VQ-PACT: TEXT-CONTROLLED TIME SERIES GENERATION WITH A COMPOSITIONAL TEMPORAL GRAMMAR

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-controlled time series generation aims to synthesize sequences that follow natural-language descriptions while remaining faithful to real data distributions. Existing methods typically couple text understanding and sequence generation within a unified representation, making it difficult to accurately capture described local variations, such as sharp peaks and deep troughs, while maintaining distributional fidelity. To address this challenge, we propose VQ-PACT, which represents morphological features in continuous-valued sequences as reusable discrete units. Guided by the temporal semantics of text, these units are composed and their attributes, including amplitude and duration, are adjusted to capture both overall changes and local features. Specifically, VQ-PACT learns separate feature codebooks for overall trends, recurring fluctuations, and local transients, allowing different types of morphological features to be represented separately and reused across combinations. An invariance constraint encourages the encoding of a feature to remain stable as other features vary, supporting consistent reuse. Textual phrases describing morphological changes are then aligned with compatible codebook features. A text-conditioned generative model learns how features from different codebooks combine, enabling adaptive feature selection, temporal organization, and attribute prediction. Joint temporal decoding reconstructs the sequence shape. Finally, a mixture density network models and samples the overall level and variability, restoring the global numerical scale to obtain the generated sequence. Experiments on twelve real-world univariate datasets, the controlled Synth-U benchmark, and real reports from eight Time-MMD domains demonstrate improved distributional fidelity and fine-grained text control, with additional evidence of generalization to unseen combinations of temporal features.

1 INTRODUCTION

Text-controlled time series generation aims to synthesize sequences that satisfy natural-language descriptions while remaining faithful to real data distributions. It extends the use of synthetic time series in scenario simulation, system behavior modeling, and decision support by incorporating semantic requirements into the generation process (Yoon et al., 2019; Gu et al., 2025; Ge et al., 2025). Compared with conventional time series generation, natural-language conditioning provides a more flexible interface for expressing scenario-specific requirements. The task therefore calls for generators that can translate these requirements into temporal behavior without compromising the distributional characteristics of real observations.

Recent studies connect textual descriptions with time series through cross-modal representation learning and language-conditioned generative modeling. VerbalTS (Gu et al., 2025) uses multi-focal alignment to learn correspondences between text and temporal features. T2S (Ge et al., 2025) integrates textual representations with length-adaptive time series representations through flow matching, supporting text-conditioned generation at different lengths. These advances strengthen textual guidance, but semantic alignment alone does not establish separately controllable representations of different temporal features. When these features remain coupled, satisfying a local requirement can also affect other aspects of the sequence, making precise local generation and overall distributional fidelity difficult to reconcile. Figure 1(a) illustrates this issue with "a gradual rise with regular oscillations and a spike near the end." Replacing "spike" with "dip" changes only the local event requirement; the gradual rise and regular oscillations should still be satisfied. However, modifying a

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.

coupled representation may also change these features. This motivates explicit representations of the morphological features described by text, together with a mechanism for selecting and composing them according to the specified temporal relations. Such representations should distinguish the features being controlled while retaining their dependencies during generation.

To this end, we propose VQ-PACT, which represents morphological features in continuous-valued sequences as reusable discrete units and composes them under textual guidance (Figure 1(b)). First, separate codebooks are learned for overall trends, recurring fluctuations, and local transients. These codebooks provide distinct representations for different types of morphological variation. An invariance constraint encourages a feature's encoding to remain stable when other features change, reducing its dependence on a particular combination and supporting reuse. Next, textual phrases describing morphological changes are aligned with codebook features according to their decoded shapes. The generative model jointly learns how features from different codebooks combine under textual constraints, allowing adaptive feature selection, temporal organization, and attribute prediction. Finally, the selected features and their attributes are jointly decoded to reconstruct the sequence shape. A mixture density network (MDN) (Bishop, 1994) models and samples the overall level and variability to restore its global numerical scale. By separating feature representations while jointly modeling their combinations, VQ-PACT provides a basis for fine-grained semantic control without treating the constituent features as statistically independent. The overall framework is presented in Figure 2.

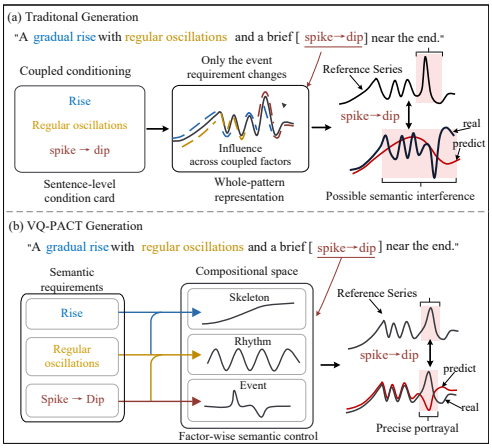


Figure 1: Conceptual comparison of local text control. (a) Coupled conditioning may also affect other features. (b) VQ-PACT separates the corresponding representations to support event-specific control while retaining the requested trend and oscillations.

Our main contributions are summarized as follows:

- A compositional text-controlled generation framework, VQ-PACT, is proposed. Sequence generation is formulated as the selection and composition of reusable morphological features under textual guidance. Different types of temporal variation are represented separately, while their combinations are modeled jointly. Joint temporal decoding and scale restoration translate these representations into continuous-valued sequences.
- Discrete temporal representations enable reuse across combinations. Separate codebooks capture overall trends, recurring fluctuations, and local transients, while an invariance constraint promotes feature encodings robust to variations in other features.
- Shape-based phrase alignment links textual phrases to compatible discrete units. Text-conditioned cross-codebook dependencies then guide feature selection, organization, and attribute prediction, while joint decoding and global scale modeling integrate local requirements into complete sequences.
- Extensive experiments are conducted on twelve real-world univariate datasets, the controlled Synth-U benchmark, and real reports from eight Time-MMD domains. The results demonstrate improved distributional fidelity and fine-grained text control. Generalization to unseen temporal feature combinations is further evaluated through held-out combination tests.

2 Related Work

2.1 Time Series Generation and Structured Representations

Time-series generation has been studied with adversarial (Yoon et al., 2019; Jeon et al., 2022), variational (Desai et al., 2021), and diffusion models (Yuan & Qiao, 2024). Recent work further

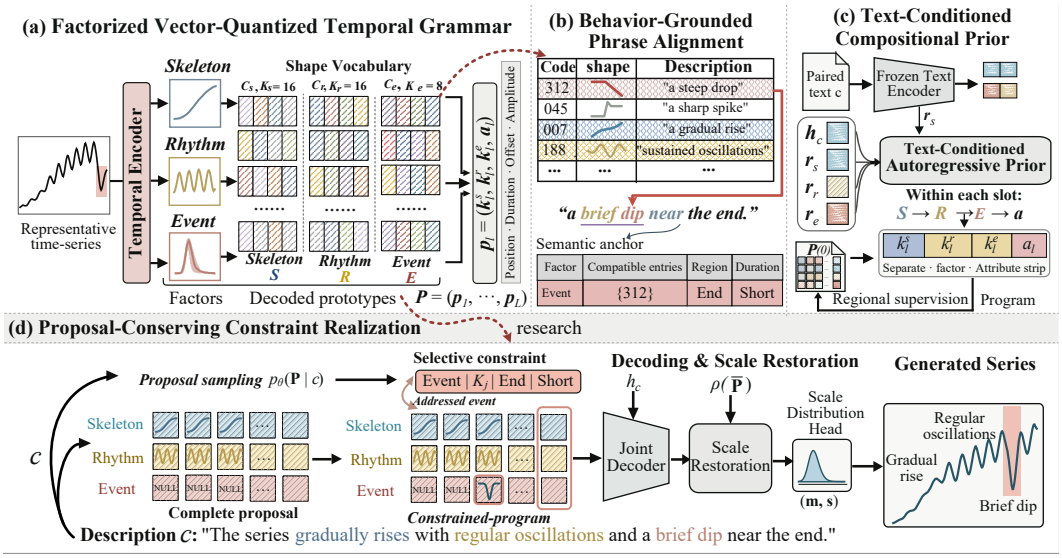


Figure 2: The framework of VQ-PACT.

introduces structured representations through temporal decomposition, frequency separation, or multiscale modeling (Lee et al., 2023; Xu et al., 2026). VQShape associates quantized representations with abstract local shapes (Wen et al., 2024). Building on these ideas, VQ-PACT learns factor-specific discrete representations and connects them to phrase-level semantics for compositional generation.

2.2 Conditional and Text-Controlled Generation

Conditional generation has incorporated metadata, domain prompts, and natural language (Narasimhan et al., 2024; Huang et al., 2025; Gu et al., 2025; Li et al., 2025; Ge et al., 2025). DiffuSETS further conditions ECG generation on clinical reports and patient information (Lai et al., 2025), while ConTSG-Bench and Time-MMD provide benchmarks and multimodal resources for conditional generation (Lan et al., 2026; Liu et al., 2024). VQ-PACT differs by explicitly mapping language requirements to composable discrete factors and their scopes of control.

2.3 Temporal Grounding and Selective Editing

Prior work links temporal structures with language through truth-conditional captioning (Jhamtani & Berg-Kirkpatrick, 2021), while TEdit and InstructTime modify selected attributes of an existing series (Jing et al., 2024; Qiu et al., 2026). VQ-PACT instead generates a complete sequence without requiring a source series, while distinguishing text-specified factors from unspecified content and preserving their respective roles during generation.

3 Methodology

As shown in Figure 2, VQ-PACT learns reusable temporal features with factor-specific codebooks and aligns them with textual phrases. A text-conditioned prior jointly predicts their combinations and continuous attributes. Selective constraint realization, joint decoding, and scale restoration produce the final sequence.

3.1 Factorized Vector-Quantized Temporal Grammar

Sequences with similar overall trends can differ in recurring fluctuations and local events. We represent these variations with three codebooks $\mathcal{C}_f = \{e_k^f : k \in \mathcal{I}_f\}$, where $f \in \{s, r, e\}$ denotes skeleton, rhythm, and event, respectively. A sequence is represented by $P = (p_1, \ldots, p_L)$, whose slots $p_\ell = (k_\ell^s, k_\ell^r, k_\ell^e, a_\ell)$ contain three code indices and shared continuous attributes $a_\ell = (t_\ell, d_\ell, \mu_\ell, \sigma_\ell)$

for position, duration, offset, and amplitude. We use $L = 64$ and validation-selected codebook sizes $(K_s, K_r, K_e) = (16, 16, 8)$.

For each standardized shape u_ℓ, smoothing, residual Fourier analysis, and transient masking yield factor views v_ℓ^s, v_ℓ^r, and v_ℓ^e. These views sum to u_ℓ; the rhythm view includes recurring variation and residual detail outside the event component. Each encoder E_f maps its view to an embedding, which is assigned to the nearest eligible code (van den Oord et al., 2017). Absent events select a reserved null code, while events use non-null entries.

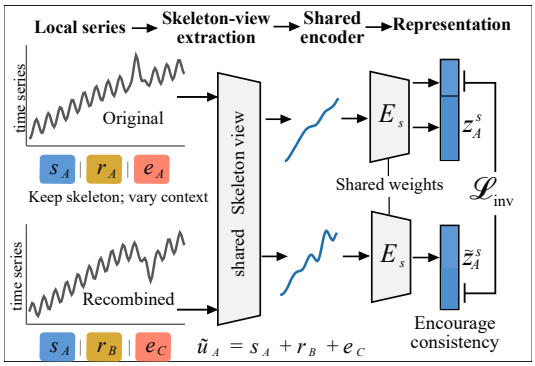


Figure 3: Re-extraction and shared encoding encourage stable skeleton representations when rhythm and event views change.

Auxiliary decoders D_f reconstruct the factor views, giving the codes morphological interpretations. A joint decoder Dec_ψ reconstructs the sequence from the same code embeddings and their attributes. A consistency loss encourages the summed factor reconstructions to agree with the joint reconstruction, connecting the shapes used for semantic alignment to sequence generation.

To support reuse across combinations, we retain one factor and recombine the others across training examples. For skeleton invariance, the skeleton of example A is combined with the rhythm of B and the event of C:

$$\widetilde{u}_A = v_A^s + v_B^r + v_C^e, \qquad \widetilde{z}_A^s = E_s(\mathcal{F}_s(\widetilde{u}_A)), \tag{1}$$

where $\mathcal{F}_s$ re-extracts the skeleton view. The invariance objective encourages $\widetilde{z}_A^s$ to remain close to $z_A^s = E_s(v_A^s)$ before quantization (Figure 3). The same construction applies to each factor by independently permuting the other two views.

The codebooks and decoders are trained on pooled training splits from twelve real-world domains using reconstruction, quantization, factor consistency, invariance, and event-presence supervision. Appendix F.1 gives the view operators, regularization terms, and optimization details.

3.2 Behavior-Grounded Phrase Alignment

Once the codebooks are frozen, each entry is decoded into a temporal segment. A deterministic verbalizer $\mathcal{V}_f$ describes its skeleton direction and topology, rhythm rate and strength, or event polarity and temporal support. For a descriptor ν, the compatible code set is

$$\mathcal{K}(f, \nu) = \left\{ k \in \mathcal{I}_f : \mathcal{V}_f\left(D_f(e_k^f)\right) = \nu \right\}. \tag{2}$$

Input phrases are matched to the descriptor vocabulary by semantic similarity and associated with the corresponding code sets. A phrase such as "a brief dip" can therefore match several codes with different depths or recovery profiles, rather than a single numerical trajectory.

For phrase j, its compatible set $\mathcal{K}_j$, temporal interval $I_j \subseteq [0, 1]$, and duration bounds form a semantic anchor. Candidate slots are identified by their continuous positions within I_j, using target positions during training and proposal positions at inference. The anchor set A^P contains at most one anchor per factor.

A frozen text encoder (Zhang et al., 2025) produces the full-text embedding h_c and factor-clause embeddings $\mathbf{r} = (r_s, r_r, r_e)$. These embeddings condition generation, while anchors specify code compatibility and temporal scope for supervision and constraint realization. The descriptor vocabulary and anchor construction are detailed in Appendix F.2.

3.3 Text-Conditioned Compositional Prior

Semantic anchors constrain selected features but do not determine a complete sequence. The conditional prior learns dependencies across factors and slots, providing a distribution over the remaining content needed to complete the sequence.

After codebook learning and any training-domain adaptation, the tokenizer is frozen and training sequences are encoded as target programs P. Paired descriptions supervise the prior, using descriptor-based captions when native text is unavailable. Let H_ℓ summarize h_c and the preceding slots $(p_1, \ldots, p_{\ell-1})$. The prior factorizes as

$$p_\theta(P \mid c) = \prod_{\ell=1}^{L} \Big[p_\theta(k_\ell^s \mid H_\ell, r_s)\, p_\theta(k_\ell^r \mid H_\ell, r_r, k_\ell^s) \times p_\theta(k_\ell^e \mid H_\ell, r_e, k_\ell^s, k_\ell^r)\, p_\theta(a_\ell \mid H_\ell, \mathbf{r}, k_\ell^s, k_\ell^r, k_\ell^e) \Big]. \tag{3}$$

The order $s \to r \to e \to a$ models within-slot dependencies; H_ℓ carries dependencies across slots. This prediction order is distinct from placement, which is determined by the continuous attributes.

Code cross-entropy and attribute regression supervise complete programs. Regional supervision additionally assigns soft weights to candidate slots within the requested interval, encouraging compatible codes and durations without imposing a unique phrase–slot correspondence. Its formulation is provided in Appendix F.2.

The twelve-domain models use factor-clause conditioning and regional supervision. Synth-U instead uses full-caption conditioning with $\mathbf{r} = 0$ and $A^P = \varnothing$; local events are realized through the output-space interface below. Time-MMD broadcasts the full-report embedding to all factor heads.

3.4 Proposal-Conserving Constraint Realization

At inference, a complete proposal $P^{(0)} \sim p_\theta(P \mid c)$ is sampled. Code-level constraints act before decoding, whereas waveform and numeric constraints act after scale restoration. Preservation concerns unaddressed code identities, samples outside constrained regions, or first differences, respectively.

For code-level control, the projection $P = \Pi_{A^P}(P^{(0)})$ retains compatible codes and corrects unsatisfied anchors within their scope. Each anchor changes at most one factor-code cell and, when required, its position or duration attributes; other code identities remain unchanged. Slot selection and attribute projection are specified in Appendix F.4. Without code-level anchors, $P = P^{(0)}$.

Joint decoding and resizing yield the sequence shape $y = \mathrm{Resize}_T(\mathrm{Dec}_\psi(\overline{P}))$. An MDN (Bishop, 1994), conditioned on h_c and, where enabled, pooled program features, jointly samples the overall level m and log scale, whose exponential gives $s > 0$. The decoded shape is standardized by its own mean and stabilized standard deviation, then scaled by s and shifted by m. Any domain-specific calibration or physical-support processing is fitted on training data and fixed before testing. The resulting sequence x^{raw} is returned for code-level control and serves as the proposal $x^{(0)}$ for output-space control.

