LIFT: LOCAL-TO-INTERACTION FLOW MATCHING FOR MULTIVARIATE TIME SERIES GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time-series generators must reproduce both the dynamics of individual channels and the dependencies that make their trajectories jointly coherent. Several existing generators provide temporal context for cross-channel computation, but do not explicitly train the representations used for such interaction through a channel-local velocity prediction objective. We introduce **LIFT**, a flow-matching framework for unconditional generation that trains channel-local temporal representations and reuses the same states for multivariate refinement. A shared temporal encoder processes each channel independently, and a local head predicts its velocity using only that channel's trajectory. An interaction module reads the resulting states and predicts an additive correction conditioned on the joint sample. The local and complete velocities are trained jointly against the same flow-matching target; under a joint realizability assumption, the ideal correction is the difference between joint and channel-local conditional velocity estimates. A mean–carrier module combines broadcast mean context with channel-specific readouts from a small set of pooled carriers, yielding computation linear in channel count at fixed width and carrier count. Across six benchmarks and eight baselines, LIFT achieves the lowest three-seed mean in 23 of 24 dataset–metric comparisons. Controlled variants further support temporally informed interaction and the reuse of locally supervised temporal states for joint refinement.

1 INTRODUCTION

Multivariate time-series generation requires more than realistic trajectories for individual channels. A generated sample must also preserve the dependencies that make those trajectories jointly coherent (Yoon et al., 2019; Yuan & Qiao, 2024). Independently reassigning complete channel trajectories across samples illustrates this distinction: each empirical channel distribution remains unchanged, while the correspondence among channels can be disrupted. A successful generator must therefore capture both how individual trajectories evolve and how they fit together as a multivariate sample. In diffusion and flow models, these requirements shape the prediction made from each intermediate generative state.

Several existing generators provide temporal context for cross-channel computation. CSDI passes temporal-attention outputs to feature attention within each residual block (Tashiro et al., 2021), while PaD-TS and DiM-TS combine temporal and variable branches (Li et al., 2025; Yao et al., 2026). These designs do not explicitly supervise the representations used for interaction through a channel-local velocity prediction objective. This leaves open a design question: can temporal states trained to predict each channel's velocity from its own trajectory also provide the basis for joint multivariate refinement? The question concerns both the supervision assigned to temporal representations and their reuse across local prediction and channel interaction.

Temporal context provides a concrete motivation for studying this question. When adjusting the relative timing of two trajectories, the same value on rising and falling segments can require corrections of opposite sign. A temporal encoder can distinguish these situations before channels exchange information. A channel-local prediction objective gives these temporal states a direct task grounded in the channel's own dynamics. Reusing the states for interaction then allows multivariate refinement to build on the representation trained for that task.

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.

We introduce **LIFT**, a local-to-interaction flow-matching framework (Lipman et al., 2023). LIFT expresses the multivariate velocity as the sum of a channel-local prediction and an interaction-conditioned correction. A shared temporal encoder processes each channel independently, using its complete trajectory together with flow time and channel identity. A local head predicts velocity from the resulting states, and an interaction refiner reuses the same states to predict a correction conditioned on the joint sample. The local and complete velocities are trained jointly against the same flow-matching target. Local supervision gives the shared states an explicit within-channel prediction objective, while the joint objective adapts them for multivariate generation.

The objective also gives this decomposition a population interpretation. For a fixed flow coupling and positive local-loss weight, under joint realizability, the ideal local field is the conditional velocity estimate given the target channel alone, while the ideal correction is the difference between the joint and local conditional estimates. At this optimum, the correction is the change in the conditional velocity estimate when the other channels become available, without requiring a separately constructed residual target.

For cross-channel communication, LIFT uses a *mean–carrier interaction module*. A broadcast mean supplies shared context, while a small set of pooled carriers returns channel-specific messages through receiver-dependent readouts. This compact exchange operates on the locally supervised temporal states and scales linearly with the number of channels at fixed hidden width and carrier count.

Across six benchmarks and eight baselines, LIFT achieves the lowest three-seed mean in 23 of 24 dataset–metric comparisons. Controlled variants connect this performance to the proposed organization: temporally informed interaction substantially improves over pointwise interaction, while reusing locally supervised states yields lower means than separate temporal encoding in most comparisons. Additional controls examine output organization, local supervision, and alternative interaction modules. Fixed-model correspondence interventions further show that the learned field responds to cross-channel alignment while the target trajectory is unchanged, and experiments with separately trained longer-window models extend the evaluation to sequence lengths up to 256.

Our contributions are:

- **Locally supervised temporal-state reuse.** We introduce a flow-matching framework that reuses channel-local prediction states for additive multivariate refinement, together with a conditional-velocity interpretation of the resulting decomposition.
- **Mean–carrier interaction.** We develop a refiner that combines shared mean context with channel-specific carrier readouts, with interaction computation linear in channel count at fixed width and carrier count.
- **Generation quality and controlled evaluation.** We establish strong benchmark performance and use targeted controls and correspondence interventions to examine the proposed design and the learned field's use of cross-channel information.

2 Related Work

Time-series generation. Generative models encode temporal structure through different objectives and representations. TimeGAN supervises transitions in an adversarial latent space, while TimeVAE and KoVAE impose structure through their decoders or latent dynamics (Yoon et al., 2019; Desai et al., 2021; Naiman et al., 2024). Diffusion-TS uses trend–seasonality decomposition (Yuan & Qiao, 2024); TimeLDM and L2D-Diff organize diffusion through latent representations (Qian et al., 2024; Shen et al., 2026). FlowTS applies rectified flow to time-series synthesis (Hu et al., 2025); PaD-TS regularizes population-level distributions (Li et al., 2025). LIFT builds on flow matching (Lipman et al., 2023; Liu et al., 2023) and studies how to couple an explicitly supervised local prediction to a joint correction.

Temporal representations and channel exchange. State-space models provide temporal representations for generative modeling (Zhou et al., 2023; Alcaraz & Strodthoff, 2023). In multivariate models, CSDI uses sequential temporal and feature attention (Tashiro et al., 2021), whereas PaD-TS and DiM-TS combine temporal and variable branches (Li et al., 2025; Yao et al., 2026). LIFT cou-

ples representation sharing to explicit local supervision: the states predicting channel-local velocity also supply the interaction-conditioned correction. Appendix B.3 details these information paths.

Compact communication has precedents in forecasting. Crossformer routes information through a small set of routers (Zhang & Yan, 2023), and SOFTS combines a shared core with individual series (Han et al., 2024). LIFT uses a broadcast mean together with carriers of centered temporal states, channel-specific readouts, and flow-conditioned gates. These mechanisms refine a locally supervised flow field. Our ablations examine temporal-state reuse and the exchange operator separately (Section 5.3).

3 PRELIMINARIES

Let $\mathbf{x}_1 \in \mathbb{R}^{C\times T}$ denote a window with C channels and T sequence positions. We use $t \in [0,1]$ for flow time, $\tau \in \{1,\dots,T\}$ for sequence position, and d for hidden width. Flow matching couples Gaussian noise and data, $(\mathbf{X}_0, \mathbf{X}_1) \sim \pi \in \Pi(p_0, p_{\mathrm{data}})$, where $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$. We use minibatch optimal transport (Tong et al., 2024) and draw $t \sim \mathcal{U}[0,1]$ independently. The interpolated state and regression target are

$$\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{X}_1, \qquad \mathbf{U} = \mathbf{X}_1 - \mathbf{X}_0. \tag{1}$$

Writing Q_π for the law of $(\mathbf{X}_t, \mathbf{U}, t)$, training minimizes $\mathbb{E}_{Q_\pi}\|\mathbf{v}_\theta(\mathbf{X}_t,t) - \mathbf{U}\|_F^2$. Generation integrates $\mathrm{d}\mathbf{x}_t/\mathrm{d}t = \mathbf{v}_\theta(\mathbf{x}_t,t)$ from $t=0$ to $t=1$. Throughout, $\mathbf{x}_{t,c}$ denotes the *complete trajectory* of channel c in the current intermediate sample.

4 METHOD

4.1 LOCAL PREDICTION AND JOINT REFINEMENT

LIFT assigns two roles to a shared temporal representation (Figure 1). Each channel's states feed a local velocity head; the collected states also feed a multivariate correction head. Their outputs sum to the complete velocity:

$$\begin{aligned} \mathbf{v}_{\theta,c}(\mathbf{x}_t,t) &= \mathbf{v}^{\mathrm{loc}}_{\phi,c}(\mathbf{x}_{t,c},t) + \mathbf{r}^{\mathrm{int}}_{\theta,c}(\mathbf{x}_t,t), \\ \mathbf{r}^{\mathrm{int}}_{\theta,c} &= g_{\psi,c}(\mathbf{H}^{\mathrm{loc}}_{\phi}, \mathbf{x}_t, \mathbf{q}_t), \qquad \theta = (\phi,\psi). \end{aligned} \tag{2}$$

Here $\mathbf{q}_t$ embeds flow time, and $\mathbf{H}^{\mathrm{loc}} \in \mathbb{R}^{C\times T\times d}$ collects the temporal states. The local predictor also receives channel identity $\mathbf{e}_c$, but only its own channel's observations. Thus, *local* refers to channels, not sequence positions. The correction reads all channels' states and current-sample statistics. The local and summed predictions train the shared encoder jointly through the objective in Section 4.4.

4.2 CHANNEL-LOCAL TEMPORAL PREDICTION

Temporal encoding distinguishes values with different trajectory contexts, such as rising and falling segments (Appendix A). We encode each complete trajectory and predict its local velocity:

$$\begin{aligned} \mathbf{H}^{\mathrm{loc}}_c &= F_\phi(\mathrm{Embed}(\mathbf{x}_{t,c}), \mathbf{q}_t, \mathbf{e}_c), \\ \mathbf{v}^{\mathrm{loc}}_{\phi,c} &= \mathbf{H}^{\mathrm{loc}}_c \mathbf{w}_{\mathrm{loc}} + b_{\mathrm{loc}}\mathbf{1}_T. \end{aligned} \tag{3}$$

The encoder contains four bidirectional diagonal state-space blocks based on S4D (Gu et al., 2022). It shares temporal weights across channels and conditions on flow time and channel identity. Numerical trajectories are processed independently, so sharing weights introduces no cross-channel observations into the local prediction. Complete-window generation permits bidirectional encoding: states at position τ summarize both earlier and later positions. Subsequent position-wise exchange therefore communicates temporally informed states, even though it does not itself attend across sequence positions. Appendix C gives the implementation.

4.3 MEAN–CARRIER INTERACTION REFINEMENT

The refiner communicates through a shared mean and a small set of carriers. The mean provides context common to every receiver. Carriers preserve additional distinctions by aggregating different

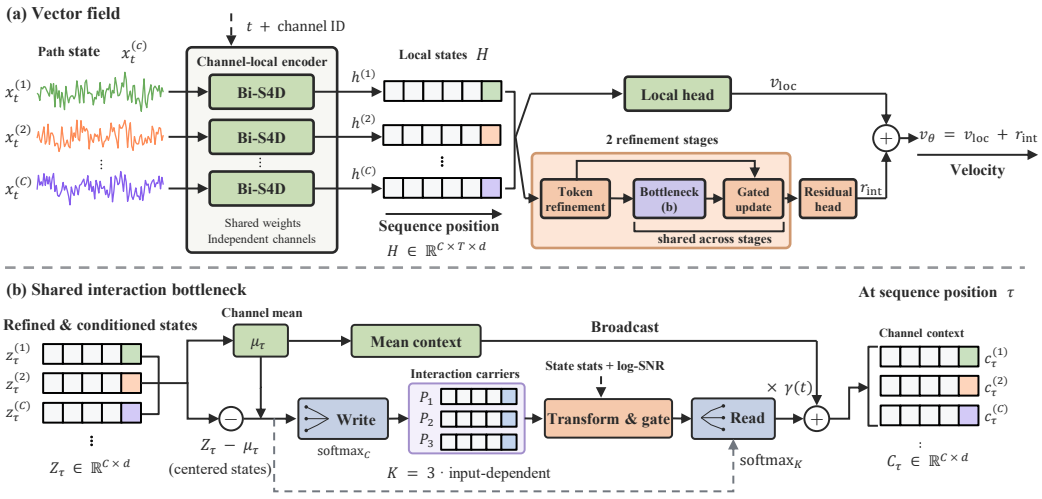

Figure 1: LIFT reuses local temporal states for joint velocity prediction. (a) A shared channel-local encoder supplies a local velocity and states for two refinement stages. The local and summed fields use the same velocity target. (b) Mean context and $K = 3$ input-dependent carriers supply channel-specific messages. Write weights normalize over channels; read weights normalize over carriers. Each stage recomputes the exchange using a shared bottleneck. Diagram symbols H, $z_\tau^{(c)}$, and $c_\tau^{(c)}$ correspond to $\mathbf{H}^{\mathrm{loc}}$, $\mathbf{N}_{c,\tau}$, and $\mathbf{Z}_{c,\tau}^{\mathrm{ctx}}$.

mixtures of channel states and allowing each receiver to read a different message. This organization avoids constructing a full channel-by-channel attention matrix.

