.346/0.347	0.116/0.119	0.336/0.338
TAO	0.998/0.998	0.997/0.997	0.994/0.994	0.994/0.994	0.954/0.953	0.984/0.984
TODS	0.610/0.617	0.911/0.915	0.596/0.605	0.907/0.912	0.586/0.595	0.905/0.909
ModernTCN						
ASD	0.119/0.127	0.624/0.627	0.121/0.127	0.670/0.675	0.121/0.127	0.665/0.669
CATSv2	0.109/0.106	0.630/0.633	0.127/0.125	0.716/0.716	0.127/0.125	0.708/0.710
CICIDS	0.001/0.001	0.670/0.678	0.001/0.002	0.607/0.618	0.002/0.002	0.622/0.632
CalIt2	0.063/0.064	0.691/0.692	0.081/0.081	0.754/0.756	0.079/0.080	0.740/0.741
Creditcard	0.088/0.090	0.958/0.958	0.054/0.055	0.920/0.921	0.051/0.053	0.911/0.911
DAPHNET	0.182/0.180	0.710/0.710	0.293/0.291	0.793/0.792	0.284/0.282	0.785/0.784
DLR	0.289/0.272	0.961/0.959	0.203/0.206	0.954/0.955	0.206/0.208	0.957/0.957
ECG	0.522/0.524	0.865/0.866	0.468/0.471	0.807/0.808	0.502/0.505	0.830/0.831

Continued on next page

Table 26 (continued)

