STREAM: Dynamic Channel Graphs with Autoregressive State Propagation for Multivariate Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Real-world multivariate series involve multiple dependencies: each channel has its own dynamics, channels interact through spatial or physical structure, and these interactions can change across regimes. Yet standard benchmarks fix the channel set and lookback, allowing models that ignore inter-channel dependencies to remain competitive and leaving their value insufficiently tested. In deployment, sensors may fail, become noisy, or be added, while longer histories accumulate. We therefore evaluate dependency modeling through a controlled robustness protocol that expands the channel set along a sensor graph while scoring a fixed core set, and independently extends the lookback. We then present STREAM, which models channel dependency within each temporal patch using a sparse graph re-estimated from the patch, and models temporal dependency with a linear-time selective state-space recurrence. Its bounded neighborhoods limit the effect of added channels, while the recurrence supports longer contexts with linear cost and patch-by-patch forecasting beyond the trained horizon.

1 Introduction

Real-world multivariate time series are governed by multiple dependencies at once. Each channel has its own temporal dynamics, channels interact through a spatial or physical structure, and these interactions can change across regimes. Yet standard long-term forecasting benchmarks largely abstract away this structure: the channel set is fixed, the lookback is short (typically L=96), and the forecasting horizons are fixed in advance. Under this setting, channel-independent models remain highly competitive, both as direct forecasters (Zeng et al., 2023; Nie et al., 2023) and as backbones of foundation models (Das et al., 2024; Liu et al., 2024a). Recent analysis further suggests that much of the measured accuracy can be explained by within-channel regularities alone (Chen et al., 2025). Thus, these benchmarks provide limited evidence about when inter-channel dependencies contribute to forecasting performance, particularly when the input configuration or available temporal context changes.

In deployment, the input configuration may vary over time. Channels can be added, corrupted, or missing: newly introduced sensors may provide useful information, while noisy or failed sensors may provide little or no useful signal. Similarly, longer temporal histories may become available as observations accumulate, although existing models can saturate or degrade with increasing lookback length (Zhang et al., 2025). These variations motivate two requirements for dependency modeling: channel dependencies should remain selective and robust to irrelevant or missing inputs, while temporal dependencies should remain effective as the available context grows.

We turn these requirements into a controlled robustness study by expanding the channel set along a physical sensor graph with informative, noisy, or failing channels while keeping the same core channels, and independently extending the lookback from a shared anchor. This separates robustness to *more channels* from robustness to *more history* under a fixed prediction target (Figure 1). For each architectural family, we examine the structural fail-

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

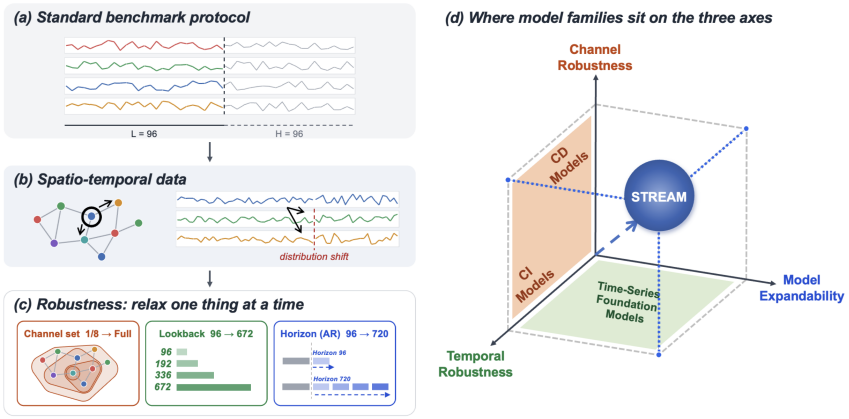


Figure 1: From benchmark protocol to robustness. (a) Standard benchmarks fix the channel set, lookback, and horizon. (b) Spatio-temporal data couple channels with a lag, and both the coupling and the marginal distributions change after a regime shift. (c) We relax one factor at a time: the channel set, the lookback, and the horizon under autoregressive rollout. (d) Existing families (channel independency (CI) and channel dependency (CD)) cover at most two of the resulting axes; STREAM targets all three.