Starting with $\mathbf{H}^{(0)} = \mathbf{H}^{\mathrm{loc}}$, we apply $S = 2$ refinement stages. At each stage, a token-wise module and flow-conditioned modulation prepare states $\mathbf{N}$. For clarity, we omit batch and stage indices in the exchange equations. At each position τ, we compute

$$\boldsymbol{\mu}_\tau = \frac{1}{C}\sum_c \mathbf{N}_{c,\tau}, \qquad \boldsymbol{\delta}_{c,\tau} = \mathbf{N}_{c,\tau} - \boldsymbol{\mu}_\tau. \tag{4}$$

The mean follows a direct broadcast path. Centered states write to $K = 3$ carriers, and each channel reads a mixture of their outputs:

$$\begin{aligned} \omega^{\mathrm{wr}}_{c,\tau,k} &= \mathrm{softmax}_c([\mathbf{W}_{\mathrm{wr}}\boldsymbol{\delta}_{c,\tau}]_k), \qquad \mathbf{P}_{k,\tau} = \sum_c \omega^{\mathrm{wr}}_{c,\tau,k}[\mathbf{W}_{\mathrm{val}}\boldsymbol{\delta}_{c,\tau}]_k, \\ \widehat{\mathbf{P}}_{k,\tau} &= \rho_k(\mathbf{x}_t, t)\Phi_{\mathrm{car}}(\mathbf{P}_{k,\tau}, \mathbf{q}_t), \quad \omega^{\mathrm{rd}}_{c,\tau,k} = \mathrm{softmax}_k([\mathbf{W}_{\mathrm{rd}}\boldsymbol{\delta}_{c,\tau}]_k), \\ \mathbf{D}_{c,\tau} &= \sum_k \omega^{\mathrm{rd}}_{c,\tau,k}\widehat{\mathbf{P}}_{k,\tau}. \end{aligned} \tag{5}$$

The write softmax normalizes over channels, selecting each channel's contribution to a carrier. The read softmax normalizes over carriers, selecting the message received by each channel. The value projection produces K width-d vectors; $[\cdot]_k$ selects the kth. Carrier contents and routing weights are recomputed from the current states at each stage.

The gate $\rho_k(\mathbf{x}_t, t)$ uses flow time and sample-level statistics to modulate carrier messages. A bounded learned scale $\gamma(t)$ combines the messages with the transformed mean:

$$\mathbf{Z}_{c,\tau}^{\mathrm{ctx}} = \Phi_{\mathrm{mean}}(\boldsymbol{\mu}_\tau, \mathbf{q}_t) + \gamma(t)\mathbf{D}_{c,\tau}. \tag{6}$$

A gated residual update combines $\mathbf{Z}_{c,\tau}^{\mathrm{ctx}}$ with the receiver's own state, and a linear head on the final states predicts $\mathbf{r}^{\mathrm{int}}$. Receiver states make the mean-path updates channel-specific; carriers additionally individualize the *received context*. Stages share exchange and update weights. Appendix C.1 specifies the gates, updates, and the effect of centering.

Computational complexity. Each position uses $C \times K$ write/read weights, with $O(TCKd)$ aggregation and redistribution cost. The carrier-specific value projection costs $O(TCKd^2)$ and materializes $O(TCKd)$ values. For fixed K, d, computation and activations therefore scale linearly with C per stage. End-to-end generation also includes temporal encoding and repeated field evaluations. Appendix G.2 reports complete-sampler costs.

4.4 Joint Learning and Sampling

The local and complete fields predict the same sampled velocity $\mathbf{U}$. At training update n, the objective is

$$\mathcal{L}_n = \mathbb{E}_{Q_\pi}\left[\|\mathbf{v}_\phi^{\mathrm{loc}} + \mathbf{r}_\theta^{\mathrm{int}} - \mathbf{U}\|_F^2 + \lambda_{\mathrm{loc}}^{(n)}\|\mathbf{v}_\phi^{\mathrm{loc}} - \mathbf{U}\|_F^2\right]. \tag{7}$$

The auxiliary term gives the local component a direct prediction objective. Supervising only the sum leaves an unrestricted decomposition underconstrained: adding a local function to one component and subtracting it from the other preserves the complete field. Local supervision constrains this freedom. The complete-field gradient reaches the encoder through both heads; the correction has no separately constructed target. We reduce λ_{loc} from 1 to 0.25 during training, following the schedule in Appendix C.

Population interpretation. The objective also explains what the two components would estimate at a population optimum. Fix the coupling π and a positive local weight, assume square-integrable targets, and suppose the model can jointly realize both conditional regressions. Then an optimum satisfies Q_π-almost surely

$$\begin{aligned} \mathbf{v}_c^{\mathrm{loc},*} &= \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathbf{X}_{t,c}, t], \\ \mathbf{r}_c^{\mathrm{int},*} &= \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathbf{X}_t, t] - \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathbf{X}_{t,c}, t]. \end{aligned} \tag{8}$$

The ideal correction is the change in the conditional velocity estimate when all channels become available. Appendix D proves this statement by decomposing the two regression risks. A finite shared network need not attain this optimum. Its correction also retains local processing capacity, so the decomposition describes interaction-conditioned refinement, not an identified causal interaction.

Generation. We jointly generate all channels with the final exponential moving average (EMA) model using 100 FP32 Euler steps. Each evaluation recomputes local states, exchanged context, and summed velocity (Algorithm 1).

5 Experiments

We evaluate LIFT through four questions: its overall generation quality, the contributions of its architectural choices, its use of cross-channel correspondence, and its performance when trained on longer sequences.

5.1 Experimental Setup

Data and baselines. The main benchmark uses windows of length $T = 24$ on Sines, Stocks, ETTh1, MuJoCo, Energy, and fMRI (Yoon et al., 2019; Zhou et al., 2021; Tassa et al., 2018; Candanedo, 2017; Smith et al., 2011). We compare eight baselines: TimeGAN, Fourier Flows, unconditional CSDI, TimeVAE, KoVAE, Diffusion-TS, FlowTS, and PaD-TS (Yoon et al., 2019; Alaa et al., 2021; Tashiro et al., 2021; Desai et al., 2021; Naiman et al., 2024; Yuan & Qiao, 2024; Hu et al., 2025; Li et al., 2025). Experiments were run on a remote server equipped with an NVIDIA RTX 5090 GPU.

Our primary goal is unconditional modeling of the observed time-series corpus. Following the full-corpus evaluation protocol used by prior unconditional time-series generators (Yoon et al., 2019; Yuan & Qiao, 2024), all methods are trained on the same preprocessed corpus and generate as many windows as that corpus for evaluation. Generated samples are compared against this common real corpus. The resulting scores measure unconditional corpus fidelity and should not be interpreted as evidence of held-out generalization. Appendices B.1–B.2 provide the dataset and implementation protocols.

Table 1: Generation quality ($T = 24$; three training seeds). Mean ± 95% CI half-width. Lower is better; bold marks the lowest mean.

Metric	Methods	Sines	Stocks	ETTh1	MuJoCo	Energy	fMRI
Context-FID Score (Lower is Better)	TimeGAN	0.058±.013	0.220±.040	0.833±.179	0.379±.052	1.863±.151	0.151±.021
	Fourier Flows	0.660±.241	0.617±.120	0.342±.046	0.678±.125	0.409±.052	0.090±.014
	CSDI	0.020±.005	0.119±.032	0.054±.010	0.052±.006	0.045±.008	0.229±.025
	TimeVAE	0.036±.010	0.218±.046	0.619±.093	0.132±.023	0.850±.017	5.397±.581
	KoVAE	0.015±.005	0.057±.007	0.624±.079	0.108±.015	0.346±.068	0.715±.057
	Diffusion-TS	0.013±.005	0.265±.056	0.149±.014	0.022±.004	0.106±.019	0.049±.004
	FlowTS	0.072±.018	0.750±.183	0.101±.012	0.245±.034	0.210±.025	0.239±.020
	PaD-TS	0.013±.003	1.164±.307	0.692±.047	0.079±.010	0.143±.027	0.040±.005
	LIFT (Ours)	**0.004±.002**	**0.006±.001**	**0.012±.002**	**0.010±.002**	**0.016±.002**	**0.035±.005**
Discriminative Score (Lower is Better)	TimeGAN	0.006±.008	0.104±.039	0.165±.018	0.173±.022	0.436±.081	0.488±.006
	Fourier Flows	0.071±.011	0.144±.025	0.148±.061	0.353±.048	0.351±.095	0.369±.034
	CSDI	0.023±.014	0.043±.008	0.029±.007	0.032±.007	0.237±.033	0.293±.048
	TimeVAE	0.010±.006	0.075±.082	0.205±.071	0.086±.023	0.500±.001	0.307±.235
	KoVAE	0.016±.013	0.027±.032	0.169±.013	0.086±.025	0.274±.052	0.487±.020
	Diffusion-TS	0.032±.017	0.097±.012	0.076±.010	0.013±.006	0.122±.005	0.188±.085
	FlowTS	0.032±.006	0.105±.039	0.033±.021	0.159±.073	0.278±.151	0.367±.022
	PaD-TS	0.009±.008	0.122±.013	0.129±.013	0.083±.013	0.138±.020	0.141±.033
	LIFT (Ours)	**0.004±.004**	**0.006±.003**	**0.005±.004**	**0.005±.003**	**0.031±.021**	**0.135±.025**
MDD Score (Lower is Better)	TimeGAN	0.772±.018	0.598±.024	0.398±.008	0.634±.016	0.975±.008	0.213±.002
	Fourier Flows	1.154±.014	0.639±.015	0.361±.008	0.682±.012	0.436±.006	0.225±.002
	CSDI	0.651±.010	0.404±.039	0.202±.007	0.344±.017	0.256±.010	0.242±.003
	TimeVAE	1.041±.021	0.442±.030	0.553±.007	0.591±.031	0.449±.004	2.165±.007
	KoVAE	0.736±.019	0.523±.022	0.283±.006	0.556±.008	0.347±.004	0.441±.003
	Diffusion-TS	0.758±.033	0.486±.026	0.204±.007	0.416±.017	0.248±.008	0.196±.003
	FlowTS	1.062±.018	0.586±.028	0.231±.007	0.620±.013	0.398±.004	0.213±.003
	PaD-TS	0.728±.012	0.616±.034	0.261±.008	0.411±.009	0.309±.005	0.192±.002
	LIFT (Ours)	**0.586±.014**	**0.346±.027**	**0.172±.004**	**0.310±.008**	**0.232±.006**	**0.183±.002**
FDDS Score (Lower is Better)	TimeGAN	0.015±.004	0.093±.017	0.119±.008	0.228±.008	0.519±.008	0.083±.001
	Fourier Flows	0.313±.011	2.257±.108	0.235±.021	0.390±.009	0.689±.011	0.036±.000
	CSDI	0.002±.001	0.146±.007	0.044±.008	0.015±.002	0.103±.003	0.007±.000
	TimeVAE	0.007±.001	1.099±.074	0.611±.035	0.044±.005	0.480±.004	1.871±.002
	KoVAE	0.002±.000	0.227±.015	0.144±.018	0.047±.005	0.163±.005	0.231±.001
	Diffusion-TS	0.002±.000	0.197±.017	0.080±.009	0.019±.003	0.186±.004	0.006±.000
	FlowTS	0.092±.005	4.527±.052	0.060±.004	0.901±.019	0.822±.013	0.016±.000
	PaD-TS	**0.001±.000**	0.283±.027	0.112±.012	0.021±.003	0.071±.001	0.005±.000
	LIFT (Ours)	0.001±.000	**0.029±.006**	**0.005±.002**	**0.012±.001**	**0.047±.002**	**0.004±.000**

Evaluation. Context-FID, Discriminative, MDD, and FDDS measure representation fidelity, distinguishability, marginal distributions, and pairwise correlation distributions, respectively; lower is better. Each condition uses three training seeds and five evaluator repeats per checkpoint. Tables report the mean and 95% CI half-width across training-seed means. Diagnostic-repeat intervals are identified separately. Rankings use unrounded means where available; the local-loss ablation is compared at its reported three-decimal precision. Mean rankings are descriptive rather than significance tests. Appendix B.4 defines the metrics and uncertainty estimates.

5.2 RQ1: Overall Generation Quality

LIFT achieves the lowest three-seed mean in 23 of the 24 dataset–metric comparisons and ranks second in the remaining Sines FDDS comparison (Table 1). It obtains the lowest mean Context-FID, Discriminative, and MDD scores on all six datasets, together with the lowest FDDS on five of six. These gains span individual-channel distributions and the evaluated cross-channel correlation distributions.

The magnitude of the improvement varies across datasets. On Stocks, Context-FID decreases to 0.006 from 0.057 for KoVAE, the strongest baseline on that metric, whereas leading methods are much closer on Sines. On fMRI, LIFT has lower Context-FID and Discriminative means than PaD-TS, although their confidence intervals overlap. We next use controlled variants to examine the design choices behind these results. Appendix H provides complementary visualizations.