Dataset	A-P↑	A-R↑	R-A-P↑	R-A-R↑	V-PR↑	V-ROC↑
Exathlon	0.784/0.787	0.887/0.887	0.755/0.751	0.867/0.865	0.760/0.757	0.867/0.865
GECCO	0.437/0.484	0.954/0.964	0.429/0.451	0.973/0.982	0.431/0.453	0.971/0.980
GHL	0.010/0.010	0.290/0.294	0.018/0.018	0.482/0.486	0.018/0.018	0.467/0.470
GutenTAG	0.061/0.064	0.607/0.609	0.141/0.147	0.707/0.710	0.124/0.130	0.695/0.698
LTDB	0.230/0.231	0.619/0.621	0.317/0.318	0.702/0.703	0.314/0.315	0.699/0.701
MITDB	0.112/0.107	0.663/0.670	0.129/0.128	0.734/0.736	0.129/0.128	0.727/0.729
MSL	0.145/0.147	0.621/0.625	0.224/0.221	0.701/0.703	0.221/0.218	0.693/0.695
NYC	0.045/0.044	0.696/0.711	0.061/0.062	0.723/0.734	0.060/0.060	0.716/0.727
PSM	0.388/0.388	0.587/0.590	0.387/0.386	0.584/0.585	0.387/0.386	0.584/0.582
PUMP	0.111/0.113	0.420/0.413	0.209/0.219	0.593/0.589	0.205/0.213	0.581/0.577
SKAB	0.439/0.439	0.586/0.587	0.674/0.674	0.746/0.746	0.659/0.659	0.739/0.740
SMAP	0.110/0.110	0.434/0.433	0.124/0.125	0.470/0.468	0.125/0.125	0.468/0.467
SMD	0.132/0.138	0.721/0.728	0.131/0.138	0.739/0.748	0.131/0.138	0.738/0.746
SVDB	0.063/0.064	0.593/0.594	0.086/0.086	0.666/0.666	0.084/0.084	0.662/0.663
SWAN	0.454/0.472	0.519/0.537	0.376/0.392	0.413/0.429	0.406/0.422	0.456/0.472
SWAT	0.092/0.092	0.240/0.242	0.135/0.136	0.354/0.356	0.130/0.130	0.345/0.346
TAO	0.999/0.999	1.000/1.000	0.997/0.998	1.000/1.000	0.949/0.950	0.983/0.983
TODS	0.173/0.206	0.618/0.634	0.183/0.211	0.640/0.662	0.184/0.211	0.640/0.660
TimesNet						
ASD	0.266/0.321	0.822/0.837	0.285/0.342	0.850/0.868	0.281/0.338	0.845/0.864
CATSv2	0.261/0.278	0.712/0.732	0.266/0.286	0.779/0.790	0.265/0.284	0.772/0.783
CICIDS	0.002/0.002	0.691/0.705	0.002/0.002	0.589/0.608	0.002/0.002	0.600/0.615
CalIt2	0.064/0.072	0.623/0.667	0.063/0.081	0.661/0.736	0.062/0.078	0.649/0.720
Creditcard	0.095/0.090	0.932/0.922	0.026/0.028	0.877/0.857	0.029/0.032	0.890/0.876
DAPHNET	0.211/0.205	0.748/0.744	0.329/0.320	0.825/0.823	0.319/0.311	0.818/0.815
DLR	0.065/0.066	0.910/0.910	0.073/0.075	0.910/0.910	0.073/0.074	0.908/0.909
ECG	0.437/0.393	0.808/0.801	0.373/0.343	0.744/0.735	0.421/0.393	0.775/0.764
Exathlon	0.794/0.798	0.920/0.916	0.759/0.770	0.890/0.892	0.765/0.774	0.891/0.891
GECCO	0.327/0.305	0.898/0.893	0.301/0.302	0.948/0.946	0.297/0.298	0.941/0.940
GHL	0.010/0.010	0.298/0.301	0.019/0.018	0.491/0.495	0.019/0.018	0.477/0.480
GutenTAG	0.227/0.295	0.753/0.766	0.244/0.350	0.822/0.827	0.238/0.334	0.812/0.819
LTDB	0.289/0.292	0.652/0.652	0.396/0.400	0.760/0.759	0.384/0.388	0.745/0.744
MITDB	0.104/0.105	0.657/0.656	0.141/0.139	0.732/0.733	0.140/0.139	0.725/0.726
MSL	0.148/0.140	0.612/0.581	0.229/0.216	0.702/0.671	0.225/0.212	0.693/0.662
NYC	0.022/0.021	0.505/0.469	0.056/0.054	0.652/0.633	0.053/0.051	0.636/0.610
PSM	0.356/0.346	0.558/0.549	0.359/0.349	0.557/0.549	0.360/0.350	0.558/0.549
PUMP	0.118/0.113	0.451/0.415	0.222/0.212	0.614/0.580	0.216/0.205	0.603/0.570
SKAB	0.441/0.438	0.586/0.581	0.679/0.676	0.747/0.743	0.665/0.661	0.743/0.738
SMAP	0.119/0.121	0.442/0.445	0.136/0.139	0.482/0.484	0.136/0.138	0.480/0.483
SMD	0.161/0.157	0.762/0.755	0.159/0.154	0.781/0.773	0.159/0.153	0.780/0.771
SVDB	0.166/0.130	0.708/0.669	0.200/0.169	0.781/0.756	0.192/0.162	0.770/0.744
SWAN	0.604/0.598	0.701/0.712	0.484/0.491	0.559/0.568	0.516/0.522	0.597/0.604
SWAT	0.109/0.095	0.288/0.253	0.186/0.154	0.402/0.367	0.175/0.147	0.392/0.357
TAO	0.900/0.893	0.970/0.971	0.831/0.824	0.945/0.948	0.815/0.811	0.944/0.948
TODS	0.660/0.668	0.908/0.916	0.619/0.654	0.906/0.920	0.608/0.641	0.900/0.915

J GENERATION UNDER PARTIAL CODE RANDOMIZATION

Figures 11–13 examine how generation responds to increasing disruption of reference-derived state and transition information. At selected positions, code IDs are replaced uniformly from the full codebook, with replacement fractions from 0% to 80%. The three rows perturb states, transitions, or both; single-stream interventions retain the other stream. Each dataset uses the first reference example, with the reference and Gaussian source fixed across panels. The Solar example illustrates changes in peak shape and amplitude as replacement increases, with joint randomization visibly disrupting the reference pattern. These comparisons show the generator's dependence on both streams.