For Synth-U's predefined local events, each non-null clause specifies an event type q_j and a region mask M_j. A conditional inpainter, trained on event-free examples, estimates the background within the union of these masks using the unmasked proposal and text-specified skeleton/rhythm context. Let m_0 and $s_0 > 0$ be the proposal's mean and stabilized standard deviation, and let $z^{(0)} = (x^{(0)} - m_0)/s_0$. The completed background b equals $z^{(0)}$ outside the masks and the inpainter's prediction inside them. Event realization gives

$$x_t^W = m_0 + s_0 \left(b_t + \sum\nolimits_j M_{j,t} \tau_{q_j,t} \right), \tag{4}$$

where τ_{q_j} is an analytic single-peak, double-peak, or sag waveform, separate from the codebooks, with parameters fixed on validation data. Reusing m_0 and s_0 preserves values outside the constrained regions. A NULL clause activates no mask and leaves event absence to the conditional prior.

A separate numeric-anchor experiment on Time-MMD uses a uniform offset to match a report-specified endpoint while preserving first differences. Full realization rules, waveform definitions, and scale processing are given in Appendix F.4.

4 Experiments

Our evaluation addresses two questions: whether the generated series preserve real-data distributions, and whether they satisfy local text requirements. We assess fidelity on real-world data and reports,

Table 1: Generation fidelity on twelve real-world benchmarks.

	Dataset	VQ-PACT (Ours)	ShapeLex (arXiv 2026)	TimeMAR (WWW 2026)	BRIDGE (ICML 2025)	T2S (IJCAI 2025)	TimeVQVAE (AISTATS 2023)	GT-GAN (NeurIPS 2022)	TimeVAE (arXiv 2021)	TimeGAN (NeurIPS 2019)
Marginal Distribution Distance ↓	Electricity	**0.0041** ± .001	0.0066 ± .002	0.0109 ± .002	0.2200 ± .070	0.0355 ± .005	1.7630 ± .088	2.0260 ± .280	3.3060 ± .044	2.4430 ± .765
	Solar	**41.0953** ± .748	52.1290 ± 2.037	53.0800 ± 4.210	375.5310 ± .001	50.1940 ± 2.615	466.1740 ± .145	476.1960 ± 17.041	365.9060 ± 6.365	460.8100 ± 14.078
	Wind	0.0490 ± .004	**0.0370** ± .004	0.0578 ± .005	0.3160 ± .031	0.2552 ± .024	0.7770 ± .028	0.7060 ± .106	0.9430 ± .008	1.1150 ± .159
	Traffic	**0.0211** ± .001	0.0244 ± .006	0.0246 ± .008	0.2540 ± .034	0.0957 ± .015	1.1700 ± .028	1.3110 ± .032	0.9840 ± .012	1.7330 ± .137
	Taxi	**0.0458** ± .001	0.0630 ± .009	0.0672 ± .012	0.3860 ± .057	0.0713 ± .010	0.5340 ± .032	1.1180 ± .157	0.6970 ± .007	1.2780 ± .168
	Pedestrian	**0.0397** ± .001	0.0442 ± .007	0.0477 ± .014	0.6210 ± .124	0.0875 ± .010	1.2250 ± .060	1.5590 ± .117	0.7770 ± .012	1.5740 ± .290
	Air	0.0311 ± .001	**0.0302** ± .007	0.0320 ± .010	0.4470 ± .112	0.0913 ± .017	0.3380 ± .012	2.8280 ± .172	1.3690 ± .040	2.0890 ± .618
	Temperature	**0.0729** ± .003	0.0964 ± .008	0.0978 ± .003	0.3420 ± .010	0.1441 ± .011	0.9430 ± .035	1.1650 ± .072	2.0440 ± .024	1.1640 ± .110
	Rain	0.2485 ± .003	**0.1887** ± .010	0.7141 ± .033	5.3400 ± .421	0.4361 ± .040	9.2430 ± .122	6.4730 ± 1.207	9.1340 ± .477	10.9370 ± 4.039
	NN5	0.1616 ± .003	**0.1445** ± .019	0.1711 ± .028	0.5910 ± .029	0.2155 ± .026	1.4240 ± .043	2.1210 ± .094	2.8710 ± .045	2.7580 ± .142
	Fred-MD	**0.0359** ± .010	0.0363 ± .006	0.4068 ± .011	0.2580 ± .045	0.2607 ± .037	2.9320 ± .133	4.0260 ± .087	2.9020 ± .215	4.0280 ± .130
	Exchange	**0.2690** ± .016	0.3103 ± .021	0.3165 ± .003	0.3740 ± .053	0.3492 ± .021	0.9930 ± .058	1.3550 ± .072	1.3310 ± .042	1.5530 ± .122
K-L Divergence ↓	Electricity	0.0133 ± .001	**0.0089** ± .002	0.0093 ± .002	0.0110 ± .010	0.2322 ± .024	0.1850 ± .018	0.4150 ± .040	0.5800 ± .005	0.3950 ± .121
	Solar	0.1421 ± .019	0.2328 ± .008	0.2533 ± .002	0.1700 ± .002	2.5605 ± .205	0.7260 ± .043	**0.1020** ± .045	0.2010 ± .008	0.8890 ± .288
	Wind	0.0601 ± .010	**0.0353** ± .009	0.0672 ± .005	0.0670 ± .030	3.7456 ± .450	0.4930 ± .081	0.5110 ± .129	0.5530 ± .014	4.5280 ± 1.743
	Traffic	**0.0017** ± .001	0.0109 ± .003	0.0116 ± .004	0.0130 ± .004	0.2902 ± .027	0.1450 ± .015	1.1080 ± .171	0.2120 ± .006	2.1340 ± .952
	Taxi	**0.0026** ± .001	0.0122 ± .004	0.0300 ± .009	0.0130 ± .009	0.0837 ± .013	0.1000 ± .014	0.6630 ± .127	0.1200 ± .005	1.1600 ± .651
	Pedestrian	0.0168 ± .001	0.0374 ± .003	1.4432 ± .102	**0.0110** ± .009	0.1892 ± .021	0.2750 ± .021	0.3470 ± .085	0.0520 ± .010	0.8810 ± .436
	Air	**0.0069** ± .001	0.0083 ± .001	0.0086 ± .002	0.0220 ± .017	0.1073 ± .018	0.0170 ± .004	0.5060 ± .091	0.1760 ± .016	0.5880 ± .369
	Temperature	**0.0058** ± .002	0.0188 ± .006	0.1159 ± .023	0.0230 ± .013	0.5432 ± .071	0.9800 ± .190	2.1770 ± .323	1.9100 ± .076	8.7750 ± 2.511
	Rain	0.0138 ± .001	0.0067 ± .001	0.0638 ± .011	**0.0060** ± .001	3.8021 ± .267	0.0080 ± .002	0.4620 ± .056	0.1750 ± .011	0.3830 ± .089
	NN5	0.0198 ± .002	0.0126 ± .005	0.7866 ± .006	**0.0100** ± .008	0.9707 ± .088	0.6030 ± .107	1.3720 ± .180	1.2840 ± .058	4.0540 ± 1.592
	Fred-MD	0.3223 ± .095	0.3007 ± .009	4.6959 ± .220	**0.0240** ± .019	4.1329 ± .248	0.7120 ± .054	3.5090 ± .299	0.3760 ± .025	5.3710 ± 1.455
	Exchange	1.6072 ± 1.575	**0.0725** ± .007	4.1141 ± .289	0.0830 ± .056	1.9615 ± .275	1.9840 ± .836	1.5830 ± .932	2.0110 ± .433	4.3760 ± .664
	Count (**best**/2nd)	**12**/7	**7**/14	0/1	**4**/1	0/1	0/0	**1**/0	0/0	0/0

then compare text-controlled generators under a shared realization procedure on Synth-U. Representation and inference ablations examine their respective contributions; held-out combinations and computational cost provide complementary evidence.

4.1 Experimental Setup

Datasets and baselines. Distributional fidelity is evaluated on twelve real-world univariate benchmarks from GluonTS and Monash (Alexandrov et al., 2020; Godahewa et al., 2021). Synth-U (Gu et al., 2025; Lan et al., 2026) provides controlled text descriptions and 4,000 test prompts; reports from eight Time-MMD domains (Liu et al., 2024) test generation with real text. Real-world comparisons include ShapeLex (Wei et al., 2026), TimeMAR (Xu et al., 2026), BRIDGE (Li et al., 2025), T2S (Ge et al., 2025), TimeVQVAE (Lee et al., 2023), GT-GAN (Jeon et al., 2022), TimeVAE (Desai et al., 2021), and TimeGAN (Yoon et al., 2019). Synth-U comparisons use VerbalTS (Gu et al., 2025), DiffuSETS (Lai et al., 2025), BRIDGE, and T2S; Time-MMD comparisons use ShapeLex, TimeMAR, BRIDGE, and T2S. All comparative results are obtained from local reproductions.

Metrics. Marginal fidelity is measured by $\mathrm{MDD} = (TB)^{-1} \sum_{t=1}^{T} \sum_{b=1}^{B} |h^{\mathrm{r}}_{t,b} - h^{\mathrm{g}}_{t,b}|$ and $D_{\mathrm{KL}}(p\|q) = \sum_b p_b \log(p_b/q_b)$, where T is the sequence length, $h^{\mathrm{r}}_{t,b}$ and $h^{\mathrm{g}}_{t,b}$ are histogram densities at temporal position t, B is the number of bins, and p and q are normalized histograms of pooled values (Huang et al., 2025; Li et al., 2025; Kullback & Leibler, 1951). MDD-32 and K–L-32 average these distances equally over 32 Synth-U strata defined by trend type, trend direction, and seasonal cycle ($4 \times 2 \times 4$); the suffix denotes strata, not histogram bins. Autocorrelation, spectral, and local-variation metrics additionally measure temporal fidelity. Event Joint requires correct events in all three temporal regions. We report it under both a frozen ShapeletCNN (Event-S) and an independently trained ResNet (Event-R). Six-Joint requires all three global attributes and all three regional events to be correct under the global and local ResNet evaluators. It therefore tests complete semantic satisfaction, rather than local events alone. Metric definitions, evaluator preprocessing, and additional recognizer checks are given in Appendix A.3.

Implementation and reporting. VQ-PACT uses 64 slots and codebook sizes $16/16/8$. Real-world generation uses grammar anchors, whereas Synth-U realizes regional events after decoding full-caption proposals (Appendix F). The primary Synth-U comparison uses one checkpoint and ten stochastic generation draws per method. Each draw contains one length-128 sequence for each of the 4,000 test prompts. All generators use the same frozen executor; metrics are computed separately for each draw and then averaged in Table 3. No waveform averaging or best-of-ten selection is used in this comparison. The separate native-pipeline evaluation, codebook comparisons, and inference ablations retain their own repetition and execution protocols (Appendix A.1); their results are not pooled with this comparison. Sampling variation is distinguished from variation across independently trained generators. Bold and underlining indicate the best and second-best means, respectively, not statistical significance.

Table 2: Time-MMD fidelity. Entries report MDD / K–L; lower is better for both.

Domain	VQ-PACT	ShapeLex	TimeMAR	BRIDGE	T2S
Energy-MMD	**0.1268 / 0.1648**	0.1283 / 0.1671	0.1318 / 0.2908	0.1548 / 0.4275	0.1407 / 0.2829
Climate-MMD	0.0568 / **0.0575**	**0.0564** / 0.0582	0.0589 / 0.0788	0.1988 / 0.9980	0.1205 / 0.2677
Economy-MMD	**0.1062 / 0.2345**	0.1075 / 0.2382	0.3998 / 1.3057	0.2436 / 1.8607	0.1713 / 0.4759
Agriculture-MMD	**0.3698** / 0.9125	0.3722 / **0.9085**	0.4608 / 1.4543	0.4746 / 1.5520	0.4186 / 1.8734
Health-Africa-MMD	**0.1105 / 0.2515**	0.1117 / 0.2533	0.2017 / 1.3248	0.2488 / 1.4601	0.2233 / 0.5821
Health-US-MMD	0.0762 / **0.1205**	0.0766 / 0.1217	**0.0448** / 0.1304	0.1924 / 1.1310	0.1629 / 0.2036
Social-Good-MMD	**0.1038 / 0.1525**	0.1049 / 0.1539	0.1220 / 0.2192	0.2088 / 1.2840	0.1621 / 0.3125
Traffic-MMD	**0.1308** / 0.4185	0.1322 / **0.4150**	0.3005 / 0.6271	0.2343 / 1.9360	0.2612 / 0.8942

4.2 Distributional Fidelity

Generation across real-world domains. VQ-PACT ranks first in 12 and second in 7 of the 24 MDD/K–L comparisons (Table 1). It obtains the lowest MDD on eight datasets, including an 18.1% reduction on Solar relative to the strongest baseline, and the lowest K–L divergence on four datasets. The advantage is not uniform: ShapeLex performs better on Wind and Rain, and several baselines attain lower K–L divergence on other domains. The results therefore support broad improvements across heterogeneous datasets rather than dominance on every distribution.

Marginal agreement alone does not characterize temporal dependence. We therefore compare independently standardized windows with ShapeLex and TimeMAR. VQ-PACT obtains the best average ranks for autocorrelation distance (1.500), spectral divergence (1.667), first-difference distance (1.583), and roughness gap (1.333). This provides evidence of temporal fidelity beyond matching value histograms. Per-domain results, coverage diagnostics, and sample-count analyses are reported in Appendix B.1.

Generation from real reports. On Time-MMD, VQ-PACT achieves the lowest MDD in six domains and the lowest K–L divergence in six domains, ranking first in 12 of the 16 comparisons and second in the remaining four (Table 2). These comparisons extend the fidelity evaluation to conditioning on real reports. The small differences from ShapeLex in several domains should be interpreted as differences in reported means, since uncertainty estimates are not available for this table.

A separate closed-book diagnostic tests the response to explicit numeric facts, with target windows, target statistics, and cached target representations excluded from inference. On reports containing parseable values, generated-mean/original-anchor correlations range from 0.8071 to 0.9187 across three domains and fall close to zero after shuffling the reports. This verifies the response of the explicit numeric realization path, rather than unrestricted language understanding. The eligible subsets and full results are given in Appendix B.2.

4.3 Fine-Grained Text Control

Comparison under a shared executor. To control for differences in local realization, we apply the same frozen waveform executor to VQ-PACT, VerbalTS, DiffuSETS, BRIDGE, and T2S on the same 4,000 Synth-U prompts. The executor uses identical event parameters and text-derived regional constraints for all methods, without generator-specific tuning. It receives generated proposals and text, but no target sequences or target statistics. The common configuration is specified in Appendix A.1.

VQ-PACT obtains the best reported mean on all five metrics in Table 3. Its MDD-32 and K–L-32 are 0.1623 and 0.0621, compared with 0.1763 and 0.1142 for VerbalTS. Event-S reaches 85.08%, exceeding the strongest baseline under that evaluator by 6.12 percentage points; Event-R reaches 71.65%, compared with 70.56% for T2S. Six-Joint reaches 66.86%, exceeding VerbalTS (60.30%) by 6.56 percentage points. The gains therefore extend beyond local-event recognition to simultaneous satisfaction of global and regional requirements. Together, these results show better distributional fidelity and semantic control under the same realization procedure.

Table 3: Synth-U generation with a shared frozen executor.

Method	Distributional fidelity		Event Joint (%)		Complete semantics
	MDD-32 ↓	K–L-32 ↓	Event-S ↑	Event-R ↑	Six-Joint (%) ↑
VQ-PACT (Ours)	**0.1623**	**0.0621**	**85.08**	**71.65**	**66.86**
VerbalTS	0.1763	0.1142	78.96	68.73	60.30
DiffuSETS	0.1985	0.1448	59.15	52.03	24.94
BRIDGE	0.2749	2.1762	30.68	45.39	18.41
T2S	0.2985	0.9091	15.13	70.56	23.50

Means of per-draw metrics over ten draws from one checkpoint per method; each draw covers all 4,000 test prompts. Event-S and Event-R use ShapeletCNN and local ResNet, respectively; Six-Joint uses the global and local ResNet evaluators. All methods share the same frozen executor and evaluation protocol.

The two event evaluators differ substantially for some baselines, so both are reported alongside Six-Joint. These comparisons concern performance under the evaluation protocol; ten draws from one checkpoint do not measure variability across training runs. The native-pipeline comparison provides an end-to-end evaluation, with scores and evaluator checks in Appendices C.1 and C.3.

Scope of a local text edit. Figure 4 illustrates a separate matched-proposal intervention: with the proposal and other clauses fixed, the middle event is changed from a double peak to a sag. All three evaluators recognize both events, and over 99.99% of the absolute output change falls within the requested interval. This selected successful case illustrates the locality of the realization mechanism, not a dataset-wide edit-success rate. The selection rule and additional selected and random examples are provided in Appendices C.5 and C.