5.3 RQ2: Which Design Choices Matter?

We compare Full LIFT (Full) with controls for temporal context and state reuse, output organization and local supervision, and the interaction mechanism. Controls share the corpus, three training seeds, 15,000-update budget, optimizer, sampler, and evaluators, changing the designated compo-

nent. Tables 2–3 report Context-FID and FDDS on five datasets. Sines offers little separation at the displayed precision; Appendix E reports all six datasets and the complete available metrics.

Temporal context. Temporally informed interaction substantially improves generation quality over the pointwise-interaction control. *Parallel (PWI)* retains access to other channels and sample-level conditioning but supplies the refiner with pointwise embeddings; *Local Only* removes interaction altogether. Full has lower unrounded mean Context-FID and FDDS than both controls on all five non-Sines datasets. For example, Stocks FDDS decreases from 3.914 for Local Only and 2.484 for PWI to 0.029 for Full. Access to other channels alone is therefore insufficient in this control: the temporal representation supplied to interaction also matters.

State reuse. Separate temporal encoders recover much of the gap between PWI and Full, indicating that temporal context is the dominant factor in these controls. Reusing the locally supervised states nevertheless provides a further advantage. *Separate 2+2* matches Full's total temporal-block count by using two blocks per path, while *Separate 4+4* matches its four-block depth per path. Across the complete four-metric, six-dataset evaluation, Full has lower means than Separate 2+2 in 24/24 comparisons and Separate 4+4 in 22/24; Separate 4+4 has a lower ETTh1 Discriminative mean and ties MuJoCo FDDS. These results support state reuse even when the interaction path is supplied with its own temporal encoder.

Table 2: Temporal-state access, reuse, output organization, and local supervision ($T = 24$; three training seeds). Mean $\pm$ 95% CI half-width; lower is better.

Variant	Stocks	ETTh1	MuJoCo	Energy	fMRI
Context-FID $\downarrow$					
Full LIFT	0.006 ± 0.001	0.012 ± 0.002	0.010 ± 0.002	0.016 ± 0.002	0.035 ± 0.005
Local Only	4.252 ± 0.555	0.851 ± 0.150	0.943 ± 0.176	2.612 ± 0.336	0.174 ± 0.048
Parallel (PWI)	0.281 ± 0.022	0.414 ± 0.052	0.809 ± 0.154	1.407 ± 0.283	0.061 ± 0.010
Separate 2+2	0.008 ± 0.001	0.015 ± 0.002	0.013 ± 0.002	0.018 ± 0.002	0.045 ± 0.005
Separate 4+4	0.007 ± 0.002	0.014 ± 0.002	0.012 ± 0.002	0.017 ± 0.002	0.047 ± 0.006
Direct output	0.008 ± 0.002	0.014 ± 0.002	0.012 ± 0.002	0.021 ± 0.003	0.050 ± 0.007
w/o local loss	0.017 ± 0.002	0.037 ± 0.002	0.010 ± 0.002	0.016 ± 0.002	0.043 ± 0.005
FDDS $\downarrow$					
Full LIFT	0.029 ± 0.006	0.005 ± 0.002	0.012 ± 0.001	0.047 ± 0.002	0.004 ± 0.000
Local Only	3.914 ± 0.097	1.095 ± 0.022	0.092 ± 0.008	0.199 ± 0.003	0.110 ± 0.001
Parallel (PWI)	2.484 ± 0.092	0.173 ± 0.010	0.068 ± 0.004	0.119 ± 0.004	0.004 ± 0.000
Separate 2+2	0.033 ± 0.005	0.006 ± 0.002	0.014 ± 0.001	0.060 ± 0.002	0.004 ± 0.000
Separate 4+4	0.031 ± 0.003	0.007 ± 0.002	0.012 ± 0.001	0.057 ± 0.002	0.004 ± 0.000
Direct output	0.036 ± 0.005	0.007 ± 0.002	0.016 ± 0.002	0.057 ± 0.002	0.005 ± 0.000
w/o local loss	0.048 ± 0.005	0.005 ± 0.002	0.031 ± 0.001	0.047 ± 0.002	0.008 ± 0.000

Direct output retains local supervision; w/o local loss sets $\lambda_{\text{loc}} = 0$. Sines and all available metrics are in Table 8.

Output organization and local supervision. *Direct output* retains the architecture and local auxiliary objective but predicts the complete field through the interaction head alone. Full instead adds the head's prediction to the local velocity. Full has lower means in 23/24 quality comparisons, tying Sines Context-FID, and lower ACF-MAE and LagCCF-MAE on all six datasets. This supports the additive organization under the shared training setup. The change also affects the initial complete field and the joint-loss gradient to the local head, so the comparison includes these optimization consequences (Appendix E).

Removing the local auxiliary loss increases mean Context-FID on Stocks, ETTh1, and fMRI and mean FDDS on Stocks, MuJoCo, and fMRI. For example, ETTh1 Context-FID rises from 0.012 to 0.037, and MuJoCo FDDS from 0.012 to 0.031. The remaining six dataset–metric pairs have equal means at the reported precision.

Temporal and cross-channel fidelity. The lagged statistics distinguish within-channel temporal fidelity from cross-channel alignment. Local Only achieves lower ACF-MAE than Full on ETTh1

and MuJoCo, yet has higher LagCCF-MAE on every dataset. Full also lowers both errors relative to PWI across all six datasets. These results reinforce the distinction motivating LIFT: accurate within-channel autocorrelation does not by itself ensure accurate cross-channel timing. The statistics cover lags 1–8 and complement FDDS, which compares within-window correlation distributions. Definitions and complete results appear in Appendices B.5 and E.

Interaction mechanism. Carrier messages improve on mean-only exchange. *Mean Only* retains the mean context and gated updates but removes carriers. Full lowers mean FDDS on all six datasets and Context-FID on four, with Context-FID ties on Sines and Stocks before rounding. We also replace the exchange operator with channel attention and STAR-style aggregation (Han et al., 2024) while retaining the local backbone, hidden width, and training recipe. These quality controls use closely matched parameter budgets (Appendix E). Full has lower ACF-MAE and LagCCF-MAE than both replacements on every dataset, although channel attention obtains a lower MuJoCo FDDS of 0.009 versus 0.012. Thus, no interaction operator dominates every metric, but mean–carrier performs consistently across the evaluated distributional and lagged statistics. Appendix E.1 further distinguishes retrained Mean Only from carrier removal in a fixed trained model.

Table 3: Interaction-module comparison ($T = 24$; three training seeds). Mean $\pm$ 95% CI half-width; lower is better.

Variant	Stocks	ETTh1	MuJoCo	Energy	fMRI
Context-FID ↓					
Full LIFT	0.006 ± 0.001	0.012 ± 0.002	0.010 ± 0.002	0.016 ± 0.002	0.035 ± 0.005
Mean Only	0.006 ± 0.001	0.013 ± 0.003	0.013 ± 0.002	0.021 ± 0.003	0.050 ± 0.007
Channel attention	0.007 ± 0.001	0.013 ± 0.003	0.012 ± 0.002	0.021 ± 0.002	0.047 ± 0.005
STAR-style	0.006 ± 0.001	0.019 ± 0.005	0.032 ± 0.005	0.059 ± 0.010	0.068 ± 0.012
FDDS ↓					
Full LIFT	0.029 ± 0.006	0.005 ± 0.002	0.012 ± 0.001	0.047 ± 0.002	0.004 ± 0.000
Mean Only	0.038 ± 0.007	0.007 ± 0.002	0.015 ± 0.002	0.061 ± 0.002	0.009 ± 0.000
Channel attention	0.040 ± 0.007	0.008 ± 0.002	0.009 ± 0.001	0.060 ± 0.002	0.006 ± 0.000
STAR-style	0.063 ± 0.007	0.014 ± 0.003	0.021 ± 0.002	0.080 ± 0.003	0.016 ± 0.000

Controls retain the local backbone and training recipe. Complete results and settings: Appendix E.

5.4 RQ3: Does LIFT Use Cross-Channel Correspondence?

To test whether the learned joint field uses cross-channel information beyond the target trajectory itself, we apply paired context interventions to the three trained Full checkpoints on fixed training-corpus subsets. We keep the trained model, target trajectory, flow time, and velocity target unchanged while replacing the other channels with a donor block. *Original* retains the sample. *Random* assigns donors randomly, while *Matched* selects donors with similar visible target-channel trajectories at the same flow time. Because the local prediction remains unchanged, changes in the complete velocity prediction arise through the interaction refiner. We measure target-channel velocity MSE; Appendix F gives the full assignment and evaluation protocol.

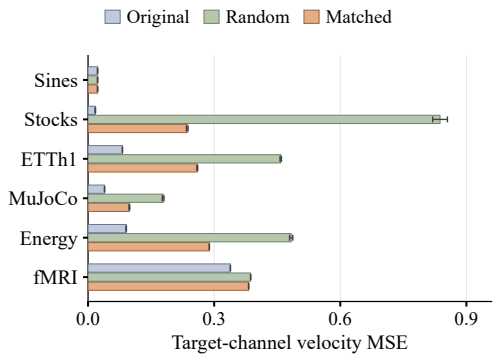

Figure 2: Target-channel velocity MSE under context replacement, with the target trajectory fixed. Error bars are diagnostic-repeat 95% CIs; Table 14 reports paired changes.

Matched replacement increases mean error on all five non-Sines datasets, with larger increases under Random (Figure 2). For fMRI, the Matched increase is 0.043, or 12.8%, whereas

independently generated Sines shows a near-zero effect of 0.012%. Matched donors reduce donor-to-receiver target-trajectory distance by 58–65% relative to Random while still substantially changing the other channels. The error increase therefore persists when the donor's visible target trajectory is closer to the receiver's. These interventions show that the learned field is sensitive to cross-channel correspondence while the target trajectory is held fixed. Replacement can create out-of-distribution joint samples, so the measurements quantify field sensitivity rather than gains in generated-sample quality.

5.5 RQ4: Generation at Longer Sequence Lengths

LIFT retains strong relative performance when trained on longer windows. We train LIFT, PWI, Diffusion-TS, FlowTS, and PaD-TS separately at $T \in \{64, 128, 256\}$ on ETTh1 and Energy, using three training seeds per setting. LIFT obtains the lowest mean in 11/12 Context-FID and FDDS comparisons, including all ETTh1 settings (Table 4). The exception is Energy FDDS at $T = 256$, where PaD-TS scores 0.011 versus 0.022 for LIFT. LIFT also outperforms PWI at every tested length across all four metrics, extending the evidence for temporally informed interaction beyond $T = 24$. Appendix G reports the complete results.

Strong relative rankings coexist with increasing absolute errors in some settings: Energy Context-FID rises from 0.032 at $T = 64$ to 0.106 at $T = 256$. These experiments evaluate separately trained models at each length, rather than extrapolation to unseen lengths. Native-sampler timing results and evaluation counts are reported in Appendix G.2.

Table 4: Long-sequence Context-FID and FDDS over three training seeds. Full metrics: Appendix G. Mean ± 95% CI half-width; lower is better. Bold marks the lowest mean.

Method	ETTh1			Energy		
	$T = 64$	$T = 128$	$T = 256$	$T = 64$	$T = 128$	$T = 256$
Context-FID ↓						
LIFT	**0.038**±.004	**0.078**±.005	**0.144**±.006	**0.032**±.006	**0.066**±.012	**0.106**±.023
PWI	0.954±.042	1.576±.201	2.314±.151	1.865±.137	2.388±.298	3.730±.570
Diffusion-TS	0.271±.030	0.374±.048	2.559±.110	0.111±.010	0.092±.010	0.130±.005
FlowTS	0.326±.068	0.482±.071	0.638±.198	0.394±.070	0.586±.065	1.356±.133
PaD-TS	1.906±.237	3.317±.460	4.765±.625	0.173±.014	0.126±.009	0.119±.011
FDDS ↓						
LIFT	**0.007**±.002	**0.010**±.001	**0.019**±.002	**0.012**±.001	**0.014**±.001	0.022±.002
PWI	0.259±.009	0.330±.015	0.417±.007	0.209±.005	0.357±.004	0.466±.007
Diffusion-TS	0.090±.005	0.073±.010	0.179±.005	0.048±.003	0.031±.001	0.035±.000
FlowTS	0.044±.006	0.054±.005	0.045±.004	0.316±.004	0.133±.002	0.076±.003
PaD-TS	0.242±.010	0.329±.015	0.341±.011	0.037±.002	0.016±.001	**0.011**±.001

Rankings use means before display rounding.

Scope. The fMRI preprocessing includes windows crossing simulated-subject boundaries, and Energy contains two identical columns. The Energy correspondence effect remains positive when the two duplicated target channels are excluded from scoring, with all input channels retained (Appendix F). Comparisons with complete generators also reflect their prescribed training and sampling recipes, while architectural controls hold the specified LIFT settings fixed (Appendices B.2 and E).