K LIMITATIONS

A single MarkovTS generation call supports 512–4096 time steps. Longer generated training counterparts are constructed through segment-wise generation followed by generated-only concatenation, as described in Appendix D. The main distributional benchmark evaluates length-168 windows cut from length-512 outputs and therefore does not assess full-length-512 fidelity or the fidelity of longer concatenated generated sequences. Moreover, the representation remains patch-based, so aggregation within patches can smooth fine-grained within-patch variation.

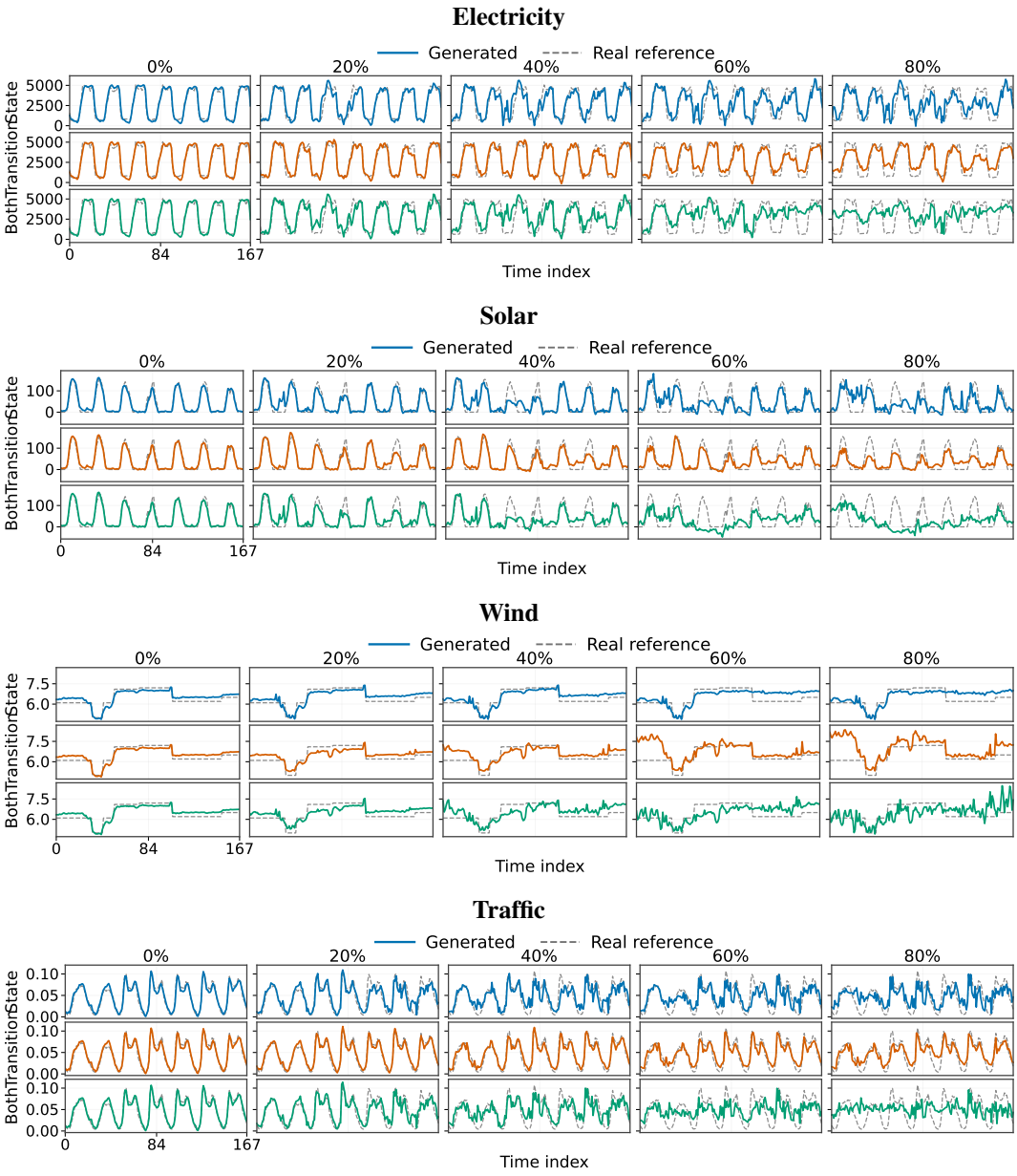


Figure 11: Generation under partial code randomization on Electricity, Solar, Wind, and Traffic. Rows randomize states, transitions, or both; columns show replacement fractions. Colored curves are generated sequences; dashed gray curves are references.