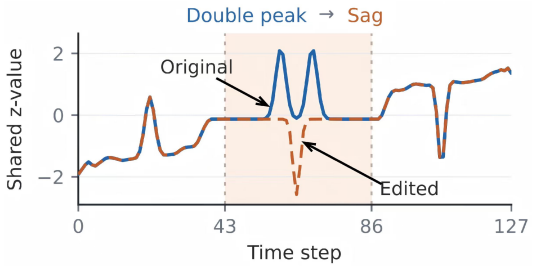


Figure 4: A selected local edit. Shading marks the edited interval; both curves share normalization.

Regional control with native generation pipelines. Figure 5 compares beginning-, middle-, and end-region event accuracy in a separate experiment on the same 4,000 Synth-U prompts. Each method uses its native generation pipeline, and all outputs are evaluated by the same frozen ShapeletCNN. VQ-PACT reaches 99.49%, 97.39%, and 96.07% regional accuracy, respectively, with 93.10% joint accuracy across the three regions. These results characterize end-to-end system performance; unlike Table 3, they do not hold the realization procedure fixed. Full scores and the evaluation protocol are provided in Appendix C.1.

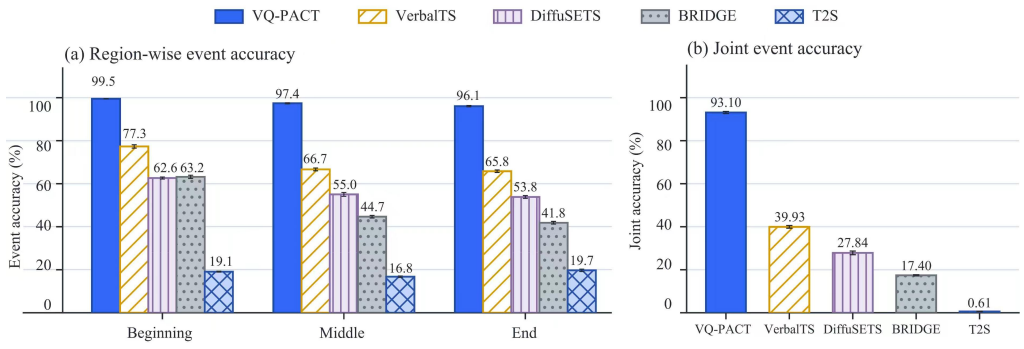


Figure 5: Synth-U local-event control with native generation pipelines: (a) regional accuracy; (b) joint accuracy. All methods use the same frozen ShapeletCNN evaluator. Error bars show population SD across generator-seed means for VQ-PACT and sampling runs for baselines (Appendix A.1).

4.4 Analysis of Representation and Realization

Factor-specific versus unified codebooks. A separate comparison evaluates a single 256-entry codebook and three codebooks of sizes $16/16/8$, both with and without selective execution (Table S1 in Appendix A.2). Without execution, three codebooks reduce MDD-32/K–L-32 from $0.1842/0.0587$ to $0.1730/0.0519$. With the same execution configuration enabled for both variants, the corresponding distances decrease from $0.1881/0.0612$ to $0.1795/0.0539$, and Six-Joint rises from 52.15% to 60.70%, an increase of 8.55 percentage points. The improvements in both settings support a contribution from the factor-specific representation configuration beyond selective execution. This is not a capacity-matched comparison: the three codebooks store 40 entries but admit 2,048 discrete tuples. Code usage and one-slot interventions provide additional diagnostics in Appendix D.1.

Contribution of selective realization. Fixed-proposal ablations examine the inference operations separately (Appendix D.2). Under the native realization configuration, removing background neutralization lowers Shapelet Event Joint from 93.10% to 80.30%. Removing full clause coverage or correct phrase–region binding lowers it to 22.24% and 8.16%, respectively. A shared event prototype yields 10.23%, showing that specifying where to intervene is insufficient without specifying the event type. These are inference-time interventions, not retraining ablations of the representation or alignment objective. They use a different execution configuration from Table 3, so their scores are compared only within the ablation protocol. Complete tables and the control–fidelity sweep are retained in Appendices D.2 and D.3.

4.5 Additional Analyses

Generalization to held-out combinations. Factor-OOD excludes 32 of 128 skeleton–rhythm–middle-event combinations while retaining all individual factor values and pairwise combinations; Temporal-OOD excludes the beginning–sag pairing while retaining sag at other locations. After retraining on each restricted split, Factor-OOD Six-Joint decreases from 61.05% on ID data to 58.12% on held-out combinations, a drop of 2.93 percentage points. Temporal-OOD decreases from 64.51% to 53.72%, a drop of 10.79 percentage points. VQ-PACT thus retains complete-constraint accuracy on both held-out sets, with a larger reduction under the tested temporal shift. These within-model results assess transfer to unseen combinations and locations separately; they do not establish superiority over other generators on these splits. Each protocol uses one training seed and ten draws. Split construction and evaluation definitions are provided in Appendix E.

Computational cost. A separate timing study measures each method's native generation path, rather than the shared-executor comparison above. On an RTX 3090 with sequence length 128, VQ-PACT achieves the highest batch-128 throughput among the evaluated methods (460.2 ± 2.9 series/s) and the second-lowest single-example latency (117.9 ± 0.7 ms). T2S is faster for individual sequences. The timing includes generation, decoding, and refinement, but excludes the shared text encoder. Parameter counts, peak memory, and the full measurement protocol are reported in Appendix D.5. These measurements characterize practical generation cost under the recorded configurations, not asymptotic time or space complexity.

5 Conclusion

We presented VQ-PACT for text-controlled time series generation, addressing the challenge of reconciling fine-grained semantic control with distributional fidelity. By representing morphological features as reusable discrete units across factorized codebooks, VQ-PACT separates overall trends, recurring fluctuations, and local transients. Behavior-grounded alignment connects textual phrases to codebook features through their decoded temporal shapes, while proposal-conserving realization constrains only text-specified aspects. Experiments across twelve real-world benchmarks, controlled Synth-U settings, and eight Time-MMD domains demonstrate improved distributional fidelity and fine-grained control, with evidence of generalization to unseen feature combinations. This compositional framework establishes explicit, testable connections between natural language requirements and observable temporal dynamics for controllable sequence generation.

Declaration of Generative AI Use

Generative AI tools were used to assist with English-language editing and Chinese translation of the manuscript. The authors reviewed and revised all AI-assisted content and take full responsibility for the final manuscript.

References

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. GluonTS: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

Christopher M. Bishop. Mixture density networks. Technical Report NCRG/94/004, Aston University, 1994.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. TimeVAE: A variational auto-encoder for multivariate time series generation, 2021.

Yunfeng Ge, Jiawei Li, Yiji Zhao, Haomin Wen, Zhao Li, Meikang Qiu, Hongyan Li, Ming Jin, and Shirui Pan. T2S: High-resolution time series generation with text-to-series diffusion models. In *IJCAI*, pp. 5208–5216. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/580.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *NeurIPS Datasets and Benchmarks*, 2021.

Shuqi Gu, Chuyue Li, Baoyu Jing, and Kan Ren. VerbalTS: Generating time series from texts. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20448–20476. PMLR, 2025.

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 *AAAI*, volume 39, pp. 17520–17527, 2025. doi: 10.1609/aaai.v39i17.33926.

Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. GT-GAN: General purpose time series synthesis with generative adversarial networks. In *NeurIPS*, volume 35, pp. 36999–37010. Curran Associates, Inc., 2022. doi: 10.52202/068431-2681.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. Truth-conditional captions for time series data. In *EMNLP*, pp. 719–733. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.55.

Baoyu Jing, Shuqi Gu, Tianyu Chen, Zhiyu Yang, Dongsheng Li, Jingrui He, and Kan Ren. Towards editing time series. In *NeurIPS*, volume 37, pp. 37561–37593. Curran Associates, Inc., 2024. doi: 10.52202/079017-1186.

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.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, volume 32, 2019.

Yongfan Lai, Jiabo Chen, Qinghao Zhao, Deyun Zhang, Yue Wang, Shijia Geng, Hongyan Li, and Shenda Hong. DiffuSETS: 12-lead ECG generation conditioned on clinical text reports and patient-specific information. *Patterns*, 6(10):101291, 2025. doi: 10.1016/j.patter.2025.101291.

Shaocheng Lan, Shuqi Gu, Zhangzhi Xiong, and Kan Ren. ConTSG-Bench: A unified benchmark for conditional time series generation, 2026. Accepted at ICML 2026; update to the proceedings record when available.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In *AISTATS*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7665–7693. PMLR, 2023.

Hao Li, Yu-Hao Huang, Chang Xu, Viktor Schlegel, Renhe Jiang, Riza Batista-Navarro, Goran Nenadic, and Jiang Bian. BRIDGE: Bootstrapping text to control time-series generation via multi-agent iterative optimization and diffusion modeling. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34742–34773. PMLR, 2025.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-MMD: Multi-domain multimodal dataset for time series analysis. In *NeurIPS*, volume 37, pp. 77888–77933. Curran Associates, Inc., 2024. doi: 10.52202/079017-2476.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7176–7185. PMLR, 2020.

Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep P. Chinchali. Time weaver: A conditional time series generation model. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37293–37320. PMLR, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, pp. 8024–8035, 2019.

Jiaxing Qiu, Dongliang Guo, Brynne Sullivan, Teague R. Henry, and Thomas Hartvigsen. Instruction-based time series editing. In *KDD*, pp. 1216–1227. Association for Computing Machinery, 2026. doi: 10.1145/3770854.3780299.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, volume 30, pp. 6306–6315, 2017.

Subo Wei, Jianqi Gao, Mingyan Fan, Shaorong Xie, Xinzhi Wang, and Yongpeng Dong. ShapeLex: Decoupling local shape symbolization and global scale modeling for text-controlled time series generation, 2026.

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 *NeurIPS*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-2929.

Xiangyu Xu, Qingsong Zhong, and Jilin Hu. TimeMar: Multi-scale autoregressive modeling for unconditional time series generation. In *WWW*, pp. 5132–5143. Association for Computing Machinery, 2026. doi: 10.1145/3774904.3792173.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *NeurIPS*, volume 32, pp. 5508–5518. Curran Associates, Inc., 2019.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *ICLR*, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. doi: 10.48550/arXiv.2506.05176.

A Experimental Details

Data and model configuration. The twelve GluonTS/Monash benchmarks use univariate length-96 windows; Synth-U uses length-128 sequences and the official 4,000-prompt test split. Time-MMD covers eight domains, with the comparative fidelity results and separate numeric-response analysis described in Appendix B.2. VQ-PACT is implemented in PyTorch (Paszke et al., 2019) with a frozen Qwen3-Embedding-0.6B encoder (Zhang et al., 2025), 64 program slots, and skeleton/rhythm/event codebook sizes $16/16/8$. Computational-efficiency measurements use an RTX 3090. The training and realization configurations are given in Appendix F.

A.1 Evaluation regimes and result provenance

The experiments below use distinct generation and uncertainty protocols; their numerical results are not interchangeable.

Matched-executor main comparison. Table 3 reports mean scores for the common-executor configuration. Each method uses one seed-0 checkpoint and ten generation draws on the same 4,000 official Synth-U prompts. A draw is a complete generated dataset containing one length-128 sequence per prompt, giving 40,000 generated sequences per checkpoint. Each metric is computed on each draw separately, and the ten metric values are averaged. This procedure neither averages waveforms nor selects a best candidate for each prompt. All methods use the same frozen conditional inpainter, the same caption parser, and partial-inpaint realization: all event types have amplitude 1.3 and width 0.025, with double-peak separation 0.22. Execution uses generated proposals and parsed captions, not real test windows or their statistics; real windows are used only during evaluation. Only means are reported for this main table; uncertainty estimates from the earlier experiments below are not attached to its scores. Ten draws from one checkpoint do not measure independent training-seed variation.

Earlier native-system and inference results. Table S6, the regional-control plot, evaluator transfer, and the fixed-proposal inference ablations retain the earlier native-system evaluation. VQ-PACT uses three independently trained autoregressive generators that share one frozen adapted seed-0 tokenizer. Ten draws are averaged within each generator seed before computing the mean and population standard deviation (SD) across the three seed means. This SD excludes independent tokenizer-training variation. The earlier baseline evaluations use ten draws from one seed-0 checkpoint without the added common executor, so their deviations describe sampling variability; the regional-control plot uses population SD across these draws. These native-system results, including VQ-PACT's 93.10% Shapelet Event Joint, do not describe the matched-executor main-table configuration.

Other evaluation regimes. On the twelve real-world domains, VQ-PACT uses one fixed trained model per domain and reports sample SDs across three generation seeds. Text-to-program robustness uses sample SD across three generator seeds. Strict holdouts use sample SD over ten draws from one training seed. Timing uncertainty is population SD across three measurement blocks. The separate three-seed, five-run codebook comparison is described in Appendix A.2; its supplied error terms do not inherit the uncertainty definition of either the earlier native evaluation or the matched-executor main comparison.

Shared-pipeline real and generated windows use matching normalization, sample counts, and metric implementations within each domain. Cross-domain summaries use ranks and best/second-best counts, not a raw mean of MDD across heterogeneous domains. Synth-U methods use the official test order and share frozen evaluators for each metric, without evaluator-based candidate selection. SDs are not confidence intervals.

The shared ShapeletCNN reaches 99.20% Event Joint on real Synth-U test series. In the earlier native-system evaluator-transfer experiment, independent ResNet and ExtraTrees recognizers are trained on real training data and evaluated on the same generated arrays within that experiment. Appendix C.3 reports those control metrics; they are not additional measurements of the new matched-executor main-table outputs.

Table S1: Codebook comparison on Synth-U. Joint scores are percentages.

Representation	Selective execution	MDD-32 ↓	K–L-32 ↓	Event Joint ↑	Six-Joint ↑
Single codebook	Disabled	0.1842 ± 0.008	0.0587 ± 0.004	10.45 ± 1.2	3.68 ± 0.9
Three codebooks	Disabled	$\mathbf{0.1730 \pm 0.006}$	$\mathbf{0.0519 \pm 0.003}$	$\mathbf{11.82 \pm 1.1}$	$\mathbf{4.14 \pm 0.8}$
Single codebook	Enabled	0.1881 ± 0.009	0.0612 ± 0.005	68.23 ± 2.8	52.15 ± 2.3
Three codebooks	Enabled	$\mathbf{0.1795 \pm 0.007}$	$\mathbf{0.0539 \pm 0.004}$	$\mathbf{72.57 \pm 2.5}$	$\mathbf{60.70 \pm 2.1}$

A.2 Codebook comparison protocol

Table S1 compares two representation configurations on Synth-U, both before and after selective execution. The single codebook contains 256 entries; the factor-specific configuration uses $16/16/8$ entries for skeleton, rhythm, and event, respectively. Thus, 40 stored entries form 2,048 possible tuples. The supplied configuration summaries list approximately 8.2M and 8.5M parameters. Their component-counting boundary is not specified, so these counts are not equated with the active/loaded parameter counts in Appendix D.5. The comparison does not match stored-entry counts or discrete combination capacity, and evaluates the representation configurations as a whole.

Each configuration is trained for 200 epochs under three training seeds, with five generation runs per seed. These are 15 generation runs, not 15 independent trained models or 15 individual sequences. When execution is enabled, the executor parameters are shared between the two configurations. This comparison is distinct from both the earlier three-seed, ten-draw native evaluation and the matched-executor main comparison in Appendix A.1. Numerical comparisons are restricted to the configurations within Table S1.

Event Joint and Six-Joint retain the definitions and error terms of the separately reported experiment. The available summary does not identify its event evaluator or whether the error terms aggregate seed means or all generation runs. We therefore interpret the mean differences within this comparison, without identifying its event scores with Event-S or Event-R in the main table or using its error terms to infer statistical significance or training-seed stability.

A.3 Metric definitions

Following TimeDP and BRIDGE (Huang et al., 2025; Li et al., 2025), let X^{r} and X^{g} contain N real and generated windows of length T. In the shared evaluation implementation, both are transformed using the same held-out real-set mean μ_{r} and standard deviation σ_{r}, $\widetilde{X}^{\mathrm{r}} = (X^{\mathrm{r}} - \mu_{\mathrm{r}})/\sigma_{\mathrm{r}}$ and $\widetilde{X}^{\mathrm{g}} = (X^{\mathrm{g}} - \mu_{\mathrm{r}})/\sigma_{\mathrm{r}}$. At each temporal position t, the real values define $B{=}20$ equal-width bins, and $h^{\mathrm{r}}_{t,b}$ and $h^{\mathrm{g}}_{t,b}$ denote the corresponding density estimates. The reported marginal-distribution distance is $\mathrm{MDD} = (TB)^{-1} \sum_{t=1}^{T} \sum_{b=1}^{B} |h^{\mathrm{r}}_{t,b} - h^{\mathrm{g}}_{t,b}|$. For K–L, the normalized values are flattened and the real values define a 50-bin histogram; the generated histogram uses the same bin edges. After adding $\epsilon = 10^{-9}$ to the generated histogram and normalizing both histograms, we compute (Kullback & Leibler, 1951) $D_{\mathrm{KL}}(p\|q) = \sum_{b=1}^{50} p_b \log(p_b/q_b)$.