ure modes induced by its dependency modeling, treating robustness—not performance on a fixed split—as the criterion for evaluating dependency modeling. This perspective distinguishes two properties: channel dependencies should be *selective*, such that irrelevant or missing channels do not unnecessarily affect other predictions, whereas temporal dependencies should be *persistent*, allowing useful information to be retained over longer contexts without jointly processing all channels and time steps. Dense channel mixing can couple predictions across channels, graph-based approaches may expand their neighborhoods as the channel set grows, and attention-based temporal mixing can incur increasing computational cost with longer contexts.

We propose STREAM (**ST**ate-space **RE**lational **A**utoregressive **M**odel) by explicitly separating these two forms of dependency. Within each temporal patch, STREAM constructs a sparse graph from the current patch and uses it for selective cross-channel mixing. The graph is re-estimated at each patch, allowing the dependency structure to adapt to changing regimes, while its bounded neighborhood limits the expansion of the receptive field as channels are added. Across patches, STREAM applies a selective state-space recurrence (Gu & Dao, 2024) independently to each channel. This yields linear temporal computation with respect to context length while keeping the temporal recurrence independent of channel ordering and count. The same causal factorization also enables patch-by-patch forecasting beyond the training horizon, which we evaluate as an additional dimension of robustness.

Our contributions are summarized as follows:

- **STREAM.** To our knowledge, the first forecaster that separates inter-channel and temporal dependency at the level of each patch: channels interact only through a top-k graph rebuilt from every patch, and time only through a per-channel recurrence, so informative neighbors are used without dilution by irrelevant or failed channels.
- **Linear-time and extensible by design.** Cost grows linearly with the lookback, memory linearly with the number of channels, and the same trained model rolls out patch by patch beyond its trained horizon.
- **A robustness protocol and its evaluation.** We vary the channel set (informative, noisy, and failing channels, scored on a fixed core) and the lookback on nine spatio-temporal datasets with physical channel graphs against eight baselines, and extend the study to autoregressive rollout.

2 Related Work

Channel strategies. Channel-independent forecasters (Zeng et al., 2023; Nie et al., 2023) remain strong on standard benchmarks, where much of the accuracy does not require inter-channel modeling (Chen et al., 2025). Channel-mixing models tokenize whole variates (Liu et al., 2024b), mix across scales (Wang et al., 2024), or scan variate tokens with a state-space model (Wang et al., 2025b), whose output then depends on the variate order. They are compared with a fixed channel set and a short lookback, although whether forecasters benefit from longer contexts is itself open (Zhang et al., 2025). We vary both and separate *using* informative channels from *ignoring* noisy ones.

Graph-based and state-space spatio-temporal models. Traffic and sensor forecasters combine graph propagation with temporal models on a predefined or globally learned adjacency (Li et al., 2018; Wu et al., 2019). TimeFilter (Hu et al., 2025) builds a kNN graph over channel–patch tokens and filters it with a mixture of experts; its neighborhood grows with the number of tokens and its cost is $O((C \cdot P)^2)$ at a fixed patch size. Two long-term forecasters combine a learned graph with a state-space scan. MambaTS (Cai et al., 2024) learns a directed graph over variables during training and derives from it a single scan order, and GG-SSM (Zubić & Scaramuzza, 2025) propagates the state along a spanning tree generated from feature similarity; DyG-Mamba (Li et al., 2025) applies continuous state-space modeling to dynamic graphs. In these models one graph serves the whole window, whereas STREAM builds a graph of fixed budget k for every patch and carries time by the recurrence (Appendix C).