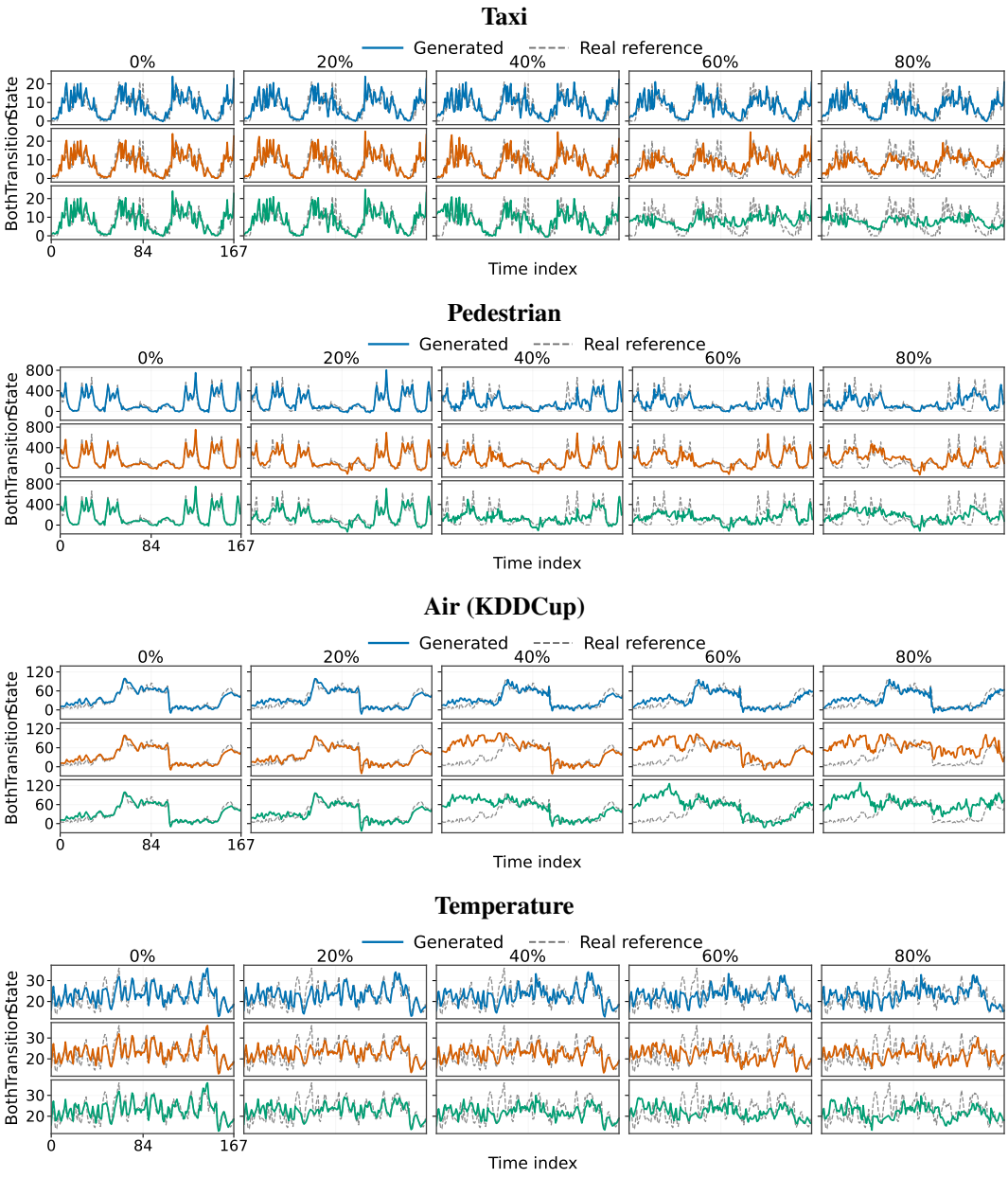


Figure 12: Generation under partial code randomization on Taxi, Pedestrian, Air (KDDCup), and Temperature.