6 Conclusion

We introduced LIFT, a flow-matching framework that reuses locally supervised temporal representations for joint multivariate refinement. The same channel-local temporal states support both local velocity prediction and an additive interaction-conditioned correction, with compact cross-channel communication provided by the mean–carrier module. Across six benchmarks, LIFT shows strong generation performance, while controlled variants demonstrate substantial benefits from temporally informed interaction, with state reuse yielding lower mean scores than separate temporal encoding in most comparisons. Correspondence interventions further show that the learned joint field responds to changes in cross-channel alignment while the target trajectory is held fixed. Taken together, these results support a design principle for multivariate time-series generation: train temporal states for channel-local prediction and reuse them for joint refinement.

AI USE STATEMENT

Portions of the manuscript were revised with assistance from ChatGPT (OpenAI) for language refinement and readability improvement. The authors reviewed, edited, and verified all AI-assisted content and take full responsibility for the correctness and originality of the final manuscript.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the implementation of LIFT and the configurations for the main experiments are available in an anonymized repository at `https://anonymous.4open.science/r/Lift-8ECB/README.md`. The repository also provides environment requirements and instructions for data preparation, training, sampling, and evaluation.

REFERENCES

Ahmed Alaa, Alex James Chan, and Mihaela van der Schaar. Generative time-series modeling with fourier flows. In *International Conference on Learning Representations*, 2021.

Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-Based Time Series Imputation and Forecasting with Structured State Space Models. *Transactions on Machine Learning Research*, 2023. URL https://mlanthology.org/tmlr/2023/alcaraz2023tmlr-diffusionbased/.

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

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

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 35971–35983. Curran Associates, Inc., 2022. doi: 10.52202/068431-2607. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e9a32fade47b906de908431991440f7c-Paper-Conference.pdf.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/754612bde73a8b65ad8743f1f6d8ddf6-Abstract-Conference.html.

Yang Hu, Xiao Wang, Zezhen Ding, Lirong Wu, Huatian Zhang, Stan Z. Li, Sheng Wang, Jiheng Zhang, Ziyun Li, and Tianlong Chen. Flowts: Time series generation via rectified flow, 2025. URL https://arxiv.org/abs/2411.07506.

Yang Li, Han Meng, Zhenyu Bi, Ingolv T. Urnes, and Haipeng Chen. Population aware diffusion for time series generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17): 18520–18529, Apr. 2025. doi: 10.1609/aaai.v39i17.34038. URL https://ojs.aaai.org/index.php/AAAI/article/view/34038.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/liu2023iclr-flow/.

Ilan Naiman, N. Benjamin Erichson, Pu Ren, Michael W. Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via koopman VAEs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eY7sLb0dVF.

Jian Qian, Bingyu Xie, Biao Wan, Minhao Li, Miao Sun, and Patrick Yin Chiang. Timeldm: Latent diffusion model for unconditional time series generation, 2024. URL https://arxiv.org/abs/2407.04211.

Lifeng Shen, Kai Syun Hou, Weiyu Chen, and James Kwok. Latent-to-data cascaded diffusion models for unconditional time series generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 35058–35079, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3b246503dce7d5f1b55d52ff43123e42-Paper-Conference.pdf.

Stephen M. Smith, Karla L. Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F. Beckmann, Thomas E. Nichols, Joseph D. Ramsey, and Mark W. Woolrich. Network modelling methods for fmri. *NeuroImage*, 54(2):875–891, 2011. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2010.08.063. URL https://www.sciencedirect.com/science/article/pii/S1053811910011602.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/cfe8504bda37b575c70ee1a8276f3486-Paper.pdf.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018. URL https://arxiv.org/abs/1801.00690.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport. *Transactions on Machine Learning Research*, 2024. URL https://mlanthology.org/tmlr/2024/tong2024tmlr-improving/.

Zihao Yao, Jiankai Zuo, and Yaying Zhang. Dim-ts: bridge the gap between selective state space models and time series for generative modeling. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i19.38649. URL https://doi.org/10.1609/aaai.v40i19.38649.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 41582–41610, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b5b66077d016c037576cc56a82f97f66-Paper-Conference.pdf.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pp. 11106–11115. AAAI Press, 2021.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42625–42643. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhou23i.html.

A Temporal Context for Interaction Refinement

Figure 3 illustrates why relative-timing corrections can depend on temporal context. Here ξ is a continuous sequence coordinate, while t remains flow time.

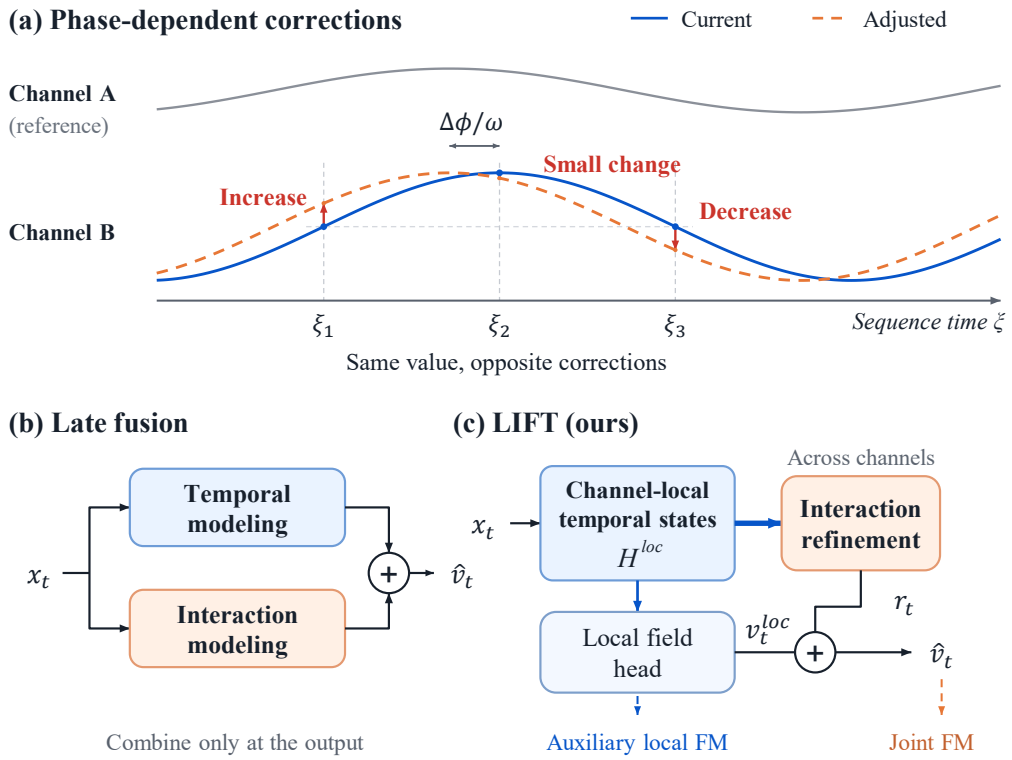

Figure 3: Temporal context for interaction refinement. (a) The same small phase shift yields opposite corrections at equal-valued positions with opposite slopes, and a small first-order correction near an extremum. (b) A schematic output-fusion design combines branch predictions. (c) LIFT conditions interaction residuals on local temporal states and multivariate context. Auxiliary and joint flow-matching losses supervise the local and summed fields, respectively. Panel (a) is an analytic illustration; flow-time conditioning is omitted from (b) and (c) for clarity.

A phase-adjustment example. Consider a periodic channel $x(\xi) = \sin(\omega\xi)$. Applying a small phase shift $\Delta\phi$ to adjust its relation to another channel gives

$$\Delta x(\xi) = \sin(\omega\xi + \Delta\phi) - \sin(\omega\xi) = \Delta\phi\cos(\omega\xi) + \mathcal{O}\big((\Delta\phi)^2\big). \tag{9}$$

At $\omega\xi = 0$ and $\omega\xi = \pi$, the channel has the same value but opposite slopes, so the first-order corrections have opposite signs. At an extremum, the first-order term vanishes, although a finite shift can still produce a second-order change. This example shows why the current point value alone does not determine the correction: temporal context is also needed. The illustrated displacement is a conceptual example, rather than a measured flow-matching velocity.

The example motivates temporal context, including in separately encoded variable branches; it does not establish an expressivity gap or distinguish mean–carrier exchange from other contextual modules. Panels (b,c) illustrate output fusion and LIFT's shared-state refinement, respectively.

B Experimental Details

This appendix specifies the data, baseline implementations, and evaluation protocols used in Section 5.

B.1 DATASETS AND PREPROCESSING

We use the six dataset recipes and archived files distributed with Diffusion-TS (Yuan & Qiao, 2024) (Table 5). Sines and MuJoCo contain independently generated trajectories (data seed 123); the remaining datasets use stride-one windows of length $T = 24$ in the main benchmark. Longer windows are evaluated in Appendix G. Energy retains all 28 numeric columns, including `rv1`/`rv2`. The fMRI benchmark contains simulated BOLD signals from NetSim Simulation 4: 50 simulated subjects with 200 observations each are concatenated before windowing, including windows that cross subject boundaries.

Table 5: Main benchmark datasets. All windows have length $T = 24$; C is the channel count and N the number of training windows. Each method generates exactly N windows for evaluation.

Dataset	Domain	C	N
Sines	Independent sinusoids	5	10,000
Stocks	Stock prices and volume	6	3,662
ETTh1	Transformer load and temperature	7	17,397
MuJoCo	Hopper physics simulation	14	10,000
Energy	Appliances and environmental sensors	28	19,712
fMRI	Simulated BOLD signals	50	9,977

Per-channel min–max scaling uses the complete corpus (the known range for Sines). LIFT operates in $[-1, 1]$; all methods are evaluated in common $[0, 1]$ coordinates without additional output clipping. Generators train on the full corpus, which also supplies the real evaluation reference; this protocol measures fidelity to that corpus, without a held-out generator test set.

B.2 BASELINES

All baselines use the same corpus and evaluators. Their architectures and budgets follow their respective recipes; training compute is not matched. We retain the following implementation-specific choices.

Adversarial, flow, and latent-variable baselines. TimeGAN (Yoon et al., 2019) uses a PyTorch GRU port whose cells and initialization differ from the original TensorFlow version. Fourier Flows (Alaa et al., 2021) uses a float64 CUDA port of version 0.0.2, jointly modeling flattened multivariate windows with the source Fourier-packing zero padding. TimeVAE (Desai et al., 2021) uses a PyTorch port with the residual decoder enabled and no polynomial trend or custom seasonality. KoVAE (Naiman et al., 2024) uses its regular-data model through a device-neutral port and the shared data adapter.

Diffusion and flow-matching baselines. Unconditional CSDI (Tashiro et al., 2021) generates complete windows without observed-value conditioning: four residual blocks, width 64, eight heads, 200 epochs, and batch size 16; generation uses final weights and 50 DDPM steps. Diffusion-TS (Yuan & Qiao, 2024) retains its official model, trainer, and sampler, with shared data loading and output conversion. FlowTS (Hu et al., 2025) retains its official model and objective with an adapted trainer (absent from the archived source); generation uses 100 Euler evaluations on its shifted grid ($\alpha = 3$). PaD-TS (Li et al., 2025) retains its official model, loss, and ancestral sampler, including sampler clipping; ETTh1 uses the Stocks recipe with seven input channels.

B.3 INFORMATION PATHS IN THE COMPARED ARCHITECTURES

The architectures differ in which representation supplies cross-channel computation (Table 6). CSDI passes temporal-attention outputs to feature attention; PaD-TS uses a separate variable branch. In LIFT, interaction reads states that also support a supervised velocity prediction using only the target channel's observations. PWI denotes the Pointwise-Interaction control from RQ2.

PaD-TS embeds complete windows in its variable branch, while CSDI's feature attention receives temporal-attention outputs. Both therefore access temporal context. PWI retains LIFT's multivariate

Table 6: Information paths in the evaluated architectures. The final column asks whether cross-channel computation reuses states of an explicit channel-local velocity predictor.

Architecture	Input to cross-channel computation	Explicit local field	Reuses its states
CSDI	Temporal-attention outputs within each residual block	No	No
PaD-TS	Window embeddings in a separate variable branch	No	No
PWI	Pointwise input embeddings and global statistics	Yes	No
LIFT	Local predictor's trajectory states and global statistics	Yes	Yes

statistics, mean pathway, and carriers but replaces temporal states with pointwise embeddings. The Separate controls in RQ2 distinguish context access from sharing a local predictor's states. Module replacements separately test the exchange operator; complete-generator benchmarks also reflect differences in training and sampling recipes.

B.4 Metrics and uncertainty

In each repeat, $X_i, \widetilde{X}_i \in \mathbb{R}^{C\times T}$, $i = 1, \dots, n$, denote real/generated windows. All metrics are lower-is-better, with shared evaluator settings.