Autoregressive forecasting. Foundation models forecast step by step or patch by patch, either decoder-only (Das et al., 2024; Liu et al., 2024c) or encoder–decoder (Ansari et al., 2024), and AutoTimes (Liu et al., 2024a) repurposes a frozen language model as a channel-independent autoregressive forecaster for arbitrary horizons. Compounding error is the central obstacle, addressed by scheduled sampling (Bengio et al., 2015) and autoregressive objectives (Li et al., 2026). We measure how much error a rollout accumulates relative to direct training on multivariate data.

3 Robustness Protocol

Let $X \in \mathbb{R}^{L \times C}$ be a lookback window of length L over C channels, where the channels are the nodes V of a sensor or spatial graph. The task is to predict the next H steps $Y \in \mathbb{R}^{H \times C}$. Every experiment starts from one **anchor**: $L = 96$, $H = 96$, all channels, clean data, and changes exactly one factor.

3.1 Axis 1: the channel set

Nested channel subsets. We order channels by breadth-first expansion over the channel graph from a center node (Appendix A, Fig. 5). This yields nested, spatially contiguous subsets $V_{1/8} \subset V_{1/4} \subset V_{1/2} \subset V$ whose sizes are $\lfloor C/8 \rfloor, \lfloor C/4 \rfloor, \lfloor C/2 \rfloor, C$. A separate model is trained and tested on each fraction f. The error is always computed on the same *core* $V_{1/8}$, so every fraction is scored on an identical target set, and the channels added beyond the core act only as context. Because the expansion follows the graph, each fraction is a coherent sub-network (a climate patch, a metropolitan sensor cluster) rather than a random sample.

Three scenarios. The scenarios differ only in what the channels beyond the core contain.

- **A (informative).** Added channels are clean neighbors, as when an operator installs detectors on adjacent road segments or extends a climate grid. A model that uses dependency should improve as f grows.
- **B (noisy).** A fixed 20% of the non-core channels are overwritten in the input and target windows of every split with i.i.d. $\mathcal{N}(0, 1)$ samples, redrawn for every window,

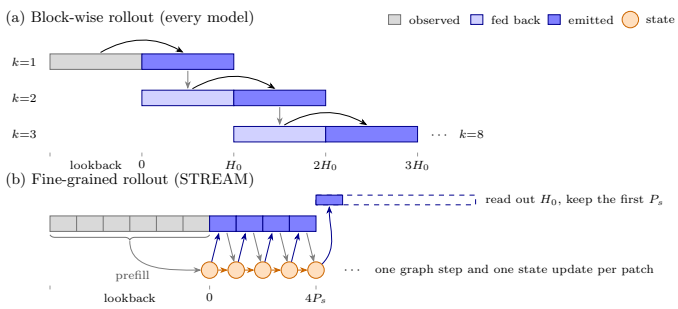


Figure 2: Extending a forecast beyond the trained horizon H_0. (a) Block-wise rollout (every model): forecast H_0 steps, slide the window, repeat; from the second block the input is the model's own output. (b) Fine-grained rollout (STREAM): encode the lookback once, then emit one patch at a time and feed it back through one graph step and one state update.

as with newly added sensors that are miscalibrated or report corrupted readings. The core is never altered. A robust model should stay flat.

- **C (test-time failure).** The model is trained on the clean V_f; at test time a fixed set E of 20% of all channels, core included, fails, as when deployed sensors go offline after training. A failed channel keeps its slot but reports $\mathcal{N}(0,1)$ noise, so no model needs a missing-input mechanism, and the core is scored without E. A larger monitored area should compensate for the failures.

Summary statistics. Each A/B curve is summarized by its normalized slope,

$$s_C = \mathrm{OLS}_f\Big(\frac{\mathrm{MSE}(f)}{\mathrm{MSE}(1/8)},\ \log_2 8f\Big), \tag{1}$$

the least-squares slope of the relative error against the number of doublings of the channel count. The desired sign is $s_C \le 0$ for A and $s_C \approx 0$ for B. A flat curve at a poor level is not robustness, so we also report the full-channel error itself. For C we report the spatial-scaling slope $s_C^{\mathrm{C}} = \mathrm{OLS}_f\big(\ln \mathrm{MSE}_{\mathrm{C}}(f), \log_2 8f\big)$: how fast the error under failure falls, in log units, each time the monitored area doubles.