For the matched-executor Synth-U comparison, the 32 strata are the Cartesian product of four trend types, two trend directions, and four seasonal-cycle classes. Within each stratum, real and generated windows are normalized using that stratum's real evaluation mean and standard deviation (with 10^{-8} added to the denominator). The distances above use 20 bins per temporal position for MDD and 50 bins on pooled values for K–L. Stratum scores receive equal weight, regardless of stratum size. For each generation draw, we average the 32 stratum scores; the main table then averages these per-draw values over ten draws. Thus, the suffix "32" specifies the number of strata, not the number of histogram bins. Unstratified MDD/K–L results under the other protocols are not numerically interchangeable with MDD-32/K–L-32.

Held-out real statistics are used for evaluation only, not supplied to the generator during closed-book inference. To characterize temporal shape separately from global scale, the temporal audit standardizes each window independently. Let $\bar{\rho}^{\mathrm{r}}_{\ell}$ and $\bar{\rho}^{\mathrm{g}}_{\ell}$ be the mean sample autocorrelations at lag ℓ. We use $\mathrm{ACF\text{-}L1} = 24^{-1} \sum_{\ell=1}^{24} |\bar{\rho}^{\mathrm{r}}_{\ell} - \bar{\rho}^{\mathrm{g}}_{\ell}|$. The normalized non-DC periodograms are averaged across windows and compared by the Jensen–Shannon divergence. First-difference distance is the L1 gap

between 19 quantiles from 0.05 to 0.95, normalized by the real 0.95–0.05 range. Roughness is the absolute log-ratio of the mean first-difference RMS values, and turning discrepancy is the absolute gap between rates of consecutive first differences with opposite signs. Lower is better for all five dynamic distances.

To quantify whether each textual event is realized at its requested temporal position, we follow the evaluator-based Synth-U protocol of VerbalTS and ConTSG-Bench (Gu et al., 2025; Lan et al., 2026) and define $\mathcal{R} = \{b, m, e\}$ for the beginning, middle, and end regions, and $\mathcal{C} = \{\text{NULL}, \text{single peak}, \text{double peak}, \text{sag}\}$. For sample i and region r, let $y_{i,r}$ be the requested event label and $\hat{y}_{i,r}$ the frozen-evaluator prediction. The three metrics below are proportions; we multiply them by 100 when reporting percentages. The indicator $\mathbf{1}[\cdot]$ is one when its condition holds and zero otherwise:

$$\text{Joint} = \frac{1}{N}\sum_{i=1}^{N}\prod_{r\in\mathcal{R}}\mathbf{1}[\hat{y}_{i,r} = y_{i,r}],$$

$$\text{Segment} = \frac{1}{N|\mathcal{R}|}\sum_{i=1}^{N}\sum_{r\in\mathcal{R}}\mathbf{1}[\hat{y}_{i,r} = y_{i,r}],$$

$$\text{MacroRecall} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{\sum_{i,r}\mathbf{1}[y_{i,r} = c \wedge \hat{y}_{i,r} = c]}{\sum_{i,r}\mathbf{1}[y_{i,r} = c]}.$$

Under the shared ShapeletCNN protocol, a region is correct only when both its peak-count and sag-presence predictions match the description. Higher values indicate stronger event control. The same definition of regional correctness is used for the regional plot and joint score.

The length-128 sequence is split into zero-indexed regions $[0, 43)$, $[43, 86)$, and $[86, 128)$. The last region is padded to length 43 by repeating its final value. ShapeletCNN receives these segments without per-region standardization and predicts peak count (0, 1, or 2) and sag presence (0 or 1). The local ResNet instead standardizes each segment by its own mean and population SD, with the SD clamped below at 10^{-5}, and predicts one of four event classes: NULL, single peak, double peak, or sag. Event-S and Event-R are the fractions of sequences whose three regions are all correct under the respective evaluator. Evaluator parameters are fixed during generation evaluation.

Global Joint requires the requested trend type, trend direction, and seasonal cycle to be correct simultaneously under a frozen global ResNet evaluator. All-factor+event Joint (Six-Joint) requires these three attributes and the three regional events to be correct in the same sample; its event predictions come from the independent local ResNet, not ShapeletCNN. Thus, Six-Joint intersects Global Joint with ResNet Event Joint rather than ShapeletCNN Event Joint. The intersection is computed for each sequence before averaging; it is not the product of aggregate global and event accuracies. The two event scores retain their evaluator-specific definitions in both the earlier native results and the matched-executor comparison. The strict holdout experiments also use the local ResNet for Event Joint.

For the paired projection audit, let $\Delta_i = x_i^{\text{projected}} - x_i^{\text{proposal}}$ and let M_i mark the non-null event regions requested by the original caption. Edit locality is the fraction of absolute change inside M_i, $100\sum_t M_{i,t}|\Delta_{i,t}| / \sum_t |\Delta_{i,t}|$. Inactive-edit MAE averages $|\Delta_{i,t}|$ where $M_{i,t} = 0$. A sample without an edit has no defined localization ratio; the pre-projection row consequently does not receive a locality score. For phrase-edit case studies, the difference is instead between the edited and original outputs with the proposal held fixed, and the mask marks the edited phrase's target interval. This differs from the caption-active-region mask used in the proposal-to-realization audit.

A.4 Validation and reproducibility

All event-template amplitudes, parsing rules, and operating points are locked using training or validation data. Test labels are used only for final evaluation. The reproducibility records contain configurations, generated arrays, per-draw measurements, figures, and source manifests. The strict OOD rows in this appendix use one training seed and ten stochastic draws, as specified above.

B Full Fidelity and Real-Report Results

B.1 Dynamic fidelity on twelve real-world domains

The following tables compare VQ-PACT, ShapeLex, and TimeMAR. Deviations are generation-seed SDs for VQ-PACT and matched-subsample SDs for the baselines. PRDC uses $k = 5$ and at most 1,000 real and generated windows per repetition in the fixed ACF–PSD–difference feature space (Kynkäänniemi et al., 2019; Naeem et al., 2020).

To test whether marginal agreement also reflects temporal structure, we complement Table 1 with dynamic metrics on per-window-standardized sequences. VQ-PACT has the best average rank for ACF-L1, spectral JS, first-difference distance, and roughness gap, although the turning-point and PRDC results expose remaining domain-specific weaknesses. Tables S2– S4 provide all per-domain values.

Autocorrelation and spectral structure. Table S2 separates autocorrelation fidelity from frequency-domain fidelity. VQ-PACT is strongest on both views for Wind, Traffic, Temperature, and Exchange, while Electricity, Pedestrian, Rain, and NN5 show that no method dominates every domain. The relevant conclusion is therefore broad improvement across heterogeneous dynamics rather than a uniform per-domain win.

Table S2: Autocorrelation and spectral fidelity after per-window standardization.

Dataset	VQ-PACT		ShapeLex		TimeMAR	
	ACF-L1 ↓	PSD-JS ↓	ACF-L1 ↓	PSD-JS ↓	ACF-L1 ↓	PSD-JS ↓
Electricity	0.0536±.001	0.0153±.001	**0.0503±.002**	**0.0134±.001**	0.2304±.004	0.1465±.004
Solar	**0.0254±.002**	0.0048±.001	0.0399±.002	0.0053±.001	0.0343±.002	**0.0039±.001**
Wind	**0.0363±.004**	**0.0021±.001**	0.1493±.006	0.0137±.001	0.0626±.002	0.0030±.001
Traffic	**0.0151±.001**	**0.0010±.001**	0.0199±.001	0.0079±.001	0.0379±.001	0.0044±.001
Taxi	0.1085±.001	0.0433±.001	0.1346±.001	**0.0389±.001**	**0.1069±.003**	0.0647±.001
Pedestrian	0.0862±.002	0.0364±.001	0.0882±.002	0.0174±.001	**0.0461±.001**	**0.0063±.001**
Air	0.0898±.002	**0.0186±.001**	0.1392±.001	0.0188±.001	**0.0610±.001**	0.0207±.001
Temperature	**0.0320±.003**	**0.0139±.001**	0.3040±.006	0.0899±.002	0.1220±.006	0.1098±.001
Rain	0.0258±.002	0.0109±.001	**0.0180±.002**	**0.0044±.001**	0.3020±.008	0.2124±.003
NN5	0.2788±.004	0.1785±.003	**0.2254±.006**	**0.1373±.003**	0.4049±.003	0.3382±.003
Fred-MD	**0.0922±.005**	0.0272±.001	0.1731±.007	**0.0121±.002**	0.3766±.000	0.1317±.000
Exchange	**0.1106±.008**	**0.0084±.003**	0.2151±.037	0.0305±.008	0.2127±.016	0.0460±.008

Local variation and turning points. Table S3 measures first-difference quantiles, roughness, and turning-point rates, which respond to short-range geometry that MDD and K-L do not capture. VQ-PACT is consistently strong on first differences and roughness, whereas the turning-gap column is more mixed. This distinction prevents a low marginal distance from being interpreted as complete recovery of every local dynamic. Here Δ-Q is normalized first-difference quantile distance; roughness is the absolute log ratio, and turning gap is the absolute difference in turning-point rates (Appendix A.3).

Support coverage. Table S4 evaluates precision, recall, density, and coverage in the fixed dynamic-feature space. VQ-PACT improves recall or coverage on several domains, including Electricity and Exchange, but Solar, Wind, and NN5 remain difficult. PRDC is consequently used as a diversity guardrail rather than as a replacement for the fidelity metrics.

Dynamic profiles and sample-count convergence. Figure S1 plots the standardized dynamic views behind the preceding tables. Agreement of the profiles indicates that the gains are not produced solely by matching raw amplitude, while visible domain-level deviations explain why the PRDC and turning-point results are not uniform.

Table S3: First-difference, roughness, and turning-point fidelity.

Dataset	VQ-PACT			ShapeLex			TimeMAR		
	Δ-Q ↓	Roughness gap ↓	Turning gap ↓	Δ-Q ↓	Roughness gap ↓	Turning gap ↓	Δ-Q ↓	Roughness gap ↓	Turning gap ↓
Electricity	**0.0160±.001**	**0.0278±.004**	**0.0440±.002**	0.0243±.001	0.2565±.003	0.1304±.001	0.0500±.001	0.5662±.008	0.1168±.002
Solar	**0.0095±.001**	**0.0431±.002**	0.2647±.001	0.0184±.001	0.1272±.001	0.2395±.002	0.0324±.001	0.1205±.003	**0.2190±.002**
Wind	1.9313±.019	**0.0777±.011**	0.4499±.002	**0.8669±.017**	0.7770±.018	0.3774±.001	1.4542±.004	0.2373±.011	**0.2426±.002**
Traffic	0.0125±.001	**0.0035±.003**	0.0272±.001	**0.0082±.001**	0.1572±.001	0.0794±.001	0.0087±.001	0.0631±.002	**0.0248±.001**
Taxi	0.0726±.001	**0.1594±.001**	0.2005±.002	0.1108±.001	0.6275±.005	**0.0229±.001**	**0.0619±.001**	0.4177±.001	0.0322±.001
Pedestrian	0.0115±.001	0.1870±.002	0.0174±.002	0.0204±.001	0.1855±.005	0.0422±.001	**0.0076±.001**	**0.0404±.005**	**0.0017±.002**
Air	**0.0186±.001**	**0.2289±.005**	0.1827±.002	0.0578±.001	0.3664±.002	**0.0603±.001**	0.0212±.001	0.2848±.003	0.1311±.002
Temperature	**0.0542±.002**	**0.0669±.010**	**0.0795±.002**	0.1090±.002	0.7070±.013	0.1195±.002	0.1148±.002	0.9123±.009	0.2143±.002
Rain	**0.0281±.001**	0.0660±.003	0.2169±.002	0.0786±.002	**0.0242±.006**	0.2018±.003	0.0533±.001	1.0534±.011	**0.1479±.002**
NN5	**0.0465±.002**	**0.0102±.005**	0.0734±.002	0.0510±.001	0.0143±.005	**0.0217±.002**	0.1360±.002	0.6068±.015	0.0485±.002
Fred-MD	0.1607±.012	0.7250±.026	0.1688±.004	**0.0356±.006**	**0.3904±.049**	**0.0450±.007**	0.1905±.000	1.2718±.000	0.3223±.000
Exchange	**0.0282±.003**	**0.0244±.028**	0.0301±.012	0.0832±.008	0.5828±.095	**0.0160±.005**	0.0702±.001	0.5473±.062	0.1388±.008

Table S4: Precision, recall, density, and coverage in dynamic-feature space.

Dataset	VQ-PACT				ShapeLex				TimeMAR			
	Prec. ↑	Rec. ↑	Dens. ↑	Cov. ↑	Prec. ↑	Rec. ↑	Dens. ↑	Cov. ↑	Prec. ↑	Rec. ↑	Dens. ↑	Cov. ↑
Electricity	0.8677±.037	**0.4233±.059**	0.6431±.060	**0.4630±.041**	0.9190±.020	0.1787±.022	0.7300±.063	0.3317±.014	**0.9747±.011**	0.0783±.004	**0.8914±.113**	0.1780±.010
Solar	0.0838±.009	**0.0257±.026**	0.0238±.002	0.0038±.001	**0.5885±.009**	0.0000±.000	**0.1663±.005**	**0.0042±.000**	0.1863±.014	0.0028±.004	0.0389±.003	0.0028±.001
Wind	**0.0000±.000**	0.0000±.000	**0.0000±.000**	**0.0000±.000**	**0.0000±.000**	0.0000±.000	**0.0000±.000**	**0.0000±.000**	**0.0000±.000**	**0.3307±.573**	**0.0000±.000**	**0.0000±.000**
Traffic	0.8240±.009	0.6740±.012	0.5971±.011	0.6317±.017	**0.8803±.013**	0.3093±.030	**0.6744±.048**	0.5013±.018	0.8330±.012	**0.7970±.028**	0.6070±.038	**0.6500±.008**
Taxi	0.2957±.044	0.2507±.007	0.1025±.017	0.1330±.005	**0.7070±.043**	**0.5310±.060**	**0.3922±.017**	**0.3843±.054**	0.5103±.005	0.3517±.015	0.2301±.009	0.2350±.010
Pedestrian	0.4630±.053	0.5890±.026	0.1568±.020	0.1810±.002	0.5340±.078	0.3327±.022	0.2194±.041	0.1623±.012	**0.6450±.008**	**0.6690±.028**	**0.2993±.028**	**0.3563±.028**
Air	0.7393±.017	**0.6890±.036**	0.4462±.012	0.4263±.005	**0.8900±.013**	0.6170±.023	**0.7680±.027**	**0.5837±.025**	0.6967±.026	0.2803±.025	0.3594±.021	0.2663±.011
Temperature	0.7718±.011	0.4576±.065	0.2992±.002	**0.0723±.006**	**0.9390±.009**	**0.4960±.239**	**0.5290±.014**	0.0610±.011	0.8384±.008	0.0045±.002	0.2667±.006	0.0316±.006
Rain	0.4508±.028	**0.2226±.032**	0.2106±.019	0.2452±.016	**0.7910±.011**	0.1661±.011	**0.7460±.090**	**0.2881±.006**	0.0000±.000	0.0000±.000	0.0000±.000	0.0000±.000
NN5	0.0060±.006	0.0270±.010	0.0012±.002	0.0060±.006	0.0210±.019	**0.5375±.234**	0.0054±.006	0.0180±.019	**0.1201±.023**	0.0090±.000	**0.0270±.007**	**0.0210±.006**
Fred-MD	0.7570±.025	**0.8567±.006**	0.4592±.013	0.4081±.068	**0.9813±.010**	0.7103±.071	**0.8972±.018**	**0.5639±.067**	0.0000±.000	0.0000±.000	0.0000±.000	0.0000±.000
Exchange	0.7153±.032	**0.9583±.000**	0.4708±.034	**0.6528±.067**	**0.7847±.074**	0.7778±.044	**0.6819±.043**	0.4792±.021	0.0556±.032	0.8958±.181	0.0194±.009	0.0694±.013

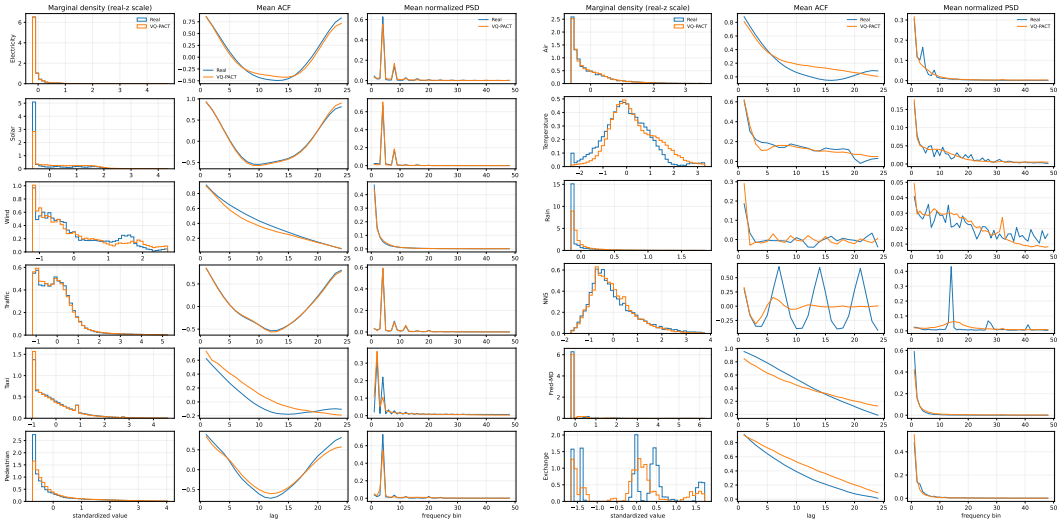


Figure S1: Real and generated temporal profiles after per-window standardization.