Context-FID. A TS2Vec-style encoder, fitted to real data for 600 updates per repeat, produces 320-dimensional max-pooled features. Their empirical means μ_r, μ_g and sample covariances Σ_r, Σ_g give

$$\text{Context-FID} = \|\mu_r - \mu_g\|_2^2 + \operatorname{tr}\Big(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\Big). \tag{10}$$

Discriminative. A GRU classifier is trained for 2,000 updates on an internal 80/20 real/generated split. For its M held-out examples with true labels y_j and predictions $\widehat{y}_j$ thresholded at 0.5, we report

$$\text{Disc} = \left| \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}\{\widehat{y}_j = y_j\} - \frac{1}{2} \right|. \tag{11}$$

MDD. For each (c, τ), use $B = 50$ equal-width bins over the real range. Real/generated densities $p^r_{c\tau b}, p^g_{c\tau b}$ are bin counts divided by $n\Delta_{c\tau}$, with bin width $\Delta_{c\tau}$:

$$\text{MDD} = \frac{1}{BC(T-1)} \sum_{c=1}^{C}\sum_{\tau=2}^{T}\sum_{b=1}^{B} |p^r_{c\tau b} - p^g_{c\tau b}|. \tag{12}$$

The first position is excluded and density differences are not multiplied by bin width. The implementation uses strict generated-bin endpoints and extends constant real ranges by 10^{-5}.

FDDS. For each channel pair $a < b$, define within-window Pearson correlations $r_i^{ab} = \text{Corr}_\tau(X_{i,a,:}, X_{i,b,:})$ and g_i^{ab} analogously; zero-variance channels have zero correlation. Following PaD-TS (Li et al., 2025), we average biased squared MMDs over channel pairs:

$$\text{FDDS} = \frac{2}{C(C-1)} \sum_{a<b} \widehat{\text{MMD}}_k^2(r^{ab}, g^{ab}), \tag{13}$$

$$\widehat{\text{MMD}}_k^2(r, g) = \frac{1}{n^2}\sum_{i,j=1}^{n} \big[k(r_i, r_j) + k(g_i, g_j) - 2k(r_i, g_j)\big]. \tag{14}$$

Here $k(u, v) = \sum_{\ell\in\mathcal{B}} \exp[-(u-v)^2/(2\ell)]$, with $\mathcal{B} = \{0.01, 0.05, 0.1, 0.2, 0.5, 0.7, 1, 1.5, 2\}$. Diagonal kernel terms are included, with no square root; the score compares pairwise correlation distributions rather than the full joint law.

Uncertainty and numerical precision. Generation scores use three independently trained models (seeds 1, 2, and 3) and five evaluator repeats per model. For score $m_{s,j}$, we first average the repeats within each training seed, $\bar{m}_s = \frac{1}{5}\sum_{j=1}^{5} m_{s,j}$, and then average the three $\bar{m}_s$. Generation-score intervals are Student-t 95% intervals across the training-seed means (two degrees of freedom); evaluator repeats are not treated as independent training runs. Fixed-model diagnostics also report three-seed means. For RQ3, each diagnostic repeat first averages results across the three checkpoints; intervals are then computed over the five repeat-level averages. These intervals describe intervention variability conditional on the fitted checkpoints, not variation across independently trained models.

Context-FID/Discriminative use all N windows; MDD/FDDS independently bootstrap $\lfloor N/5 \rfloor$ windows from each population, sharing draws between metrics. Temporal-statistics tables report means of deterministic per-checkpoint estimates without evaluator-repeat intervals. Quality scores and CI half-widths are displayed to three decimals. Comparisons use unrounded means except for the local-loss ablation, whose supplied aggregates have three-decimal precision. Parameter counts are integers or millions with three decimals, latency uses milliseconds or seconds with two decimals, and memory uses MiB with one decimal.

B.5 Temporal and lagged cross-channel statistics

We use the canonical real/generated samples from RQ2, computing each statistic per checkpoint and averaging across three training seeds. For each lag $\ell \in \{1, \dots, 8\}$, define

$$R^r_\ell(a,b) = \mathrm{Corr}_{n,\tau}\big(x^r_{n,\tau,a}, x^r_{n,\tau+\ell,b}\big), \qquad n = 1,\dots,N, \quad \tau = 1,\dots,T-\ell. \tag{15}$$

Each endpoint is centered and normalized over these valid pairs; no pair crosses a window boundary. A zero-variance endpoint gives zero correlation. Defining R^g_ℓ analogously, we report

$$\text{ACF-MAE} = \frac{1}{8C}\sum_{\ell=1}^{8}\sum_{a=1}^{C} |R^r_\ell(a,a) - R^g_\ell(a,a)|, \tag{16}$$

$$\text{LagCCF-MAE} = \frac{1}{8C(C-1)}\sum_{\ell=1}^{8}\sum_{a\neq b} |R^r_\ell(a,b) - R^g_\ell(a,b)|. \tag{17}$$

Ordered channel pairs retain both directions at positive lags. All statistics use float64. These are deterministic full-corpus point estimates per training seed, averaged across three trained models without evaluator-repeat confidence intervals. Pooled correlations can include shared position-dependent trends and do not identify conditional dependence or causal direction.

Tables 9 and 12 report all available results. Full lowers both errors relative to PWI on all six datasets. Local Only retains lower ACF-MAE on ETTh1 and MuJoCo despite substantially worse LagCCF-MAE, distinguishing temporal fidelity within a channel from timing relationships across channels.

C LIFT Implementation and Complexity

Table 7 summarizes the saved LIFT configurations underlying RQ1. RQ2 uses this same shared Full LIFT configuration for all datasets, changing only the designated component.

Parameterization. Stages have separate token-wise MLPs ($d \to 2d \to d$) and share the mean–carrier bottleneck and gated update. The carrier gate is a sigmoid MLP ($5 \to d/2 \to K$). It receives four channel-averaged statistics and path log-SNR, $2\log[\bar{t}/(1-\bar{t})]$, where $\bar{t} = \mathrm{clip}(t, 10^{-5}, 1 - 10^{-5})$. The learned $\tanh$ carrier scale and residual head are zero-initialized. Appendix C.1 specifies routing and updates.

Training and generation. Algorithm 1 summarizes the procedure; MSE averages over batch, time, and channels. The local weight is 1 for the first 1,000 updates, decreases linearly to 0.25 at update 10,000, and remains fixed thereafter. Sampling uses seed 303 and returns windows in model coordinates, without OT or output clipping.

Table 7: Key LIFT settings for RQ1, shared across all six datasets.

Setting	Value
Hidden width / local Bi-S4D blocks	128 / 4
State dimension / channel embedding	64 / 32
Interaction stages / carriers	2 / 3
Local / interaction dropout	0 / 0.1
Physical batch / gradient accumulation	64 / 2
Optimizer updates / warmup	15,000 / 1,000
AdamW / gradient norm cap	$\beta = (0.9, 0.99)$, $\epsilon = 10^{-8}$ / 1.0
Learning rate (ordinary / state dynamics)	3×10^{-4} / 10^{-4}
Weight decay (ordinary / state dynamics)	10^{-4} / 0
EMA decay / generation	0.999 / 100-step FP32 Euler

Algorithm 1 LIFT training and generation

Require: Corpus $\mathcal{D}$ of N windows; batch $B = 64$; accumulation $A = 2$; updates $U = 15{,}000$; EMA decay $\beta = 0.999$.

Training

Initialize θ; set $\bar{\theta} \leftarrow \theta$ and $u \leftarrow 0$.
while $u < U$ **do**
  $g \leftarrow 0$.
  for $a = 1, \ldots, A$ **do**
    Draw B data windows $X_1^{(i)}$ from $\mathcal{D}$; draw $X_0^{(i)} \sim \mathcal{N}(0, I)$ and $t_i \sim \mathcal{U}(0, 1)$.
    $D_{ij} \leftarrow \|X_0^{(i)} - X_1^{(j)}\|_F^2$; $\pi \leftarrow \mathrm{Hungarian}(D)$.
    $X_{t_i}^{(i)} \leftarrow (1 - t_i)X_0^{(i)} + t_i X_1^{(\pi(i))}$; $U_i^\star \leftarrow X_1^{(\pi(i))} - X_0^{(i)}$.
    $(H, V_{\mathrm{loc}}) \leftarrow \mathrm{Local}_\theta(X_t, t)$.
    $R \leftarrow \mathrm{Interaction}_\theta(H, X_t, t)$; $V \leftarrow V_{\mathrm{loc}} + R$.
    $\ell \leftarrow \mathrm{MSE}(V, U^\star) + \lambda_{\mathrm{loc}}(u)\,\mathrm{MSE}(V_{\mathrm{loc}}, U^\star)$.
    $g \leftarrow g + A^{-1}\nabla_\theta \ell$.
  end for
  if any loss or gradient is nonfinite, discard this attempt and **continue**.
  Apply AdamW to θ using g clipped to norm 1 and the scheduled learning rates.
  $u \leftarrow u + 1$; $\bar{\theta} \leftarrow \beta\bar{\theta} + (1 - \beta)\theta$.
end while

Generation

Use the final $v_{\bar{\theta}}$ in evaluation mode and FP32; draw N Gaussian windows $X^{(0)}$.
for $j = 0, \ldots, 99$ **do**
  $X^{(j+1)} \leftarrow X^{(j)} + \frac{1}{100} v_{\bar{\theta}}(X^{(j)}, j/100)$.
end for
return $X^{(100)}$.

Computational scope. Section 4.3 gives the per-stage costs. The $O(TCKd^2)$ value projection is part of the full interaction computation; $O(CK)$ alone describes only routing size per position. The analysis excludes minibatch OT assignment during training. Measured native-sampler costs appear in Appendix G.2.

C.1 Refinement and conditioning details

The following equations specify the refinement in Section 4.3. At refinement stage m, a stage-specific, flow-conditioned token module prepares $\widetilde{\mathbf{H}}^{(m)}$. The shared bottleneck applies flow-conditioned modulation to obtain $\mathbf{N}^{(m)}$, then constructs mean and carrier context from these states. Each stage recomputes the messages from its updated inputs.

Routing dimensions and centering. The write and read maps are $\mathbf{W}_{\mathrm{wr}}, \mathbf{W}_{\mathrm{rd}} \in \mathbb{R}^{K\times d}$. The map $\mathbf{W}_{\mathrm{val}} \in \mathbb{R}^{Kd\times d}$ produces K width-d vectors; $[\mathbf{W}_{\mathrm{val}}\boldsymbol{\delta}]_k$ selects the kth vector after reshaping to $K \times d$. Affine biases are omitted in Eq. equation 5. For a fixed carrier, subtracting the channel mean from linear write logits subtracts the same scalar from every channel logit. The write softmax is therefore unchanged. Centering changes the transmitted values and can change the carrier-normalized read weights. This separation concerns hidden states; it does not assume identifiable common and idiosyncratic factors in the observations.

Sample- and time-conditioned gates. The gate $\rho_k(\mathbf{x}_t, t) \in (0, 1)$ uses channel averages of four statistics: the last value, window mean, within-channel standard deviation, and absolute first-to-last difference, together with path log-SNR. It is shared over sequence positions and refinement stages within a field evaluation and supplies multivariate numerical information only to the residual. The mean path remains available independently of the learned $\tanh$ carrier scale $\gamma(t)$. These gates allow exchange to vary with the path state without prescribing a fixed trend over flow time.

Gated update and residual head. Each receiving channel combines its own state with the received context. Restoring stage indices, the update with unit update scale is

$$\begin{aligned}
\mathbf{s}_{c,\tau}^{(m)} &= [\mathbf{N}_{c,\tau}^{(m)}; \mathbf{Z}_{c,\tau}^{\mathrm{ctx},(m)}],\\
\mathbf{g}_{c,\tau}^{(m)} &= \sigma(\mathbf{W}_g \mathbf{s}_{c,\tau}^{(m)} + \mathbf{b}_g),\\
\mathbf{H}_{c,\tau}^{(m)} &= \mathrm{LN}\Big(\widetilde{\mathbf{H}}_{c,\tau}^{(m)} + \mathbf{g}_{c,\tau}^{(m)} \odot \mathrm{MLP}_{\mathrm{upd}}(\mathbf{s}_{c,\tau}^{(m)})\Big).
\end{aligned} \tag{18}$$

Brackets denote concatenation, σ is the sigmoid, and layer normalization acts over the hidden dimension. After S stages,

$$\mathbf{r}_{\theta,c}^{\mathrm{int}} = \mathbf{H}_c^{(S)} \mathbf{w}_{\mathrm{int}} + b_{\mathrm{int}} \mathbf{1}_T. \tag{19}$$

The bottleneck and gated-update weights are shared across stages; outer token modules are stage-specific. All stages occur within one vector-field evaluation.

D Training Objective: Interpretation and Proof

We formalize the local and joint conditional regressions in Eq. equation 8 under a fixed coupling law.

Setting and population optimum. Let Q_π be the joint law of minibatch coupling and path-time sampling, and fix a channel c with $\mathbb{E}_{Q_\pi}\|\mathbf{U}_c\|_2^2 < \infty$. The local and full information sets are $\mathcal{F} = \mathcal{F}_c^{\mathrm{loc}} = \sigma(\mathbf{X}_{t,c}, t, c)$ and $\mathcal{G} = \mathcal{F}_c^{\mathrm{joint}} = \sigma(\mathbf{X}_t, t, c)$, with $\mathcal{F} \subseteq \mathcal{G}$. For unrestricted $f \in L^2(\mathcal{F}; \mathbb{R}^T)$ and $r \in L^2(\mathcal{G}; \mathbb{R}^T)$ and any fixed $\lambda > 0$, the channel-wise population objective is minimized by

$$\begin{aligned}
f_c^\star &= \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathcal{F}],\\
f_c^\star + r_c^\star &= \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathcal{G}],\\
r_c^\star &= \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathcal{G}] - \mathbb{E}_{Q_\pi}[\mathbf{U}_c \mid \mathcal{F}].
\end{aligned} \tag{20}$$

Thus the ideal residual is the increment between the joint and local conditional velocity estimates under the same coupling law.