3.2 Axis 2: the lookback length

With all channels and clean data, we train and test at $L \in \{96, 192, 384, 672\}$ with $H = 96$; the $L = 96$ point is the full-channel point of Axis 1. We report

$$s_L = \mathrm{OLS}_L\Big(\frac{\mathrm{MSE}(L)}{\mathrm{MSE}(96)},\ \log_2 \frac{L}{96}\Big), \qquad \rho_{672} = \frac{\mathrm{MSE}(672)}{\mathrm{MSE}(96)}, \tag{2}$$

where $s_L \le 0$ and $\rho_{672} \le 1$ mean that longer context helps. As on Axis 1, the endpoint MSE(672) must itself be competitive, and a model that cannot run at $L = 672$ is reported as OOM.

3.3 Axis 3 (extension): the horizon under autoregressive rollout

Every model is trained once at the anchor ($L = H_0 = 96$). It is evaluated (i) *directly* at H_0 and (ii) by *block-wise rollout*: the model emits H_0 steps, appends them to its input, slides the window by H_0, and repeats until $H = 720$ (Figure 2a). This applies the same procedure to every model, including direct forecasters that were never trained to consume their own output. STREAM additionally supports *fine-grained* rollout (Figure 2b): after one pass over the lookback it emits one patch of P_s steps at a time and feeds it back into its cached state, so the graph and the recurrent state are refreshed every P_s steps rather than every H_0 steps, from the same trained checkpoint.

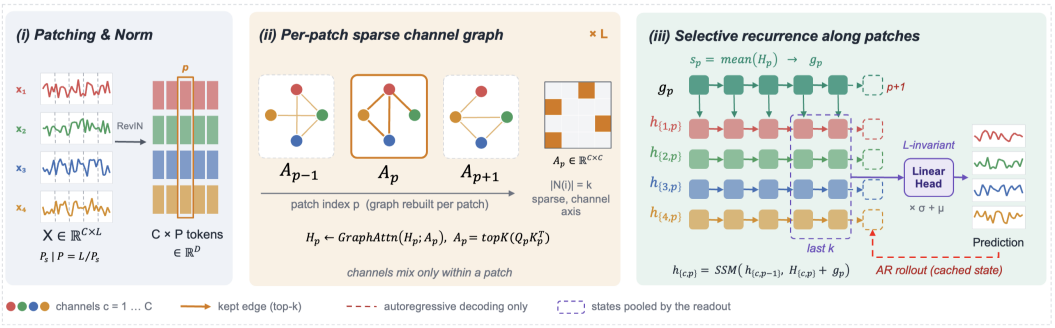


Figure 3: STREAM (C=4 channels shown). (i) Each channel is instance-normalized and cut into P patches. (ii) Within every patch, each channel keeps a sparse neighborhood of k channels (Eq. 3); the graph is rebuilt per patch and this is the only place where channels mix. (iii) Time is carried across patches by a causal recurrence per channel, conditioned on the graph-summary stream g_p; the readout maps the last κ states to the horizon (Eq. 4). Dashed: autoregressive rollout with cached state.

3.4 Expected failure modes

For each family we derive the behavior its structure implies on each axis: no path from a neighbor to a target for channel-independent models, a path from every channel for dense mixing, a neighborhood that grows with C for similarity graphs, and a single graph or scan order per window for graph-guided state-space models.

4 STREAM

STREAM separates the two axes of a multivariate series. Channel dependency is resolved *within* a temporal patch by a sparse graph; temporal dependency is carried *across* patches by a selective state-space recurrence run independently for each channel. Because patches interact only through this causal recurrence, the model also factorizes by patch, which gives patch-wise autoregressive decoding without a separate decoder: after one pass over the lookback, each further patch costs one graph step and one state update from a cache. Its cost grows linearly with the lookback (Appendix C). Figure 3 gives the overview.