The following two figures separate complementary fidelity checks. Figure S2 compares marginal and dynamic quality across domains, whereas Figure S3 verifies that the reported distances stabilize as the generated sample count increases.

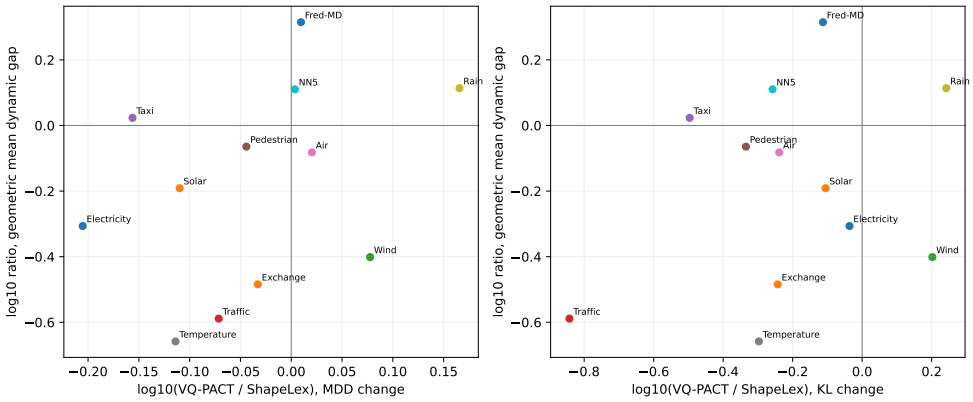


Figure S2: Marginal and dynamic fidelity on the same generated sequences.

The convergence analysis covers six domains with at least 2,000 windows each. Figure S3 addresses a different source of uncertainty: whether the reported distances are stable at the selected sample budget. The median deviation from the full-array estimate decreases as the cohort grows; at 2,000 windows per domain it is 0.096 for MDD and 0.111 for K-L, while ACF-L1, PSD-JS, and Δ-Q are at or below 0.023. The remaining shaded spread records genuine inter-domain variation rather than a single pooled error bar.

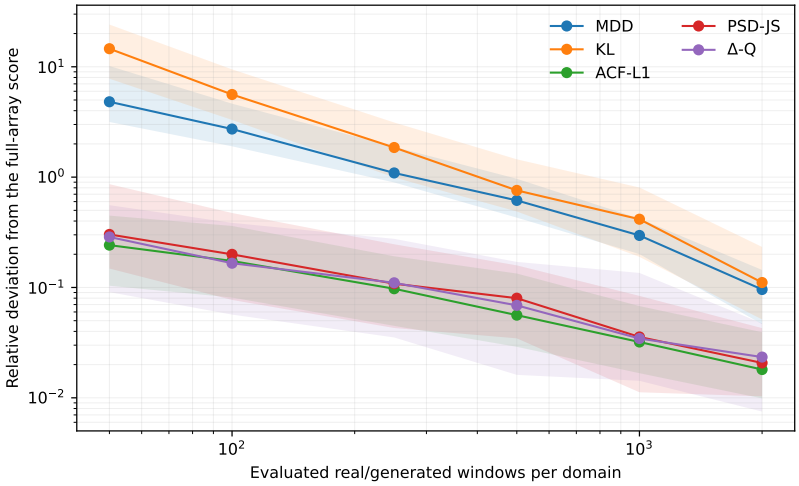


Figure S3: Metric convergence with sample count. Shading shows inter-domain variation.

B.2 Generation from Time-MMD reports

Table 2 compares VQ-PACT, ShapeLex, TimeMAR, BRIDGE, and T2S on eight Time-MMD domains. The cross-method point estimates are reported in the main paper rather than duplicated here. The numeric-response analysis below uses a separate protocol, with its own eligible sample set and uncertainty estimates.

Response to explicit numeric values. This diagnostic reports means and SDs over three generation seeds. Only the report/search-derived fact is visible at inference; target windows, target statistics,

cached target tokens, and prediction fields are excluded. A 96-step embargo separates training and test windows. The explicit-number compiler applies only to parseable numeric constraints.

Table S5 isolates rows whose text contains a parseable numeric value. The correlation between the supplied anchor and the generated-window mean is 0.9185 on Energy, 0.8071 on Health-US, and 0.9187 on SocialGood. Shuffling the texts drives the corresponding correlations with the originally matched anchors close to zero. This intervention tests the numeric realization path's response to the supplied fact, with generated arrays evaluated against the original row-to-fact matching. It does not isolate the neural generator's ability to interpret unrestricted report text. Terminal-value translation is applied only to anchor-valid rows. This intervention is separate from the cross-method fidelity comparison.

Table S5: Generated-mean/original-anchor correlations under original and shuffled text.

Domain	Rows	Generated mean	Shuffled text
Energy	132	0.9185 ± 0.018	-0.0370 ± 0.088
Health-US	150	0.8071 ± 0.031	-0.0395 ± 0.153
SocialGood	19	0.9187 ± 0.040	0.0133 ± 0.011

The comparison in Table 2 evaluates distributional fidelity, whereas this intervention tests the response to explicit factual values on the anchor-valid subset. These results characterize different aspects of real-report generation and are reported separately.

C Additional Text-Control Results and Examples

This section preserves earlier native-system text-control results, multi-draw evaluation, evaluator transfer, text perturbations, and qualitative examples. These experiments are distinct from the matched-executor comparison in Table 3. Selected successful edits are distinguished from output-independent random generations.

C.1 Earlier native-system control comparison

Table S6 supplements the native-pipeline regional comparison in Figure 5 with complete scores from three evaluators. These results compare complete generation systems, not generators under a common executor. VQ-PACT's native realization setting and the unmodified baseline outputs are retained as originally evaluated. The uncertainty aggregation differs between VQ-PACT and the baselines, as specified in Appendix A.1. Segment Accuracy, Macro Recall, and ExtraTrees scores in this table belong only to this earlier experiment and are not transferred to the new main table.

Table S6: Native-system Synth-U control under three frozen evaluators (%).

	ShapeletCNN			Independent evaluators	
Method	Joint ↑	Segment ↑	Macro Recall ↑	ResNet Joint ↑	ExtraTrees Joint ↑
VQ-PACT (Ours)	**93.1025** $\pm$ .482	**97.6517** $\pm$ .168	**98.5019** $\pm$ .108	**72.5700** $\pm$ .558	**86.7970** $\pm$ .492
VerbalTS	<u>39.9325</u> $\pm$ .636	<u>69.9333</u> $\pm$ .484	<u>64.3626</u> $\pm$ .594	<u>38.9180</u> $\pm$ .497	<u>42.1600</u> $\pm$ .698
DiffuSETS	27.8425 $\pm$ .845	57.1708 $\pm$ .451	50.7989 $\pm$ .509	19.7800 $\pm$ .413	28.5650 $\pm$ .629
BRIDGE	17.4025 $\pm$ .285	49.9017 $\pm$ .435	48.5796 $\pm$ .486	18.3480 $\pm$ .326	17.3850 $\pm$ .364
T2S	0.6100 $\pm$ .080	18.5425 $\pm$ .271	22.8571 $\pm$ .334	5.2500 $\pm$ .186	4.6980 $\pm$ .220

C.2 Supplemental Synth-U control protocols

The earlier native-system comparison in Table S6 reports event control under three frozen evaluators; the regional plot is retained in the main text as Figure 5. The single-draw protocol computes metrics for individual generated datasets before aggregating repetitions. Table S7 additionally evaluates the ten-draw lower-median aggregation protocol on the same 4,000 official Synth-U test prompts. The single-draw columns retain the original scores for a direct within-method comparison; aggregation is a different inference protocol, not ten independent model-training runs. The lower-median protocol

produces one aggregate per released baseline checkpoint and one per VQ-PACT generator seed. VQ-PACT generators share the same frozen adapted seed-0 tokenizer. The baseline lower-median columns have no SD because each checkpoint yields one aggregate, not ten independent aggregate estimates. Aggregation does not improve every method: the baseline Joint scores in this protocol are lower than their single-draw scores.

For each prompt and generator seed, aggregation acts on the ten generated waveforms pointwise. At time t, sort the ten values as $x_{(1),t} \leq \cdots \leq x_{(10),t}$ and set $\widetilde{x}_t = x_{(5),t}$. This is the lower median implemented by `torch.median`, not the average of the fifth and sixth values. The frozen ShapeletCNN is then applied to the aggregated sequences, and control metrics are computed from its predictions. No logits, predicted labels, or metric scores are aggregated, and no complete candidate waveform is selected. For VQ-PACT, the three seed-level aggregate scores are summarized by their mean and population SD.

Method	Single-draw control (%)			10-draw lower-median control (%)		
	Joint Accuracy	Segment Accuracy	Segment Macro Recall	Joint Accuracy	Segment Accuracy	Segment Macro Recall
VQ-PACT (Ours)	**93.1025**±.482	**97.6517**±.168	**98.5019**±.108	**98.2583**±.153	**99.4167**±.054	**99.6160**±.037
VerbalTS (ICML 2025)	39.9325±.636	69.9333±.484	64.3626±.594	11.1500	47.2250	34.7846
DiffuSETS (Patterns 2025)	27.8425±.845	57.1708±.451	50.7989±.509	8.6750	43.8583	32.0657
BRIDGE (ICML 2025)	17.4025±.285	49.9017±.435	48.5796±.486	8.5250	41.8417	29.9615
T2S (IJCAI 2025)	0.6100±.080	18.5425±.271	22.8571±.334	0.5750	18.8250	23.2915

Table S7: Single-draw and ten-draw aggregated Synth-U control (%).

C.3 Complete evaluator-transfer statistics

Table S8 adds Segment Accuracy and Macro Recall for the three frozen evaluators. Each is trained only on real Synth-U training data and is not used to train or select VQ-PACT. All metrics in this earlier native-system experiment use the same 4,000 prompts and 30 draws. Values are means and population SDs across three generator seeds after averaging ten draws per seed. They are not computed from the matched-executor main-table outputs.

Table S8: Complete VQ-PACT control metrics under three evaluators (%).

Frozen evaluator	Joint Accuracy ↑	Segment Accuracy ↑	Macro Recall ↑
ShapeletCNN	93.1025 ± 0.4824	97.6517 ± 0.1679	98.5019 ± 0.1081
Independent ResNet	72.5700 ± 0.5576	90.0639 ± 0.1794	93.6691 ± 0.1198
ExtraTrees	86.7967 ± 0.4916	95.3147 ± 0.1769	96.3297 ± 0.1255

C.4 Text-to-program robustness

Values are means and sample SDs across three generator-training seeds, with within-seed draws averaged first. Exact-program identity and factor consistency are proportions.

Program stability under text perturbations. Table S9 compares each perturbed prompt with the program produced from its released caption by the same frozen generator. Synonym rewrites retain $79.4 \pm 2.3\%$ of factor assignments despite only $1.0 \pm 0.8\%$ exact-program identity, while sentence reordering retains $90.5 \pm 0.5\%$. Shuffling captions across test rows reduces factor consistency to $65.8 \pm 0.7\%$ and yields the largest histogram divergence. These results measure stability of typed factor assignments under text perturbation; they do not require literal invariance of every discrete slot.

Figure S4 visualizes the contrast between meaning-preserving rewrites and destructive perturbations. Factor consistency captures stability that exact-program identity misses.

Table S9: Text-to-program robustness on 512 stratified Synth-U prompts.

Text condition	Emb. cosine	Exact program	Factor consistency	Hist. JS ↓
Original	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.000 ± 0.000
Synonym rewrite	0.892 ± 0.000	0.010 ± 0.008	0.794 ± 0.023	0.064 ± 0.011
Sentence reorder	0.967 ± 0.000	0.063 ± 0.016	0.905 ± 0.005	0.019 ± 0.001
Redundant clause	0.957 ± 0.000	0.055 ± 0.007	0.891 ± 0.012	0.022 ± 0.005
Event deletion	0.915 ± 0.000	0.002 ± 0.000	0.818 ± 0.007	0.047 ± 0.004
Contradictory edit	0.967 ± 0.000	0.040 ± 0.021	0.888 ± 0.006	0.024 ± 0.002
Shuffled text	0.795 ± 0.000	0.001 ± 0.001	0.658 ± 0.007	0.138 ± 0.003

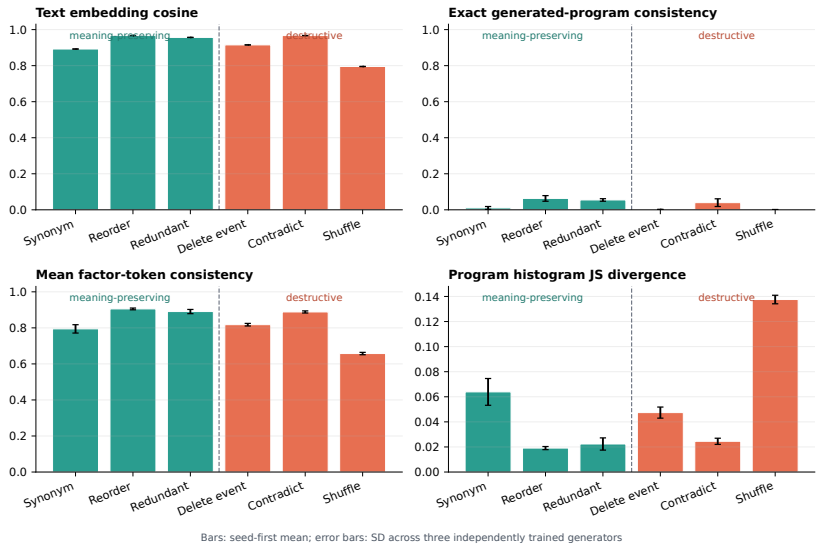


Figure S4: Program stability under meaning-preserving and destructive text changes.

C.5 Complete mechanistic case

Figure S5 expands the selected single-panel example in Figure 4 into the complete Synth-U realization path. The full caption conditions the learned proposal, while its beginning, middle, and end clauses provide regional waveform anchors for the executor. The displayed event strip records these requested anchors rather than predicted factor-code or slot assignments. The same frozen evaluators reject the VerbalTS output but accept VQ-PACT and its single-phrase counterfactual; off-target intervals remain nearly unchanged under the matched intervention. Selection fixes the category, seed, and draw, then takes the smallest test index satisfying three-evaluator dual success. Neither VerbalTS performance nor visual smoothness is a selection criterion.

C.6 Minimal-pair phrase-edit gallery

The two interventions in Figure S6 test a stricter property than prompt-level correctness: only one event phrase is changed while the stochastic draw, decoder, and all other conditions remain fixed. Over 99.99% of the absolute change lies in the target interval in both cases. ShapeletCNN, independent ResNet, and ExtraTrees correctly identify both original and edited events in both cases. These matched interventions illustrate localized control in the selected cases; they do not estimate a dataset-level edit success rate. Case (a) is the frozen main-text case. Case (b) is the first qualifying item in a seed-fixed random bank under the pre-specified edit, evaluator, locality, and reproduction checks.