Proof. All expectations below are under Q_π. Set $h = f + r$. Since $\mathcal{F} \subseteq \mathcal{G}$, any such f and any $h \in L^2(\mathcal{G}; \mathbb{R}^T)$ are feasible together by taking $r = h - f$. Hence f and h can be chosen independently. Write $m_\mathcal{F} = \mathbb{E}[\mathbf{U}_c \mid \mathcal{F}]$ and $m_\mathcal{G} = \mathbb{E}[\mathbf{U}_c \mid \mathcal{G}]$. Conditional-expectation orthogonality gives

$$\begin{aligned} \mathcal{R}_\lambda(f, r) &= \mathbb{E}\|h - \mathbf{U}_c\|_2^2 + \lambda \mathbb{E}\|f - \mathbf{U}_c\|_2^2 \\ &= R_0 + \mathbb{E}\|h - m_\mathcal{G}\|_2^2 + \lambda \mathbb{E}\|f - m_\mathcal{F}\|_2^2, \end{aligned} \tag{21}$$

where $R_0 = \mathbb{E}\|\mathbf{U}_c - m_\mathcal{G}\|_2^2 + \lambda \mathbb{E}\|\mathbf{U}_c - m_\mathcal{F}\|_2^2$ is constant. The two remaining terms vanish exactly at $f^\star = m_\mathcal{F}$ and $h^\star = m_\mathcal{G}$, so $r^\star = m_\mathcal{G} - m_\mathcal{F}$ almost surely. This proves Eq. (20); summing the channel-wise risks gives the corresponding full-field statement.

Risk reduction and scope. The tower property gives $\mathbb{E}[r^\star \mid \mathcal{F}] = 0$, and

$$\mathbb{E}\|\mathbf{U}_c - m_\mathcal{F}\|_2^2 - \mathbb{E}\|\mathbf{U}_c - m_\mathcal{G}\|_2^2 = \mathbb{E}\|r^\star\|_2^2 \geq 0. \tag{22}$$

At $\lambda = 0$, the optimal complete predictor is still $m_\mathcal{G}$, but its additive decomposition need not be unique. For a finite LIFT network, shared states and compressed interaction can prevent joint realizability. The residual also retains local processing capacity, so these population identities neither identify a pure interaction effect nor guarantee improvements in generation metrics.

E Controlled Architecture and Module Comparisons

The retrained comparisons fix the main-benchmark corpus, 128-wide backbone, 15,000-update budget, optimizer recipe, final EMA sampling, and evaluation protocol. The objective is shared except in the local-loss ablation. Tables report results averaged over three training seeds.

What the Separate controls hold fixed. Full uses four temporal blocks once for both local output and refinement. Separate 2+2 splits those blocks into two independently parameterized stacks, matching the total parameter count while shortening both paths. Separate 4+4 matches the path depth with an additional four-block stack. Input projections, channel/time embeddings, and boundary components are not all duplicated. The comparison therefore concerns temporal-stack sharing, not two fully independent generators. At ETTh1, counts are 1,033,644/1,033,644/1,431,980 for Full/Separate 2+2/Separate 4+4.

Direct-output control. This control retains all Full parameters and their initialization, as well as local supervision, but predicts the complete velocity using the interaction head alone. Full adds that head's prediction to the local field. Because the interaction head is zero-initialized, Direct output starts with a zero complete field; Full starts with the local prediction. The complete-field loss still reaches the temporal backbone through interaction, while the local head receives only its auxiliary-loss gradient in Direct output. This is an output-parameterization control, not a loss ablation or a claim of identical optimization dynamics.

Local-loss control. The w/o local loss variant retains Full's summed output and sets $\lambda_{\text{loc}} = 0$ throughout training. The complete-field gradient still updates both the local head and the backbone. This control has Context-FID and FDDS results, reported as means and 95% CI half-widths under the same three-training-seed protocol. Only three-decimal aggregates are available; equality at that precision does not establish equality before rounding.

Complete temporal-statistics comparison. Table 9 reports the six architecture variants with available lagged statistics on all six datasets. Each entry averages deterministic full-corpus estimates from three checkpoints, using the pooled-correlation definitions in Appendix B.5.

Alternative interaction modules. Both replacements operate on $[B, C, T, d]$ local states, with $d = 128$. Channel attention applies four-head scaled dot-product attention along C independently at each position, using PyTorch's optimized dispatcher. STAR-style uses a 128-dimensional shared core: it samples channel values per core coordinate during training and uses probability-weighted aggregation at inference, followed by channel/core fusion. Pooling probabilities are computed in FP32. This is a conditioned refiner based on STAR, not the complete SOFTS forecasting model.

Table 8: Architecture results on all six datasets over three training seeds. Mean ± 95% CI half-width; lower is better.

Variant	Sines	Stocks	ETTh1	MuJoCo	Energy	fMRI
Context-FID ↓						
Full LIFT	0.004±0.002	0.006±0.001	0.012±0.002	0.010±0.002	0.016±0.002	0.035±0.005
Local Only	0.004±0.002	4.252±0.555	0.851±0.150	0.943±0.176	2.612±0.336	0.174±0.048
Parallel (PWI)	0.004±0.002	0.281±0.022	0.414±0.052	0.809±0.154	1.407±0.283	0.061±0.010
Separate 2+2	0.004±0.002	0.008±0.001	0.015±0.002	0.013±0.002	0.018±0.002	0.045±0.005
Separate 4+4	0.004±0.002	0.007±0.002	0.014±0.002	0.012±0.002	0.017±0.002	0.047±0.006
Direct output	0.004±0.002	0.008±0.002	0.014±0.002	0.012±0.002	0.021±0.003	0.050±0.007
w/o local loss	0.004±0.001	0.017±0.002	0.037±0.002	0.010±0.002	0.016±0.002	0.043±0.005
Discriminative ↓						
Full LIFT	0.004±0.004	0.006±0.003	0.005±0.004	0.005±0.003	0.031±0.021	0.135±0.025
Local Only	0.004±0.004	0.403±0.079	0.365±0.082	0.371±0.070	0.499±0.001	0.460±0.019
Parallel (PWI)	0.004±0.003	0.065±0.094	0.138±0.028	0.366±0.063	0.353±0.015	0.195±0.018
Separate 2+2	0.004±0.004	0.007±0.007	0.008±0.007	0.006±0.005	0.053±0.010	0.159±0.053
Separate 4+4	0.006±0.005	0.008±0.010	0.004±0.003	0.006±0.002	0.045±0.023	0.169±0.031
Direct output	0.006±0.007	0.008±0.006	0.006±0.005	0.006±0.004	0.059±0.024	0.160±0.038
MDD ↓						
Full LIFT	0.586±0.014	0.346±0.027	0.172±0.004	0.310±0.008	0.232±0.006	0.183±0.002
Local Only	0.589±0.016	0.372±0.005	0.193±0.005	0.362±0.012	0.262±0.004	0.194±0.002
Parallel (PWI)	0.587±0.017	0.438±0.013	0.257±0.005	0.376±0.011	0.299±0.006	0.196±0.002
Separate 2+2	0.592±0.018	0.382±0.024	0.185±0.005	0.327±0.007	0.250±0.006	0.196±0.002
Separate 4+4	0.587±0.015	0.357±0.026	0.185±0.006	0.323±0.011	0.241±0.006	0.196±0.002
Direct output	0.589±0.017	0.399±0.033	0.187±0.005	0.332±0.007	0.245±0.005	0.197±0.001
FDDS ↓						
Full LIFT	0.001±0.000	0.029±0.006	0.005±0.002	0.012±0.001	0.047±0.002	0.004±0.000
Local Only	0.001±0.000	3.914±0.097	1.095±0.022	0.092±0.008	0.199±0.003	0.110±0.001
Parallel (PWI)	0.001±0.000	2.484±0.092	0.173±0.010	0.068±0.004	0.119±0.004	0.004±0.000
Separate 2+2	0.001±0.000	0.033±0.005	0.006±0.002	0.014±0.001	0.060±0.002	0.004±0.000
Separate 4+4	0.001±0.000	0.031±0.003	0.007±0.002	0.012±0.001	0.057±0.002	0.004±0.000
Direct output	0.001±0.000	0.036±0.005	0.007±0.002	0.016±0.002	0.057±0.002	0.005±0.000
w/o local loss	0.001±0.000	0.048±0.005	0.005±0.002	0.031±0.001	0.047±0.002	0.008±0.000

The local-loss ablation retains the summed field with $\lambda_{\text{loc}} = 0$; only Context-FID and FDDS are available for this variant. Its comparisons use reported three-decimal means; other rankings use unrounded means.

Table 9: Temporal and lagged cross-channel errors, averaged over three training seeds. Lower is better.

Variant	Sines	Stocks	ETTh1	MuJoCo	Energy	fMRI
ACF-MAE ↓						
Full LIFT	0.001	0.002	0.009	0.004	0.002	0.004
Local Only	0.001	0.003	0.007	0.003	0.004	0.004
Parallel (PWI)	0.001	0.005	0.025	0.006	0.003	0.005
Separate 2+2	0.001	0.003	0.012	0.005	0.005	0.005
Separate 4+4	0.001	0.004	0.013	0.005	0.005	0.004
Direct output	0.001	0.005	0.013	0.006	0.005	0.005
LagCCF-MAE ↓						
Full LIFT	0.008	0.004	0.007	0.009	0.011	0.008
Local Only	0.009	0.854	0.172	0.089	0.271	0.012
Parallel (PWI)	0.009	0.123	0.062	0.078	0.193	0.010
Separate 2+2	0.009	0.006	0.008	0.009	0.010	0.009
Separate 4+4	0.011	0.006	0.007	0.008	0.012	0.008
Direct output	0.008	0.008	0.008	0.009	0.012	0.009

Three-seed means of full-corpus estimates at lags 1–8.

The local initialization, two outer refinement stages and their FFNs/time conditioning, shared exchange weights, zero-initialized output head, gate, and output normalization are preserved. A separate identity-initialized FiLM exposes the same five state statistics to each replacement. The mean/carrier/update subpaths are replaced together, so internal projections differ. The 15,000-update budget, objective, local supervision, optimizer recipe, final EMA, 100-step Euler sampler, and evaluators remain unchanged.

Table 10: Complete interaction-module comparison over three training seeds. Mean ± 95% CI half-width; lower is better.

Variant	Sines	Stocks	ETTh1	MuJoCo	Energy	fMRI
Context-FID ↓						
Full LIFT	0.004 ± 0.002	0.006 ± 0.001	0.012 ± 0.002	0.010 ± 0.002	0.016 ± 0.002	0.035 ± 0.005
Mean Only	0.004 ± 0.002	0.006 ± 0.001	0.013 ± 0.003	0.013 ± 0.002	0.021 ± 0.003	0.050 ± 0.007
Channel attention	0.004 ± 0.002	0.007 ± 0.001	0.013 ± 0.003	0.012 ± 0.002	0.021 ± 0.002	0.047 ± 0.005
STAR-style	0.004 ± 0.002	0.006 ± 0.001	0.019 ± 0.005	0.032 ± 0.005	0.059 ± 0.010	0.068 ± 0.012
Discriminative ↓						
Full LIFT	0.004 ± 0.004	0.006 ± 0.003	0.005 ± 0.004	0.005 ± 0.003	0.031 ± 0.021	0.135 ± 0.025
Mean Only	0.005 ± 0.005	0.008 ± 0.006	0.006 ± 0.005	0.007 ± 0.004	0.067 ± 0.037	0.217 ± 0.034
Channel attention	0.005 ± 0.005	0.007 ± 0.005	0.008 ± 0.005	0.006 ± 0.002	0.044 ± 0.052	0.159 ± 0.026
STAR-style	0.006 ± 0.006	0.012 ± 0.011	0.009 ± 0.004	0.025 ± 0.043	0.159 ± 0.040	0.263 ± 0.035
MDD ↓						
Full LIFT	0.586 ± 0.014	0.346 ± 0.027	0.172 ± 0.004	0.310 ± 0.008	0.232 ± 0.006	0.183 ± 0.002
Mean Only	0.589 ± 0.016	0.379 ± 0.030	0.186 ± 0.005	0.326 ± 0.007	0.251 ± 0.005	0.196 ± 0.002
Channel attention	0.591 ± 0.018	0.403 ± 0.032	0.190 ± 0.006	0.324 ± 0.009	0.256 ± 0.005	0.195 ± 0.002
STAR-style	0.587 ± 0.019	0.396 ± 0.030	0.195 ± 0.006	0.363 ± 0.014	0.257 ± 0.005	0.196 ± 0.002
FDDS ↓						
Full LIFT	0.001 ± 0.000	0.029 ± 0.006	0.005 ± 0.002	0.012 ± 0.001	0.047 ± 0.002	0.004 ± 0.000
Mean Only	0.001 ± 0.000	0.038 ± 0.007	0.007 ± 0.002	0.015 ± 0.002	0.061 ± 0.002	0.009 ± 0.000
Channel attention	0.001 ± 0.000	0.040 ± 0.007	0.008 ± 0.002	0.009 ± 0.001	0.060 ± 0.002	0.006 ± 0.000
STAR-style	0.001 ± 0.000	0.063 ± 0.007	0.014 ± 0.003	0.021 ± 0.002	0.080 ± 0.003	0.016 ± 0.000

Attention and STAR-style use closely matched parameter budgets; Table 11 reports ETTh1 counts.