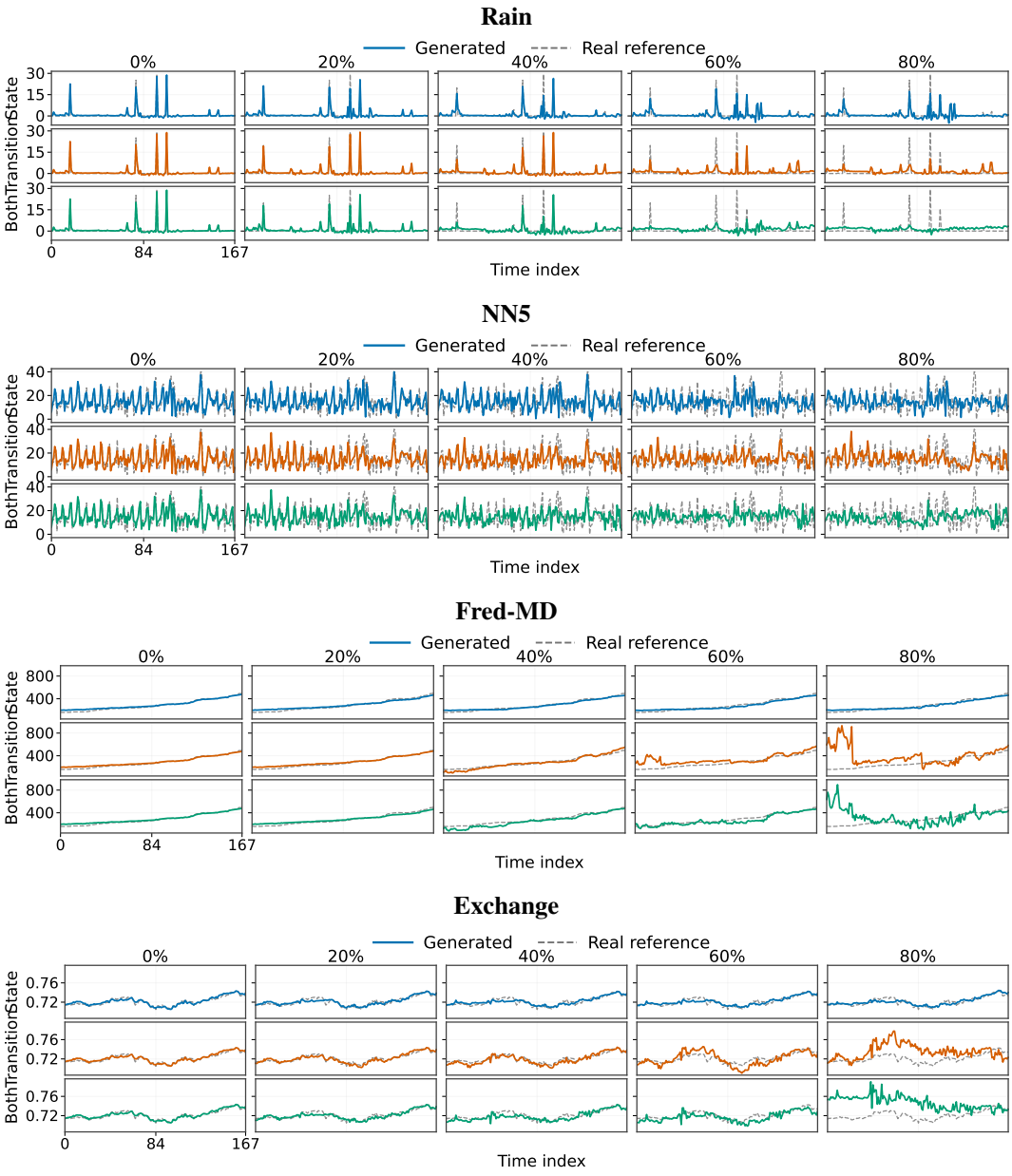


Figure 13: Generation under partial code randomization on Rain, NN5, Fred-MD, and Exchange.