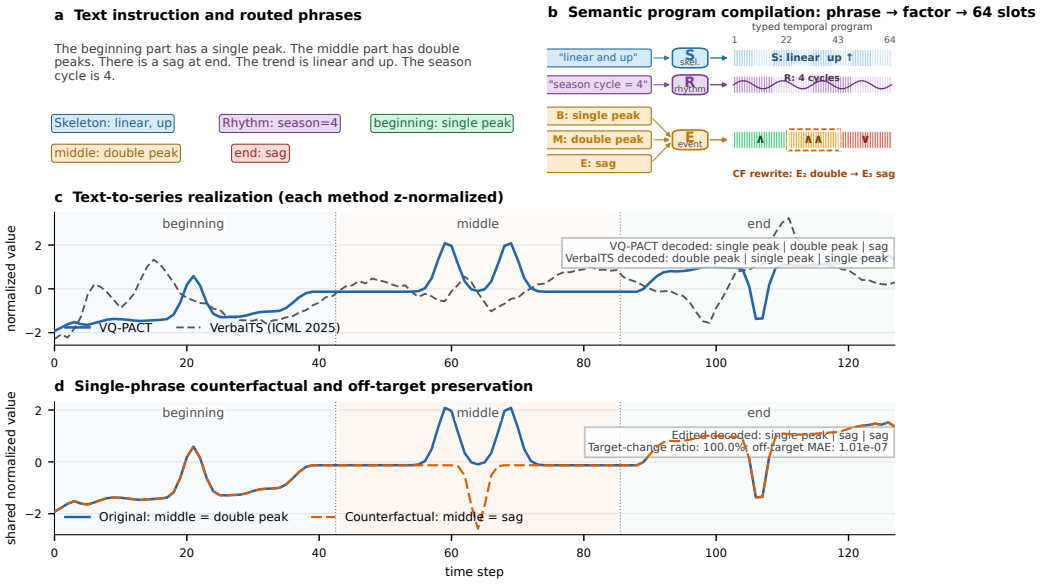


Figure S5: Expanded selected case: (a) text; (b) requested regional anchors; (c) generated sequences; (d) middle-event edit.

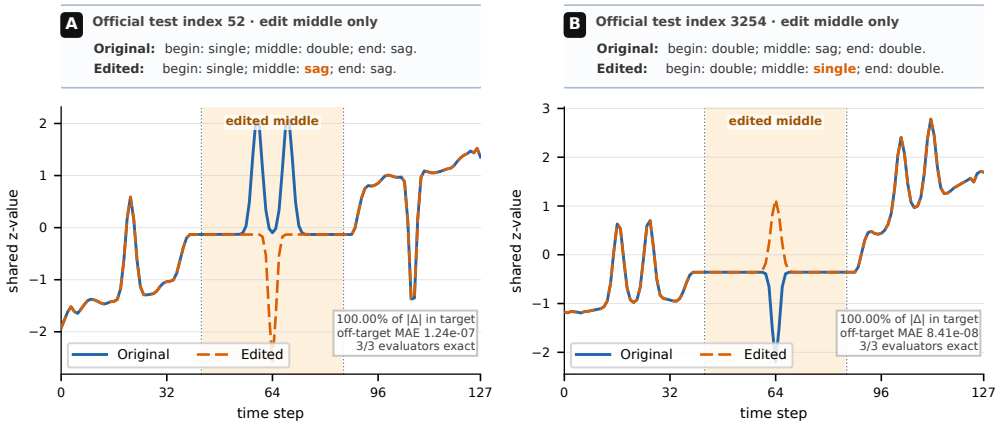


Figure S6: Selected middle-event edits: (a) double peak to sag; (b) sag to single peak. Shading marks the target interval.

C.7 Random generations

Figures S7 and S8 provide a selection-free visual audit: the cases are drawn without inspecting evaluator correctness or smoothness, so they expose ordinary variation rather than only selected showcases. Together they span trend directions, rhythms, and local-event combinations and complement the aggregate control metrics in the main paper.

Figure S7 covers the first half of the fixed random bank and spans different trend directions, rhythms, and event combinations. The stochastic draw cloud varies around a stable conditional layout rather than reproducing the held-out reference pointwise. This visual variability is therefore compatible with high event control and should not be read as pointwise reconstruction error.

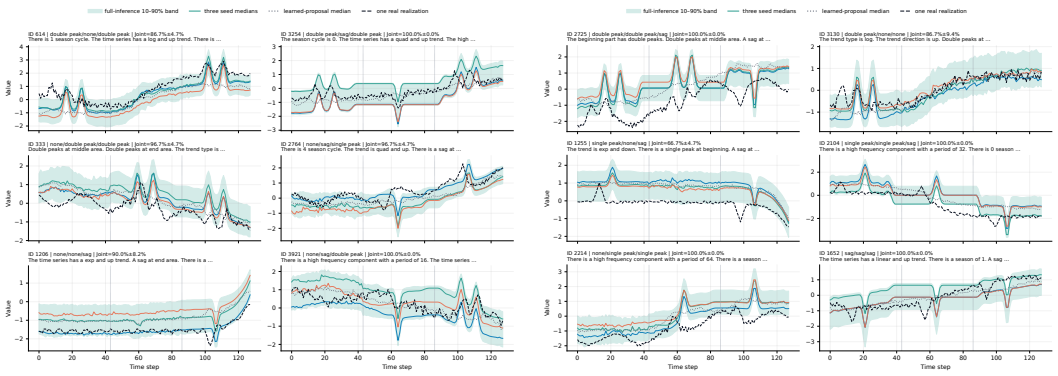


Figure S7: Random Synth-U generations, pages 1–2.

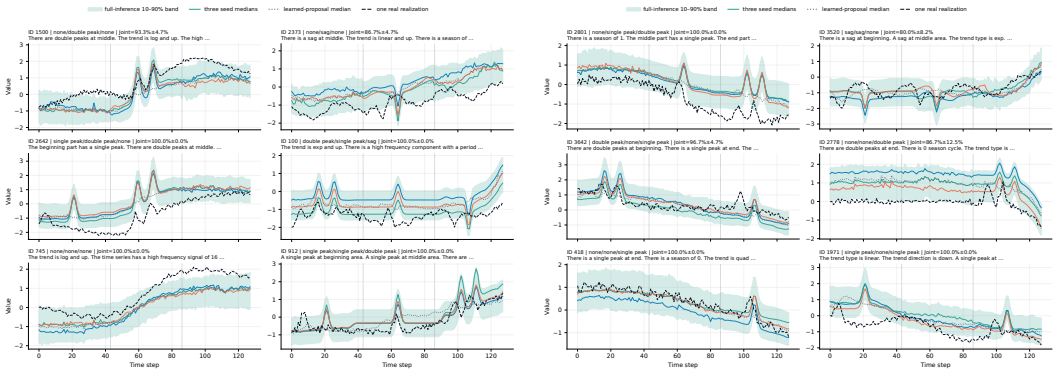


Figure S8: Random Synth-U generations, pages 3–4.

Figure S8 contains the remaining random cases, including combinations with weak local events and stronger global trends. They extend the visual coverage beyond the single mechanistic example while retaining a fixed, output-independent sampling rule.

D Additional Ablations and Mechanism Analyses

We examine code behavior, the components of selective realization, and their control–fidelity trade-off. The remaining analyses test the frozen executor on other backbones and measure computational cost.

D.1 Code use and factor responsibility

Code statistics use the 4,000 official Synth-U test series. Swap uncertainty is the population SD over 1,000 deterministic one-slot interventions per codebook.

Utilization and factor responsibility. Table S10 combines a utilization audit with two complementary diagnostics for the validation-selected $16/16/8$ tokenizer. Target explained variance (EV) is descriptive: it measures association between a code identity and the recurrence-aware target assigned to that stream. Swap sensitivity is interventional: it measures the normalized output change caused by replacing one active code at one slot while holding every other program variable fixed. Neither statistic is used to select the generator or its test outputs.

Figure S9 visualizes the support statistics from the table over all 4,000 official test series. Skeleton and Rhythm activate all 16 codes, with normalized perplexities of 0.856 and 0.869. Event activates 7 of 8 codes; its lower unconditional normalized perplexity (0.333) is expected because the explicit NULL code occupies 73.77% of slots. After conditioning on non-NULL slots, 6 of 7 event codes

Table S10: Codebook use, target association, and one-slot swap sensitivity.

Codebook	Active	Perplexity	Norm. perplexity	Dominant share	Target EV	Swap sensitivity
Skeleton	16/16	13.690	0.856	0.138	0.783	0.00214 ± 0.00179
Rhythm	16/16	13.897	0.869	0.116	0.055	0.00172 ± 0.00177
Event	7/8	2.668	0.333	0.738	0.042	0.00380 ± 0.00437

are active and perplexity is 4.700 (normalized perplexity 0.671). The plot makes both the structural NULL mass and the one inactive event code explicit.

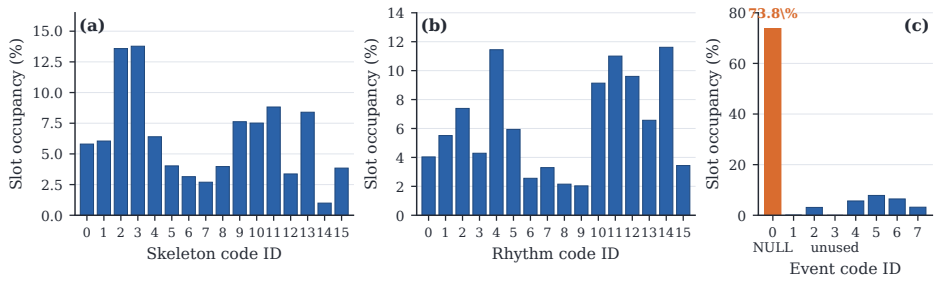


Figure S9: Codebook occupancy. Event code 0 denotes NULL.

Figure S10 then places observational and interventional evidence side by side. Skeleton identity explains 0.783 of its assigned target variance, while the corresponding Rhythm and Event diagonals are 0.055 and 0.042. Off-diagonal EV values remain small, but the unequal diagonal strengths show that the three streams should not be interpreted as fully separated. In the independent swap audit, Event has the largest normalized output sensitivity (0.00380), followed by Skeleton (0.00214) and Rhythm (0.00172). The codebooks therefore have different measured effects on the output, without establishing one-to-one semantic specialization.

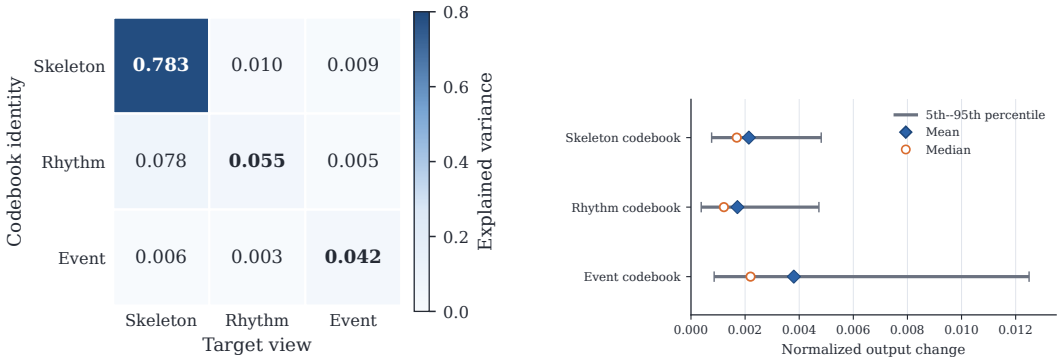


Figure S10: Code–target explained variance (left) and one-slot swap sensitivity (right).

Decoded factor-code atlases. Figure S11 visualizes decoder responses under one common reference context. The skeleton codes span different endpoint slopes, and the rhythm codes vary primarily in dominant frequency and roughness. The final event lexicon has eight codes. An isolated event response does not establish factor responsibility. We therefore complement this view with controlled code swaps in Appendix D.1, rather than inferring semantic specialization from amplitude alone.

D.2 Inference-component ablations

These are inference-time ablations, not retraining ablations of the tokenizer, invariance objective, or grammar-space phrase alignment.



Figure S11: Decoded skeleton (left) and rhythm (right) prototypes under a common context.

The earlier native-setting ablation reported here changes only the deterministic inference policy after the learned $16/16/8$ proposal has been sampled. Its full-realization setting differs from the common executor used in Table 3. Every row uses the same three independently trained generators, the same ten draws per seed, the same official test order, the same caption parser, and the same frozen evaluators. The renderer parameters are selected on validation data and are never changed in response to a test waveform or an evaluator prediction. Table S11 states the interventions exactly.

Table S11: Inference-component interventions and their fixed conditions.

Variant	**Exact intervention**	**Held fixed**
Full realization	Event-blind background; correct third; typed validation prototype	Proposal, captions, parser, prototype parameters, evaluators
w/o background neutralization	Skip event removal and add the typed prototype directly to the proposal	Binding, prototype parameters, proposal, evaluators
w/o full clause coverage	Keep the middle clause and set beginning/end event routes to NULL	Background, typed prototypes, proposal, evaluators
w/o correct phrase–region binding	Apply the same no-fixed-point three-cycle (seed 1729) to every caption	Background, typed prototypes, proposal, evaluators
w/o event-type specificity	Replace all non-NULL types by the equal barycenter of three locked prototypes	Background, correct binding, proposal, evaluators
w/o selective realization	Bypass event removal and prototype insertion	Exact stochastic proposal samples and evaluators

D.2.1 Unified six-row comparison

Table S12 reports regional control and global consistency on the same generated arrays. Shapelet Joint, Segment, and Macro Recall use ShapeletCNN; Global Joint and Six-Joint use the ResNet evaluators defined in Appendix A.3. Draws are averaged within a training seed before uncertainty is computed across seeds. The full path has the highest reported Event Joint, Segment Accuracy, Macro Recall, and Six-Joint scores, but not the highest Global Joint. Distributional fidelity is evaluated separately in the real-world benchmark and the Synth-U control–fidelity sweep. These rows test inference components only, not the training contribution of the tokenizer or phrase alignment.

Table S12: Complete inference-component ablations on Synth-U (%).

Method	Shapelet Joint	Segment	Macro Recall	Global Joint	Six-Joint
w/o selective realization	10.008 ± 0.260	46.169 ± 0.417	34.667 ± 0.576	35.436 ± 0.327	4.143 ± 0.103
Full realization	93.102 ± 0.482	97.652 ± 0.168	98.502 ± 0.108	38.297 ± 0.111	25.695 ± 0.490
w/o background neutralization	80.298 ± 1.449	92.571 ± 0.630	92.170 ± 0.686	33.422 ± 0.205	17.744 ± 0.573
w/o full clause coverage	22.243 ± 0.392	63.834 ± 0.302	56.573 ± 0.391	41.309 ± 0.250	8.762 ± 0.134
w/o correct phrase–region binding	8.159 ± 0.011	30.227 ± 0.138	28.479 ± 0.199	38.681 ± 0.111	1.879 ± 0.013
w/o event-type specificity	10.227 ± 0.027	44.453 ± 0.160	32.343 ± 0.097	42.604 ± 0.034	4.983 ± 0.069

D.2.2 Clause-to-region routing controls

To test phrase–region binding, we remove two event routes or apply a fixed permutation to all three routes while retaining the same proposals and event prototypes (Table S13). Keeping only the middle clause reduces Joint from 93.10% to 22.24%; the seed-fixed derangement reduces it to 8.16% and moves one third of the edit mass outside the caption-active regions. Because the same derangement is applied to every row, the result cannot be attributed to per-example search.

Table S13: Region-binding controls and changes outside caption-active regions.

Routing	Joint ↑	Segment ↑	Macro Recall ↑	Locality ↑	Inactive-edit MAE ↓
Correct caption thirds	$93.103 \pm .482$	$97.652 \pm .168$	$98.502 \pm .108$	$100.000 \pm .000$	$(5.61 \pm .01) \times 10^{-9}$
Middle clause only	$22.243 \pm .392$	$63.834 \pm .302$	$56.573 \pm .391$	$100.000 \pm .000$	$(5.61 \pm .01) \times 10^{-9}$
Seed-fixed shuffled thirds	$8.159 \pm .011$	$30.227 \pm .138$	$28.479 \pm .199$	$66.274 \pm .133$	$0.1564 \pm .0011$

D.2.3 Background, event type, and projection

Background conservation and event typing address different failure modes. Skipping background neutralization leaves existing transients in the requested region and lowers Event Joint to 80.30%. Conversely, an event-blind background with a generic prototype preserves edit locality but cannot distinguish single peak, double peak, and sag, yielding 10.23% Joint. These controls separate *what* is inserted from what the untouched background must conserve.

Finally, Table S14 compares the exact same samples immediately before and after the complete local projection. The learned proposal reaches 10.01% Event Joint before realization. The deterministic projection raises Joint to 93.10%, while 99.999999% of its absolute change remains inside caption-active thirds. Locality is undefined for the pre-projection row because no before/after edit exists.

Table S14: Matched proposals before and after selective realization.