Table 11: Parameter counts for the interaction-module quality comparison on ETTh1.

Variant	Complete model	Interaction refiner	Total difference vs. Full
Full LIFT	1,033,644	580,811	—
Channel attention	1,033,346	580,513	-0.029%
STAR-style	1,032,802	579,969	-0.081%

Both replacements match Full to within 0.1% in total parameters. Complete-model counts include the local encoder and prediction/refinement heads; the remaining 452,833 parameters are common to all three variants. Parameter matching does not imply matched compute.

Table 12: Temporal and lagged cross-channel errors of interaction replacements, averaged over three training seeds. Lower is better.

Variant	Sines	Stocks	ETTh1	MuJoCo	Energy	fMRI
ACF-MAE ↓						
Full LIFT	0.001	0.002	0.009	0.004	0.002	0.004
Mean Only	0.001	0.004	0.012	0.005	0.004	0.004
Channel attention	0.001	0.003	0.011	0.005	0.004	0.004
STAR-style	0.001	0.003	0.019	0.005	0.005	0.004
LagCCF-MAE ↓						
Full LIFT	0.008	0.004	0.007	0.009	0.011	0.008
Mean Only	0.009	0.007	0.008	0.009	0.013	0.009
Channel attention	0.009	0.006	0.009	0.009	0.014	0.009
STAR-style	0.010	0.006	0.009	0.017	0.021	0.010

Lags 1–8 use the same definitions as the architecture controls.

The attention and STAR-style controls use closely matched parameter budgets. On ETTh1, their complete-model counts differ from Full by only 0.029% and 0.081%, respectively (Table 11).

E.1 CARRIER REMOVAL WITH AND WITHOUT RETRAINING

The trained Full field can rely on carriers even when a model trained without them learns to compensate. We separate these effects by comparing fixed-model carrier removal with retrained Mean Only. Generation scores for Full and retrained Mean Only appear in Table 10.

Fixed-model intervention. We also disable the carrier contribution at both interaction stages of each trained Full model, without changing its weights, and average over three training seeds. Each dataset uses the same 1,024 training-corpus windows, Gaussian noise, flow times, and physical-batch Hungarian pairings across cases (diagnostic seed 202, batch 64, FP32, final EMA). The local output is unchanged under this intervention. Table 13 reports full-field FM MSE and its percentage change relative to Full, $100(E/E_{\text{Full}} - 1)$. These are training-corpus diagnostics, not held-out or generation errors.

Table 13: Full-field FM error under carrier removal and retraining. Means over three training seeds. Positive changes indicate higher MSE than Full.

Dataset	Full MSE	Carriers off ($\Delta\%$)	Mean Only ($\Delta\%$)
Sines	0.025	+0.18	+0.03
Stocks	0.028	+62.39	+0.78
ETTh1	0.106	+173.36	+1.78
MuJoCo	0.042	+148.21	+2.77
Energy	0.099	+285.44	+3.95
fMRI	0.353	+7.54	+1.06

Removing carriers increases the trained Full field's error most on Energy, ETTh1, and MuJoCo. Retrained Mean Only has much smaller increases, consistent with compensation by the remaining network. The local output is unchanged by fixed-field removal, but later mean-path inputs can change; the effect therefore includes downstream activation changes. These training-corpus point estimates show reliance on the learned pathway without establishing that carriers are indispensable.

F CROSS-CHANNEL CORRESPONDENCE DIAGNOSTICS

RQ3 replaces cross-channel context while preserving the target trajectory and leaving all model pathways active.

Protocol. RQ3 uses the same Full final-EMA checkpoints as Table 1, one per training seed. Evaluation uses FP32, with a fixed subset of 1,024 training-corpus windows per dataset. Each of five diagnostic repeats (seeds 202, 1202, 2202, 3202, and 4202) draws new Gaussian noise and physical-batch groupings, then applies the existing exact Hungarian OT within batches of 64. The resulting paths and original velocity targets are fixed before intervention; OT is not recomputed after mismatch. We evaluate $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, keeping the target trajectory, time, label, and model weights unchanged and scoring only the target channel.

Donor assignment. For receiver i and target channel c, all other channels come jointly from donor $\pi(i)$ at the same t. Both conditions use a one-to-one assignment over the 1,024-window bank, preserving the empirical distribution of the donor blocks. Random minimizes independent uniform random edge costs; its assignment is reused across channels and times within a repeat. Matched instead minimizes $\sum_i T^{-1}\|\mathbf{x}_{t,i}^{(c)} - \mathbf{x}_{t,\pi(i)}^{(c)}\|_2^2$ separately for each (c,t). Matching uses only visible target trajectories, not clean endpoints, velocity labels, other channels, or model predictions. Both conditions exclude self-donors and donors from the same OT batch. Stocks, ETTh1, Energy, and fMRI additionally exclude overlapping raw windows (start-index separation less than T). Numerical checks confirm unchanged target-local outputs and agreement with native full-model evaluation.

Paired error. For each repeat and condition q, we compute

$$\Delta_q = \frac{1}{5NC} \sum_{t,i,c} \left[\frac{\|v_{\theta,c}(\widetilde{\mathbf{x}}^{q,c}_{t,i}, t) - \mathbf{u}^*_{i,c}\|_2^2}{T} - \frac{\|v_{\theta,c}(\mathbf{x}_{t,i}, t) - \mathbf{u}^*_{i,c}\|_2^2}{T} \right], \tag{23}$$

where $N = 1{,}024$ and $\widetilde{\mathbf{x}}^{q,c}$ preserves channel c while replacing its context. For each reported score, we first average the three seed-specific values within diagnostic repeat j to obtain $\bar{z}_j = \frac{1}{3}\sum_{s=1}^{3} z_{s,j}$. We then report the mean of the five $\bar{z}_j$ and its Student-t 95% CI (four degrees of freedom; Table 14). The interval describes variation across diagnostic repeats after seed averaging, not uncertainty across training seeds. Relative increases are computed per repeat as $100\Delta_q/\mathrm{MSE}_{\mathrm{original}}$ and then averaged. These are training-corpus velocity diagnostics, not held-out or generation scores; the time-grid average is not an exact continuous-time risk integral.

Table 14: Original target-channel MSE (mean only) and paired increases under mismatch (mean $\pm$ 95% CI half-width). Each diagnostic repeat averages three training seeds; intervals use five repeat-level averages.

Dataset	Original MSE	Random ΔMSE	Matched ΔMSE
Sines	0.023	$(2.19 \pm 0.37) \times 10^{-6}$	$(2.62 \pm 1.53) \times 10^{-6}$
Stocks	0.017	0.820 ± 0.017	0.219 ± 0.002
ETTh1	0.082	0.377 ± 0.002	0.178 ± 0.001
MuJoCo	0.039	0.139 ± 0.002	0.059 ± 0.001
Energy	0.091	0.392 ± 0.004	0.198 ± 0.000
fMRI	0.339	0.048 ± 0.000	0.043 ± 0.000

Matching quality. Matched donors have target trajectories closer to the receiver's while still changing its context. Their mean squared donor-to-receiver target-trajectory distance is 58–65% lower than Random's. Mean squared context perturbations retain 84.3% of Random's magnitude on Stocks and 97.9–100.3% on the other datasets. Matching is approximate: it neither samples the exact conditional distribution nor guarantees in-distribution combinations. Mean pooling, carriers, and state-statistic gates can all respond to the intervention, so its effect measures their combined response.

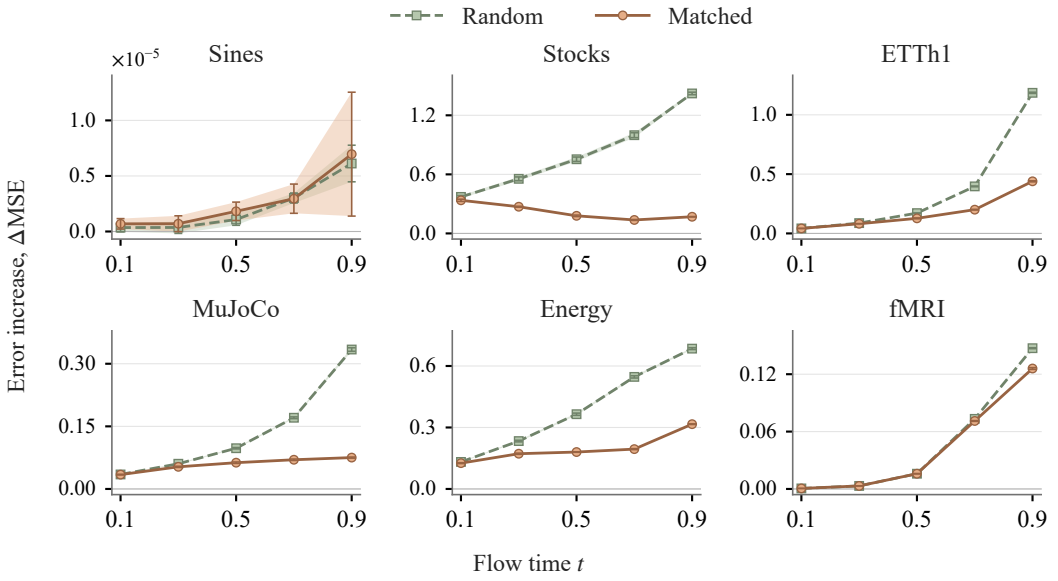

Figure 4: Mismatch-induced error increase across flow time. Shading and error bars show 95% CIs over diagnostic repeats; vertical scales vary by dataset.

Time and channel dependence. Matched increases remain positive at every sampled time on the five non-Sines datasets (Figure 4). The fMRI effect is concentrated later in the path: ΔMSE increases from 5.5×10^{-4} at $t = 0.1$ to 0.126 at $t = 0.9$. Stocks instead decreases before a final uptick; at $t = 0.9$, its Matched context perturbation is only 14.5% of Random's, so there is no universal time trend. All 105 non-Sines target channels have positive mean Matched increases after averaging times and repeats, a descriptive check rather than a multiple-testing significance claim.

Energy redundancy check. The canonical Energy columns `rv1` and `rv2` are identical and contribute 42.5% of the sum of channel-wise Matched increases. Averaging the saved errors over the remaining 26 target channels still gives $\Delta\text{MSE} = 0.122 \pm 0.001$ (95% CI). This is a descriptive slice of the existing scores: all 28 input channels remain present, with no retraining or new intervention. Table 14 retains the canonical 28-channel result.

G Long-Sequence Generation

G.1 Generation quality at each trained length

We use the same ETTh1 and Energy channels and normalization as the main benchmark, with stride-one windows of $T = 64, 128, 256$. The corresponding corpus sizes are $17{,}357/17{,}293/17{,}165$ for ETTh1 and $19{,}672/19{,}608/19{,}480$ for Energy. Each model is trained separately at each length with seeds 1, 2, and 3 and generates the corresponding corpus size per run. Table 15 reports three-seed means and 95% confidence intervals.

All five methods retain their training budgets and native samplers. LIFT and PWI keep the shared 128-wide, four-block configuration. These comparisons use length-specific training without matching compute.

Table 15: Long-sequence quality over three training seeds. Mean ± 95% CI half-width; lower is better. Bold marks the lowest mean.