Inference stage	Joint ↑	Segment ↑	Macro Recall ↑	Locality ↑	Inactive-edit MAE ↓
Learned proposal	$10.008 \pm .260$	$46.169 \pm .417$	$34.667 \pm .576$	–	–
+ Selective realization	$93.102 \pm .482$	$97.652 \pm .168$	$98.502 \pm .108$	$100.000 \pm .000$	$(5.61 \pm .01) \times 10^{-9}$

D.3 Control–fidelity operating curve

Effect of event amplitude. Table S15 shows that most of the control gain occurs before the validation-selected 1.0 setting: Joint Accuracy rises from $36.42 \pm 0.31\%$ at 0.4 to $92.03 \pm 0.49\%$ at 0.8, and reaches $93.10 \pm 0.48\%$ at 1.0. Increasing the multiplier to 1.2 or 1.4 adds only 0.04 percentage points. The locked point also improves KL-32 relative to 0.8 (0.0539 versus 0.0555), so the sweep supports the validation decision without redefining it on the test set. The multiplier scales only validation-locked event-template amplitudes. MDD-32 and KL-32 weight the 32 official global-attribute strata equally. Ten stochastic draws are averaged within each generator seed before computing the across-seed mean and population SD.

Figure S12 shows the control plateau beyond the validation-selected multiplier. The remaining settings are diagnostic and are not used to retune the test operating point.

Table S15: Control and fidelity across event-amplitude multipliers.

Operating point	Joint ↑ (%)	MDD-32 ↓	KL-32 ↓
Learned proposal	10.01 ± 0.26	0.1730 ± 0.0009	0.0519 ± 0.0018
0.0	5.68 ± 0.01	0.1860 ± 0.0005	0.1087 ± 0.0020
0.4	36.42 ± 0.31	0.1792 ± 0.0006	0.0728 ± 0.0014
0.6	80.44 ± 0.50	0.1798 ± 0.0006	0.0613 ± 0.0015
0.8	92.03 ± 0.49	0.1797 ± 0.0006	0.0555 ± 0.0016
1.0	93.10 ± 0.48	0.1795 ± 0.0006	0.0539 ± 0.0015
1.2	93.14 ± 0.48	0.1794 ± 0.0006	0.0543 ± 0.0018
1.4	93.14 ± 0.47	0.1794 ± 0.0006	0.0556 ± 0.0018

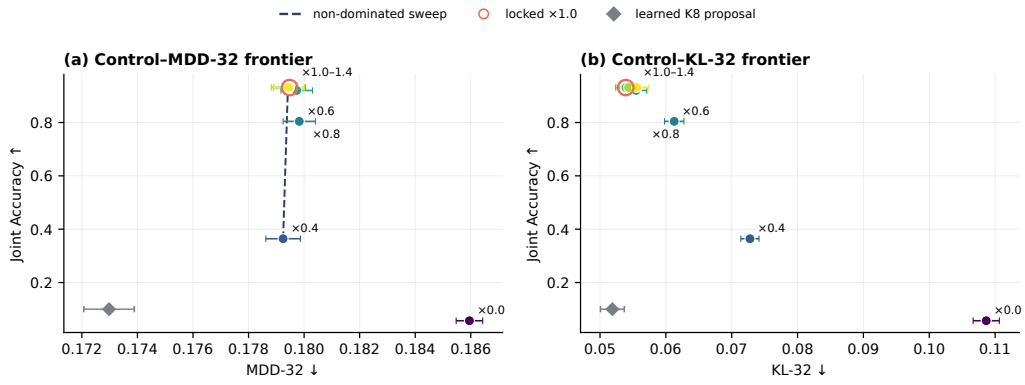


Figure S12: Event control and fidelity across amplitude multipliers. The validation-selected setting is 1.0.

D.4 Earlier transfer of the frozen waveform executor

This is a previously completed, independent transfer experiment, not the generation run summarized in Table 3. Each released seed-0 backbone is evaluated over ten stochastic draws; error terms describe sampling variation. These error terms and the base-generation scores below are not assigned to the new main-table means.

To isolate the contribution of waveform realization across generators, we apply the same frozen neutralize–write executor to four released backbones. Table S16 uses all-factor+event Joint under the same frozen global and local ResNet evaluators: trend type, trend direction, seasonal cycle, and all three local events must be correct together. Each backbone uses the same generated proposals before and after realization, with no backbone-specific executor retuning. All four backbones share a validation-locked transfer configuration with event amplitude 1.3 and width 0.025 for each event type. Its double-peak separation is 0.22. These nominal settings match the common-executor configuration specified for Table 3, but the tables report separate experimental runs and must not be treated as the same generated outputs. This transfer setting differs from the earlier native inference ablation in Table S12, whose single-peak, double-peak, and sag amplitude/width pairs are $(1.7, 0.035)$, $(1.9, 0.035)$, and $(2.1, 0.025)$, respectively. The transfer results assess paired within-backbone gains within this experiment only; no before/after gains are computed across runs.

All four backbones improve under the common realization policy. VerbalTS increases from 35.540% to 60.303%, while T2S increases from 0.653% to 23.503%. This experiment measures the portability of the executor across the tested proposal generators; strict compositional holdouts use the separate protocol in Appendix E.

Table S16: Independent executor-transfer study. Δ is the within-run Six-Joint gain in percentage points.

Backbone	Base	+VQ-PACT executor	Δ
VerbalTS (ICML 2025)	35.540±.415	**60.303**±.256	+24.763
DiffuSETS (Patterns 2025)	14.228±.318	**24.940**±.588	+10.712
BRIDGE (ICML 2025)	11.655±.405	**18.415**±.255	+6.760
T2S (IJCAI 2025)	0.653±.108	**23.503**±.339	+22.850

D.5 Computational efficiency

This is a separate timing study of the native generation paths, not a timing comparison of all backbones followed by the common executor in Table 3. VQ-PACT's measured path includes its native decoding/refinement, while the baseline paths are those listed in Table S17.

Timing results are means and population SDs across three 20-batch blocks. Peak GPU memory is measured at batch size 128. Bold and underlining denote best and second-best values, respectively.

We compare VQ-PACT with four complete, released text-conditioned time-series generators. We measure single-example latency and batched throughput on each native generation path in this separate study, rather than inferring computational cost from parameter count.

Measurement protocol. All methods are evaluated on one NVIDIA GeForce RTX 3090 using Synth-U sequences of length $T{=}128$ and one generated draw per prompt. Cached Qwen3 text embeddings are resident before timing. The timed region includes every method-specific conditioning operation, all serial sampling or refinement steps, and final decoding. It excludes checkpoint and data loading, text encoding, host–device input transfer, file I/O, metric aggregation, and evaluator execution. CUDA events measure batch-1 latency and batch-128 throughput; peak memory is measured for batch 128. Every method–batch pair contains 60 timed batches. Standard cells use five warm-up batches followed by 20 timed batches in each of three repeats. For BRIDGE at batch 128, 60 one-batch repeats are separated by an untimed eight-second cooling interval because uninterrupted execution induced thermal throttling; the retained measurements are regrouped into three blocks of 20 for the same uncertainty calculation. Active parameters are modules executed on the measured path, whereas loaded parameters include all resident task-model modules. The shared frozen text encoder is excluded from both counts.

Table S17 summarizes the executed-path measurements under this common timing boundary.

Table S17: Native-path generation cost on Synth-U. The text encoder is excluded.

Method	**Executed serial path**	**Active params (M)** ↓	**Loaded params (M)** ↓	**Latency (ms)** ↓ $B=1$	**Throughput** ↑ **(series/s)**, $B=128$	**Memory (MiB)** ↓
VQ-PACT (Ours)	64 AR slots + decode/refine	<u>7.63</u>	<u>15.64</u>	<u>117.9 ± 0.7</u>	$\mathbf{460.2 \pm 2.9}$	<u>173.4</u>
VerbalTS (ICML 2025)	50 DDIM steps	39.81	52.49	177.3 ± 0.5	225.9 ± 5.3	267.2
DiffuSETS (Patterns 2025)	50 DDIM steps	18.28	30.59	277.8 ± 1.1	<u>417.2 ± 0.6</u>	194.7
BRIDGE (ICML 2025)	50 DDIM steps + prototype encoding	21.78	22.12	701.6 ± 2.7	93.7 ± 0.4	275.0
T2S (IJCAI 2025)	10 flow steps (20 NFE)	**0.62**	**0.97**	$\mathbf{18.0 \pm 0.3}$	333.1 ± 3.6	**127.6**

Results. VQ-PACT achieves the highest batch-128 throughput at 460.2 ± 2.9 series/s. This is $2.04\times$ the throughput of VerbalTS, $1.10\times$ that of DiffuSETS, $4.91\times$ that of BRIDGE, and $1.38\times$ that of T2S. At batch size one, VQ-PACT ranks second with 117.9 ± 0.7 ms latency; measured by the corresponding latency ratios, it is $1.50\times$, $2.36\times$, and $5.95\times$ faster than VerbalTS, DiffuSETS, and BRIDGE, respectively. It also ranks second in active parameters, loaded parameters, and peak GPU memory, and uses fewer task-model parameters and less memory than the three diffusion-based systems.

Interpretation. The timing results favor different methods at different batch sizes. The current VQ-PACT path generates 64 autoregressive program slots without key–value caching, with joint decoding, scale restoration, and one refinement pass. T2S is therefore substantially better for single-example latency and model size, whereas VQ-PACT benefits more from batching and yields the highest measured throughput. Because the methods execute different operators and serial paths, we do not infer a lower asymptotic complexity from this finite-hardware benchmark.

E Full Compositional Generalization Results

The two protocols isolate reuse across feature combinations and reuse at an unseen temporal location. Trainable generation components are retrained on train-ID, and checkpoint and operating-point selection use only validation-ID. Held-out validation examples are not used for selection. ID and held-out test subsets share one normalization fitted on the complete real test set within each protocol; these statistics are used only for evaluation, not for generation.

Factor-OOD. Let $s = 2u + v \in \{0, \ldots, 7\}$ combine trend type $u \in \{0, 1, 2, 3\}$ (linear, quadratic, exponential, logarithmic) and direction $v \in \{0, 1\}$ (up, down). Let $r \in \{0, 1, 2, 3\}$ index the seasonal-cycle classes $\{0, 1, 2, 4\}$, and let $e \in \{0, 1, 2, 3\}$ denote the middle-region event (NULL, single peak, double peak, or sag). We hold out the 32 of 128 triples satisfying $e = (s + r) \bmod 4$; r in this rule is the class index, not the cycle value. All individual factor values and all pairwise combinations remain in training. The ID/held-out counts are 17,967/6,033 for training, 3,031/969 for validation, and 3,031/969 for testing. Only the ID portions of training and validation are used for fitting and model selection.

Temporal-OOD. The beginning–sag pairing is excluded from training and validation-ID, while sag events remain represented in the middle and end regions. The ID/held-out counts are 19,151/4,849 for training, 3,202/798 for validation, and 3,201/799 for testing. This split tests reuse of a known event at an unseen location, rather than holding out its event class.

Evaluation and uncertainty. Each protocol uses one trained model to generate ten complete test datasets, with one sequence per prompt in each draw. Metrics are computed separately on the ID and held-out subsets in each draw. We report the mean and sample SD across the ten per-draw scores (SD denominator $10 - 1$); this measures sampling variation, not variation across independent training runs. Event Joint and Middle Event use the independent local ResNet; Six-Joint combines its event predictions with the global ResNet attribute predictions.

Control and fidelity under holdout. Table S18 reports results after retraining under each restricted split. Factor-OOD reduces Six-Joint from 61.05% on ID data to 58.12% on held-out combinations, a decrease of 2.93 percentage points. Global Joint changes from 74.64% to 66.03%, while Event Joint changes from 66.34% to 65.80%. Temporal-OOD reduces Six-Joint from 64.51% to 53.72%, a decrease of 10.79 percentage points. Its Global Joint changes from 67.43% to 60.65%, whereas Event Joint changes from 69.65% to 71.50%. The latter result illustrates why event accuracy alone does not determine complete-constraint satisfaction. Uncertainty reflects stochastic draws from one trained model per protocol, not variation across independent training runs.

Table S18: Full results on seen and unseen feature combinations.

Protocol	Split	Six-Joint ↑	Global joint ↑	Event joint ↑	Middle event ↑	MDD ↓	KL ↓
Factor-OOD	ID	61.05 ± 0.43	74.64 ± 0.26	66.34 ± 0.55	87.80 ± 0.50	0.0347 ± 0.0006	0.0145 ± 0.0008
Factor-OOD	Held-out	58.12 ± 1.23	66.03 ± 0.85	65.80 ± 1.29	86.79 ± 0.74	0.0440 ± 0.0010	0.0157 ± 0.0013
Temporal-OOD	ID	64.51 ± 0.70	67.43 ± 0.72	69.65 ± 0.94	89.67 ± 0.51	0.0361 ± 0.0008	0.0175 ± 0.0013
Temporal-OOD	Held-out	53.72 ± 0.76	60.65 ± 1.11	71.50 ± 1.16	82.95 ± 1.14	0.0582 ± 0.0015	0.0263 ± 0.0030

Mean ± sample SD over ten draws from one trained model per protocol.

Figure S13 separates complete-constraint and regional control. Temporal-OOD also reduces middle-event accuracy from 89.67% to 82.95%. Uncertainty reflects stochastic draws from one trained model per protocol, not multiple independent training runs.

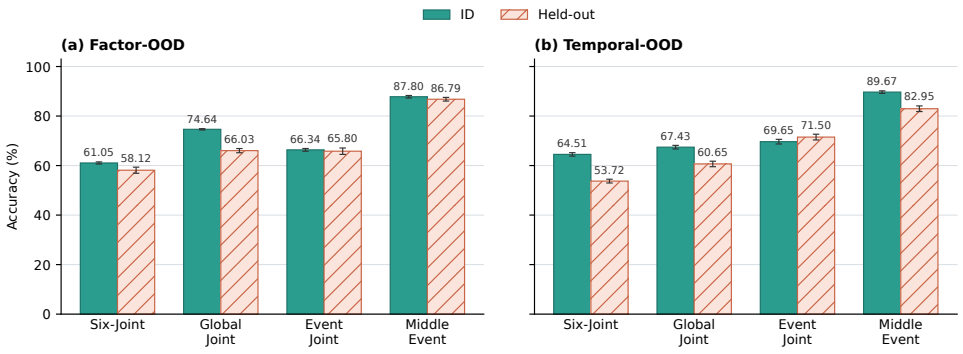


Figure S13: Control on ID and held-out combinations after retraining. Error bars denote sampling standard deviations.

The same separation appears in distributional fidelity. As shown in Figure S14, Factor-OOD changes MDD from 0.0347 to 0.0440 and KL from 0.0145 to 0.0157, whereas Temporal-OOD worsens MDD/K-L from $0.0361/0.0175$ to $0.0582/0.0263$. A common test normalization is used within each protocol, so the ID–held-out comparison is not caused by rescaling the two subsets separately.

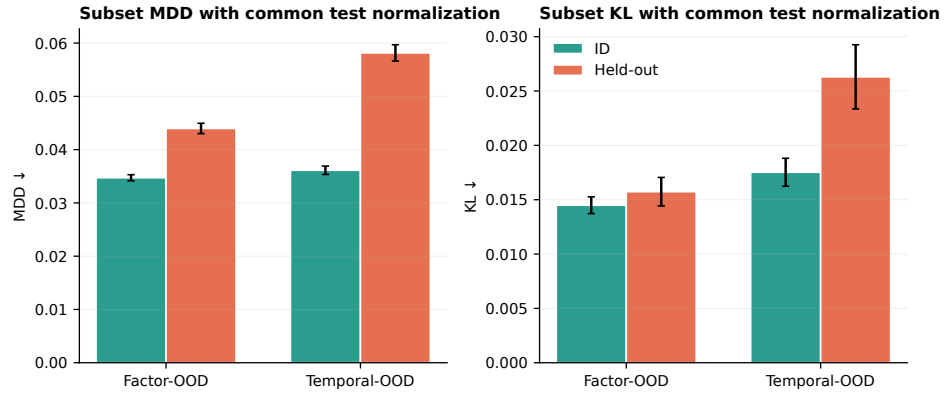


Figure S14: ID and held-out fidelity under a common normalization within each protocol.

Interpretation. VQ-PACT retains complete-constraint accuracy on both held-out sets, with a smaller Six-Joint reduction for Factor-OOD than for Temporal-OOD in the tested protocols. These within-model comparisons assess transfer to unseen factor combinations and event locations separately. They provide evidence of generalization in these settings, but do not establish an advantage over other generators on the same splits.

F Implementation Details and Structural Properties

This section records the tokenizer objective, phrase compiler, realization interfaces, and domain-specific output paths used in the experiments.

F.1 Factor-typed compositional tokenization and optimization

Initialization. The local-shape backbone is initialized from the released VQShape checkpoint (Wen et al., 2024). The three factor codebooks are learned on the pooled twelve-domain training data.

Deterministic factor views. For each standardized local shape u_ℓ, a length-17 smoother extracts the skeleton view v_ℓ^s. The three strongest non-DC Fourier components of the residual give the recurring estimate v_ℓ^{rec}. A median-absolute-deviation transient mask m_ℓ then defines the event and rhythm views:

$$v_\ell^e = m_\ell \odot \left(u_\ell - v_\ell^s - v_\ell^{\mathrm{rec}}\right), \qquad v_\ell^r = u_\ell - v_\ell^s - v_\ell^e. \tag{F.1}$$

Here, $\odot$ denotes elementwise multiplication. The three views sum to u_ℓ by construction. The rhythm view contains recurring variation and residual detail not assigned to the skeleton or event; it is not restricted to a purely periodic component.

Eligible-code quantization. Each factor encoder produces $z_\ell^f = E_f(v_\ell^f)$, where $f \in \{s, r, e\}$ denotes skeleton, rhythm, or event. The nearest eligible code is selected from the corresponding codebook (van den Oord et al., 2017):

$$k_\ell^f = \operatorname*{arg\,min}_{k \in \mathcal{J}_\ell^f} \left\| z_\ell^f - e_k^f \right\|_2^2. \tag{F.2}$$

The eligible set $\mathcal{J}_\ell^f = \mathcal{I}_f$ contains all entries for skeleton and rhythm. Robust event energy with threshold 0.75 determines event presence. An absent event uses $\mathcal{J}_\ell^e = \{0\}$, with 0 reserved for the null event; a present event uses $\mathcal{J}_\ell^e = \mathcal{I}_e \setminus \{0\}$, containing the seven non-null entries. This presence decision also supervises a learned event gate, which does not override the deterministic decision during tokenization.

Full objective. Besides full-series and normalized-shape reconstruction, the tokenizer uses VQ commitment, three factor-decoder losses, decoded-factor-sum consistency, factor-incompatible spectral leakage, factor-preserving latent invariance, and the null-event gate. On the shared twelve-domain tokenizer,

$$\begin{aligned}\mathcal{L}_{\mathrm{tok}} = \mathcal{L}_{\mathrm{ts}} + \mathcal{L}_{\mathrm{shape}} + \mathcal{L}_{\mathrm{vq}} + 0.1\mathcal{L}_{\mathrm{sem}} + 0.1\mathcal{L}_{\mathrm{sum}} + 0.05\mathcal{L}_{\mathrm{leak}} \\ + 0.003\mathcal{L}_{\mathrm{inv}} + 0.05\mathcal{L}_{\mathrm{null}}.\end{aligned} \tag{F.3}$$

Here, $\mathcal{L}_{\mathrm{ts}}$ and $\mathcal{L}_{\mathrm{shape}}$ supervise full-series and normalized-shape reconstruction; $\mathcal{L}_{\mathrm{vq}}$ is the VQ term; $\mathcal{L}_{\mathrm{sem}}$ supervises the three factor decoders; $\mathcal{L}_{\mathrm{sum}}$ encourages the sum of factor reconstructions to match the joint reconstruction; $\mathcal{L}_{\mathrm{leak}}$ penalizes incompatible spectral leakage; $\mathcal{L}_{\mathrm{inv}}$ is the factor-preserving latent constraint; and $\mathcal{L}_{\mathrm{null}}$ supervises the null-event gate. For $\mathcal{L}_{\mathrm{inv}}$, one factor view is held fixed while the other two are independently batch-permuted. The recombined local shape is passed through the deterministic view extractor again, and the encoder processes only its re-extracted factor view. A VICReg-style invariance, variance, and covariance penalty is then applied to the original and recombined pre-quantization representations. During Synth-U adaptation, the weights of $\mathcal{L}_{\mathrm{inv}}$ and $\mathcal{L}_{\mathrm{null}}$ become 0.01 and 0.1, respectively; the other weights are unchanged.

The shared tokenizer uses three continuous warm-up epochs, k-means initialization with at most $100{,}000$ latent samples and 30 iterations, and seven quantized epochs. Synth-U adapts it for eight epochs from the shared initialization. Capacity is selected on development data and fixed to $16/16/8$ before test generation.

F.2 Behavior-grounded phrase alignment and compiler

Prototype naming and prompts. Each factor entry is decoded through its auxiliary decoder. The deterministic verbalizer describes direction/topology for skeleton, dominant rate/strength for rhythm, and polarity/support/location for event. Equal descriptors form one set-valued equivalence class; no generative language model assigns code semantics.

For a tokenized window, factor f selects the production maximizing prototype salience times $\sigma_\ell\sqrt{d_\ell}$. Its normalized coordinate becomes one of five phrases: *at the beginning*, *in the early part*, *around the middle*, *in the late part*, or *near the end*, with bounds $[0, .12)$, $[.12, .32)$, $[.32, .68)$, $[.68, .88)$, and $[.88, 1]$. Short, medium, and long durations use $[0, .12)$, $[.12, .30)$, and $[.30, \infty)$. We emit one compositional caption and one clause per factor. Here, σ_ℓ and d_ℓ are the slot amplitude and duration, as defined in the main method. The full caption and three clauses are embedded by the frozen 1,024-dimensional Qwen3-Embedding-0.6B encoder.

Semantic anchors and candidate regions. After matching a phrase to a descriptor, its compatible code set $\mathcal{K}_j$ is retrieved using Equation 2. Temporal qualifiers specify a normalized interval $I_j = [b_j^-, b_j^+] \subseteq [0,1]$ and duration bounds $[\delta_j^-, \delta_j^+]$. The candidate slots are $\mathcal{R}_j(P) = \{\ell : t_\ell \in I_j\}$, determined by continuous positions rather than slot indices. Target-program positions are used during training, and sampled-proposal positions during inference. The resulting anchor is $A_j^P = (f_j, \mathcal{K}_j, \mathcal{R}_j, [\delta_j^-, \delta_j^+])$, together with its associated interval I_j. For example, *a brief dip near the end* specifies event codes compatible with a dip, a late interval, and a short duration. The anchor set A^P contains at most one anchor per factor.

Regional supervision. A regional phrase need not identify a unique slot. Let $p_{\ell,k}^{f_j}$ be the prior's probability of code k for factor f_j at slot ℓ. The compatible log-probability mass is $u_{j,\ell} = \log \sum_{k \in \mathcal{K}_j} p_{\ell,k}^{f_j}$. For a nonempty candidate region, the regional loss is

$$\begin{aligned} \mathcal{L}_{\text{reg}} &= -\frac{1}{\max(|A^P|, 1)} \sum_j \sum_{\ell \in \mathcal{R}_j} \alpha_{j,\ell} u_{j,\ell}, \\ \alpha_{j,\ell} &= \frac{\exp(u_{j,\ell}/\tau_{\text{reg}})}{\sum_{\ell' \in \mathcal{R}_j} \exp(u_{j,\ell'}/\tau_{\text{reg}})}. \end{aligned} \tag{F.4}$$

Here, j ranges over semantic anchors and $\tau_{\text{reg}} > 0$ is a temperature. The weights $\alpha_{j,\ell}$ distribute supervision within the requested region and are also used for duration supervision. Region membership uses target positions and does not change with current position predictions. If an anchor has no candidate slots, its regional and duration losses are set to zero without evaluating the regional softmax; complete-program supervision remains active.

Dataset interfaces. On the twelve real-world domains, clause embeddings are routed to matching factor heads. Region-MIL uses $\tau_{\text{reg}} = 0.1$; complete-code and region-code terms have unit weight, and complete-attribute and duration terms have weight 2. For datasets without native text, verbalization is split-wise: training-derived prompts alone optimize or select parameters, while a held-out prompt is used only as a generation condition.

On Synth-U, only the official full-caption embedding conditions the autoregressive proposal. Factor routes are zero, $A^P = \varnothing$, and program correction is the identity. The generator uses label-smoothed code cross entropy (smoothing 0.02), $2\mathcal{L}_{\text{MSE}}$ on continuous attributes, and $0.1\mathcal{L}_{\text{scale}}$ on its auxiliary scale head; a text-only scale MDN is trained separately. The official event clauses are parsed only into waveform anchors. For Time-MMD free-form facts, the full embedding is broadcast to all three factor heads; the numeric parser is used only for an explicitly supported terminal value.

F.3 Proposal-conserving closed-ontology waveform realization

Synth-U divides length 128 into $[0, 43)$, $[43, 86)$, and $[86, 128)$. Only thirds requesting a non-null event are masked. The conditional 1-D U-Net sees the masked standardized proposal, binary mask, and global trend type, direction, and seasonal cycle, but never the local event class. It is trained only on all-null training windows from the official split while uniformly sampling seven masks: three single thirds, three pairs, and all thirds. With standardized target z, prediction $\widehat{z}$, a nonempty binary region mask M, and sequence length $T = 128$, the training loss is

$$\mathcal{L}_{\text{inp}} = 4\,\frac{\|M \odot (\widehat{z} - z)\|_2^2}{\|M\|_1} + \frac{\|\widehat{z} - z\|_2^2}{T}. \tag{F.5}$$

Proposal standardization and background completion. The executor runs after the proposal is decoded, resized, scale-restored, and clipped to training quantiles. Let $x^{(0)}$ denote this complete proposal, with mean m_0 and stabilized standard deviation $s_0 > 0$. Its standardized form is $z_t^{(0)} = (x_t^{(0)} - m_0)/s_0$. These proposal statistics are computed after domain processing and are distinct from the MDN-sampled level and scale.

Each non-null clause specifies an event type q_j and a region mask M_j. Let $M = \bigvee_j M_j$ and $\overline{M} = 1 - M$. The conditional inpainter I_ω uses the unmasked proposal, the mask, and the text-

specified global context $\gamma(c)$ to complete the background:

$$b = \overline{M} \odot z^{(0)} + M \odot I_\omega\Big(\overline{M} \odot z^{(0)}, M, \gamma(c)\Big). \tag{F.6}$$

The context $\gamma(c)$ contains trend type, direction, and seasonal cycle, but not the requested local event class. The event waveform is then written into this background using Equation 4. Reusing m_0 and s_0 for inverse normalization preserves the proposal's raw sample values outside the active thirds. A NULL clause activates no mask and leaves event absence to the conditional prior.

Analytic event waveforms. For normalized within-region time $u \in [0, 1]$, define a Gaussian profile $G(u; c_0, w) = \exp[-(u - c_0)^2/(2w^2)]$, where c_0 is its center and w its width. Let $\mathcal{N}_\infty(q) = q/\max_u |q(u)|$ normalize a waveform by its maximum absolute amplitude. The validation-locked primitives are

$$\begin{aligned}
\tau_{\text{single}}(u) &= a_s \mathcal{N}_\infty\big(G(u; \tfrac{1}{2}, w_s)\big), \\
\tau_{\text{double}}(u) &= a_d \mathcal{N}_\infty\big(\tfrac{1}{2}G(u; \tfrac{1}{2} - \tfrac{\delta_d}{2}, w_d) + \tfrac{1}{2}G(u; \tfrac{1}{2} + \tfrac{\delta_d}{2}, w_d)\big), \\
\tau_{\text{sag}}(u) &= -a_g \mathcal{N}_\infty\big(G(u; \tfrac{1}{2}, w_g)\big),
\end{aligned} \tag{F.7}$$

Here, a_s, a_d, a_g are amplitudes, w_s, w_d, w_g are widths, and δ_d is the separation between the double peaks. In the earlier native VQ-PACT setting, their values are $(a_s, w_s) = (1.7, 0.035)$, $(a_d, w_d, \delta_d) = (1.9, 0.035, 0.22)$, and $(a_g, w_g) = (2.1, 0.025)$. The matched-executor comparison in Table 3 instead fixes all three amplitudes at 1.3, all three widths at 0.025, and $\delta_d = 0.22$, with the same frozen partial-inpaint configuration across methods. A NULL clause has $M_j \equiv 0$ and writes no primitive. The test-time executor never queries an evaluator.

F.4 Code-level constraints, scale, and domain realization paths

Code-level constraint realization. Given a complete sampled proposal $P^{(0)}$, code-level constraints produce $\overline{P} = \Pi_{A^P}(P^{(0)})$ before joint decoding. If a compatible code already occurs within the requested interval, it is retained. Otherwise, the candidate slot closest to the interval center is selected, and its addressed factor code is replaced with a canonical member of $\mathcal{K}_j$. When the interval contains no candidate slot, the nearest slot is selected and its position is projected onto I_j. The selected duration is clipped to $[\delta_j^-, \delta_j^+]$ when necessary. Each anchor changes at most one factor-code cell and the required position or duration attributes; other code identities remain unchanged. Because position and duration are shared within a slot, this statement concerns code identities rather than a guarantee that all decoded factor waveforms remain unchanged. Without program anchors, or when using output-space constraints, $\overline{P} = P^{(0)}$.

Joint decoding and scale restoration. Joint decoding and resizing give $y = \text{Resize}_T(\text{Dec}_\psi(\overline{P}))$. A mixture density network (MDN) (Bishop, 1994) models training-window means and log standard deviations. It is conditioned on the full-text embedding h_c and, where enabled, the domain-specific pooled program summary $\rho_{\mathcal{D}}(\overline{P})$. The MDN jointly samples the overall level m and a log standard deviation, whose exponential gives $s > 0$. The output is

$$\widetilde{x}_t = m + s\,\frac{y_t - \mu(y)}{\sigma(y) + \epsilon}, \qquad x^{\text{raw}} = \mathcal{S}_{\mathcal{D}}(\widetilde{x}). \tag{F.8}$$

Here, $\mu(y)$ and $\sigma(y)$ are the decoded shape's mean and standard deviation, and $\epsilon > 0$ stabilizes normalization. The map $\mathcal{S}_{\mathcal{D}}$ applies training-fitted scale calibration or physical-support processing. Its configuration is fixed before testing, and it is the identity when no processing is used. Code-level constraints return x^{raw}. The output-space interfaces instead use $x^{(0)} = x^{\text{raw}}$ as their proposal; the Synth-U neutralize–write procedure is detailed in Appendix F.3.

Domain-specific output processing. Table S19 records the frozen output path for each twelve-domain experiment. Mean/log-std transport adjusts sampled window scale; global monotone uses a train-fitted value map; support-aware count restores discrete nonnegative support; and program-conditioned scale augments the caption with a pooled free-running program. All fitted statistics use training data, every choice is fixed before outer testing, and identity is retained among the candidates. Three domains retain identity realization; six use a scale or monotone-value adaptation, Taxi uses the count-support link, and Wind and Exchange activate program-conditioned scale. The table specifies the output transformation associated with each reported domain result.

Table S19: Frozen scale and support processing by domain.

Dataset	Realization path
Electricity	Identity
Solar	Global monotone value map
Wind	Program-conditioned scale + mean/log-std transport
Traffic	Global monotone value map
Taxi	Support-aware count link
Pedestrian	Identity
Air	Mean transport + global monotone value map
Temperature	Mean/log-std transport
Rain	Identity
NN5	Global monotone value map
Fred-MD	Mean/log-std transport
Exchange	Program-conditioned scale

Numeric-anchor intervention. For the numeric-anchor intervention, a requested endpoint $r^\star$ is realized by

$$x_t^{N,\star} = x_t^{\mathrm{raw}} + (r^\star - x_T^{\mathrm{raw}})\,. \tag{F.9}$$

This unique constant translation reaches $r^\star$ while leaving every first difference unchanged. This numeric-anchor intervention is evaluated separately from the cross-method fidelity comparison in Table 2. Its correlations describe the response to explicit numeric constraints on anchor-valid reports.

F.5 Structural properties of grounding and realization

This subsection gives two structural properties of the grounding and realization mechanisms in the main method. They follow directly from the construction and do not require an assumption about the data distribution.

Code-index invariance of decode–name–reground alignment. Consider an arbitrary permutation π_f of factor-f code indices, together with the corresponding permutation of code vectors and output-head coordinates. For every original entry e_k^f, the relabeled tokenizer stores the same vector at index $\pi_f(k)$; hence its auxiliary decoded waveform and deterministic descriptor are unchanged. Writing $e^{f,\pi}_{\pi_f(k)}$ for the relabeled embedding and $\nu_k^f = \mathcal{V}_f(D_f(e_k^f))$ for its descriptor, we have

$$D_f(e^{f,\pi}_{\pi_f(k)}) = D_f(e_k^f), \qquad \mathcal{V}_f\Big(D_f(e^{f,\pi}_{\pi_f(k)})\Big) = \nu_k^f. \tag{F.10}$$

It follows that the equivalent-code set under the new indexing is $\mathcal{K}_\pi(f,\nu) = \pi_f(\mathcal{K}(f,\nu))$. The decoded factor-view referent selected by a phrase is therefore unchanged; only its storage address is relabeled.

Constraint-local conservation. For a program anchor, the executor either retains an already compatible code or replaces one factor–slot code. Because at most one anchor is emitted per factor, the number of changed code cells is bounded by $|A^P|$. For a waveform anchor, the neutralize–write executor multiplies every learned or analytic modification by M. Multiplying the construction by $(1-M)$ therefore gives $(1-M)\odot x^W = (1-M)\odot x^{(0)}$. Finally, the numeric executor adds the same constant to every sample, so

$$x_{t+1}^{N,\star} - x_t^{N,\star} = x_{t+1}^{\mathrm{raw}} - x_t^{\mathrm{raw}} \quad \text{for all } t < T. \tag{F.11}$$

These three results establish conservation in the native variable space of each supported anchor type.