Method	ETTh1			Energy		
	$T = 64$	$T = 128$	$T = 256$	$T = 64$	$T = 128$	$T = 256$
Context-FID ↓						
LIFT	**0.038**±.004	**0.078**±.005	**0.144**±.006	**0.032**±.006	**0.066**±.012	**0.106**±.023
PWI	0.954±.042	1.576±.201	2.314±.151	1.865±.137	2.388±.298	3.730±.570
Diffusion-TS	0.271±.030	0.374±.048	2.559±.110	0.111±.010	0.092±.010	0.130±.005
FlowTS	0.326±.068	0.482±.071	0.638±.198	0.394±.070	0.586±.065	1.356±.133
PaD-TS	1.906±.237	3.317±.460	4.765±.625	0.173±.014	0.126±.009	0.119±.011
Discriminative ↓						
LIFT	**0.009**±.007	**0.022**±.017	**0.025**±.026	**0.065**±.015	**0.077**±.052	**0.175**±.103
PWI	0.187±.045	0.227±.054	0.277±.082	0.380±.061	0.381±.145	0.348±.171
Diffusion-TS	0.088±.007	0.087±.008	0.214±.013	0.116±.024	0.205±.029	0.267±.113
FlowTS	0.109±.021	0.134±.053	0.167±.082	0.345±.101	0.371±.090	0.373±.075
PaD-TS	0.148±.027	0.189±.026	0.275±.035	0.216±.012	0.191±.074	0.257±.159
MDD ↓						
LIFT	**0.178**±.002	**0.187**±.004	**0.193**±.001	**0.240**±.003	**0.241**±.001	**0.242**±.002
PWI	0.258±.009	0.274±.005	0.286±.004	0.278±.005	0.263±.003	0.264±.003
Diffusion-TS	0.208±.008	0.215±.009	0.307±.009	0.250±.004	0.259±.005	0.259±.004
FlowTS	0.258±.004	0.288±.006	0.295±.002	0.410±.003	0.420±.005	0.466±.009
PaD-TS	0.310±.005	0.341±.004	0.360±.004	0.311±.005	0.295±.004	0.281±.005
FDDS ↓						
LIFT	**0.007**±.002	**0.010**±.001	**0.019**±.002	**0.012**±.001	**0.014**±.001	0.022±.002
PWI	0.259±.009	0.330±.015	0.417±.007	0.209±.005	0.357±.004	0.466±.007
Diffusion-TS	0.090±.005	0.073±.010	0.179±.005	0.048±.003	0.031±.001	0.035±.000
FlowTS	0.044±.006	0.054±.005	0.045±.004	0.316±.004	0.133±.002	0.076±.003
PaD-TS	0.242±.010	0.329±.015	0.341±.011	0.037±.002	0.016±.001	**0.011**±.001

Rankings use means before display rounding.

G.2 Model size and generation time on ETTh1

On a remote server with an exclusively used NVIDIA RTX 5090 GPU (PyTorch 2.11.0, CUDA 12.8), we generate 1,024 FP32 sequences with batch size 64 and TF32 disabled. Table 16

reports median [Q1, Q3] time across five measurements after three warm-up batches, using one checkpoint per setting. Timing includes noise, the complete sampler, and CPU conversion; it excludes loading, disk writes, and evaluation. Counts use one inference model from the frozen configuration, excluding duplicate EMA copies and buffers.

The native evaluation counts are 100 for LIFT/PWI/FlowTS, 500 for Diffusion-TS, and 250 for PaD-TS. LIFT is 3.54–$4.44\times$ faster than Diffusion-TS and 1.95–$2.30\times$ faster than PaD-TS under these recipes; FlowTS has similar times. These are complete-sampler comparisons with unequal evaluation counts, not measurements of isolated interaction cost. Repeated timings are distinct from training-seed averages.

Table 16: ETTh1 model size (M parameters) and generation time (seconds) for 1,024 samples. Times report median [Q1, Q3] over five measurements.

Method	$T = 64$		$T = 128$		$T = 256$	
	Params	Time	Params	Time	Params	Time
LIFT	1.034	30.19 [29.86, 30.41]	1.034	24.70 [24.54, 24.71]	1.034	29.62 [29.58, 29.73]
PWI	1.034	30.87 [30.80, 31.52]	1.034	30.89 [30.69, 30.93]	1.034	30.06 [29.92, 30.18]
Diffusion-TS	0.374	106.80 [106.62, 106.81]	0.432	109.72 [109.67, 109.90]	0.648	115.80 [115.77, 115.90]
FlowTS	0.440	25.03 [25.01, 25.10]	0.491	25.62 [25.61, 25.63]	0.690	26.70 [26.68, 26.76]
PaD-TS	2.310	58.85 [57.97, 58.99]	2.343	56.71 [56.20, 56.76]	2.409	59.11 [58.93, 59.41]

Native samplers: 100 evaluations for LIFT, PWI, and FlowTS; 500 for Diffusion-TS; 250 for PaD-TS.

H Additional Qualitative Results

The three views compare sequence projections and pooled value distributions for all nine methods in Table 1 ($T = 24$), including CSDI. Figures 5–7 show PCA, t-SNE, and kernel density estimates (KDE), respectively. We use the training-seed-1 models and 3,000 real and 3,000 generated windows selected with plotting seed 0. Each window is averaged across channels before plotting, with no additional normalization or clipping.

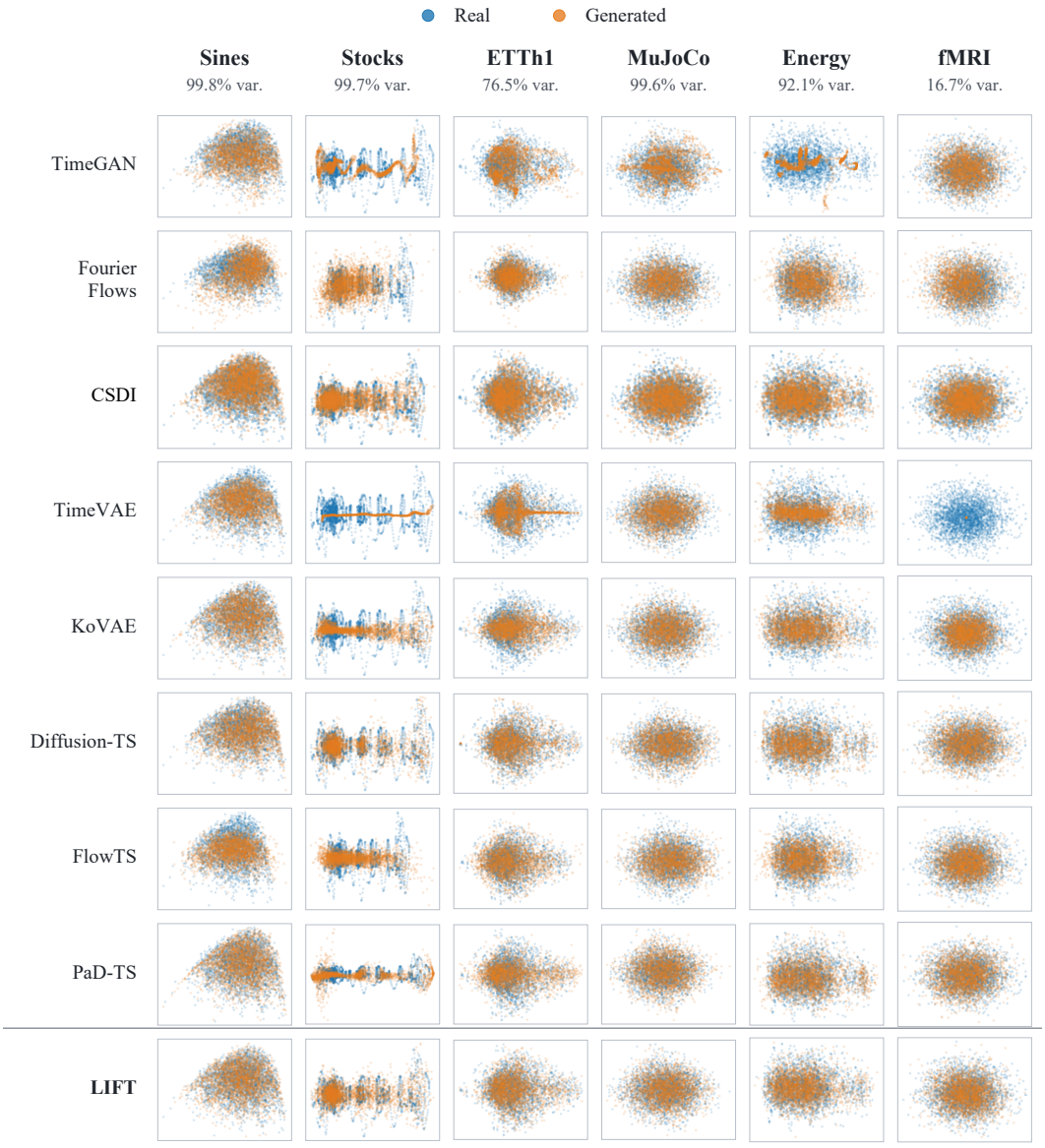

Figure 5: PCA of channel-mean sequences, fitted on real windows and applied to both sets (horizontal: PC1; vertical: PC2). Headers give combined explained variance, only 16.7% for fMRI. Axis ranges vary by panel.

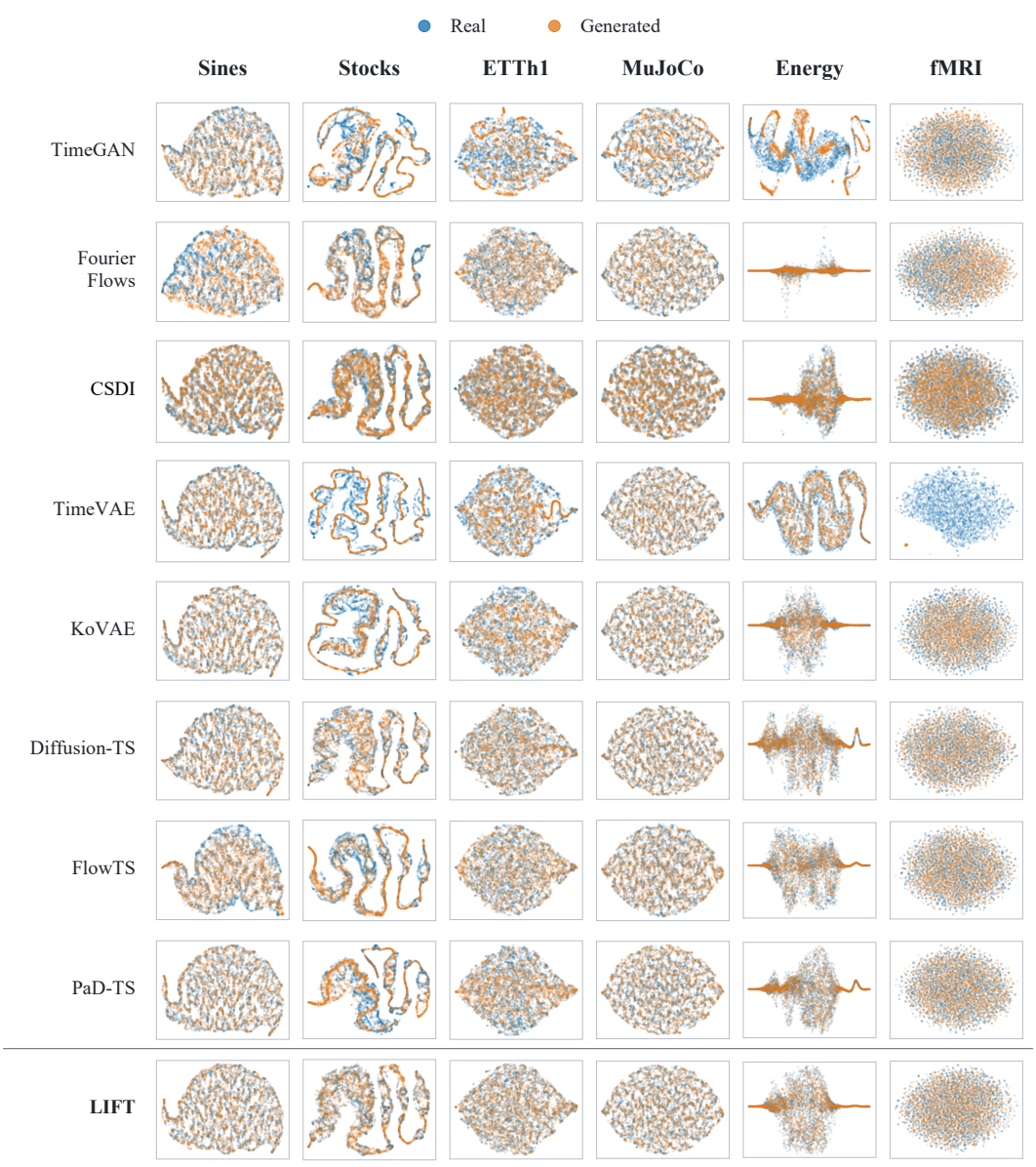

Figure 6: Joint t-SNE embeddings of real and generated channel-mean sequences within each panel (perplexity 40, PCA initialization, 300 iterations, seed 0). Coordinates and distances are not comparable across panels.



Figure 7: Kernel density estimates of channel-mean values pooled over windows and time. Real values use solid blue curves and generated values use dashed orange curves. Gaussian kernels use Scott's bandwidth separately for each set; axis ranges vary by panel.

The views reveal different discrepancies. On Stocks, TimeVAE's pooled density broadly follows the real peaks, while its PCA projection has a narrower vertical spread. On fMRI, its generated values concentrate into a sharp density peak, consistent with the compact projected samples. For Energy, LIFT and several baselines show a concentrated region in t-SNE even when their pooled densities largely overlap the real curve. Channel averaging discards channel-specific structure, so these views do not establish multivariate fidelity; FDDS, LagCCF-MAE, and the correspondence diagnostic provide complementary cross-channel evidence.