4.1 Sparse channel graph within a patch

Each channel is normalized by its own lookback statistics (Kim et al., 2022), cut into $P = L/P_s$ non-overlapping patches, and embedded linearly with a learned channel embedding, $h_{p,c} = W_e\, x_{p,c} + e_c$; no positional embedding is used, since the recurrence supplies the order. For each patch p the C tokens are projected to multi-head queries, keys, and values, and each channel keeps its k highest-scoring neighbors,

$$e_{ij}^{(p)} = \frac{\langle q_{p,i},\, k_{p,j}\rangle}{\sqrt{d}}, \qquad \mathcal{N}_p(i) = \mathrm{top}_k\big(e_{i,\cdot}^{(p)}\big), \qquad a_{ij}^{(p)} = \mathrm{softmax}_{j\in\mathcal{N}_p(i)}\big(e_{ij}^{(p)}\big), \tag{3}$$

and aggregates $\sum_{j\in\mathcal{N}_p(i)} a_{ij}^{(p)}\, v_{p,j}$, followed by a residual connection, layer normalization, and a feed-forward block; the layer is stacked twice and no predefined adjacency is used. Because k does not grow with C, adding channels changes *which* channels are attended, not how many, and an uninformative channel is out-competed for the k slots. The graph is rebuilt for every patch, and only the top-k pairs are kept, so it needs $O(P \cdot C \cdot k)$ memory.

4.2 Causal streams across patches

Time is carried along the patch axis by a selective state-space model (Gu & Dao, 2024), in two streams. In the *node stream*, each channel's tokens $(h_{1,c}, \ldots, h_{P,c})$ pass through a causal

recurrence with parameters shared across channels and no mixing between them, so the sequence length is P rather than $C \cdot P$ and nothing depends on the order in which channels are listed (Wang et al., 2025b); the state $z_{p,c}$ summarizes $[0, pP_s)$ of channel c together with the messages it received. In the *graph stream*, the patch summary $s_p = W_s \frac{1}{C} \sum_c h_{p,c}$ and its change enter a one-layer recurrence, $g_p^{\text{in}} = s_p + W_\Delta(s_p - s_{p-1})$, $g_p = \text{LN}\big(g_p^{\text{in}} + \text{SSM}_g(g^{\text{in}})_p\big)$, and g_p is added to every channel before the node stream, $h_{p,c} \leftarrow h_{p,c} + g_p$. It is the path through which each channel sees how the dependency structure of the patch has been evolving.

4.3 Readout and training objective

At a read position p, the last κ node states of channel c are layer-normalized, concatenated, and mapped to the horizon,

$$\hat{Y}_{:,c}^{(p)} = W_o\big[\text{LN}(z_{p-\kappa+1,c}); \ldots; \text{LN}(z_{p,c})\big], \qquad W_o \in \mathbb{R}^{H \times \kappa D}, \tag{4}$$

with W_o shared across channels, missing slots zero-filled, and the output de-normalized. The readout length κ is a hyperparameter; we set it to half of the lookback, $\kappa = \text{round}(P/2)$ (Table 7). Unlike flatten-and-project heads (Nie et al., 2023), the head reads only the most recent states, and because every state is causal it can be read at any position; the main forecast is $\hat{Y} = \hat{Y}^{(P)}$.

The objective has two terms, both measured with the Huber penalty

$$\ell_\delta(e) = \begin{cases} e^2, & |e| \le \delta, \\ 2\delta|e| - \delta^2, & |e| > \delta, \end{cases} \tag{5}$$

which equals the squared error for small residuals and grows linearly for large ones, so that a few extreme errors (e.g., sensor spikes or missing-value artifacts) do not dominate the gradient. We use $\delta = 0.5$ on the standardized scale of the data; validation and test are scored with MSE. The main forecast is fitted to the target,

$$\mathcal{L}_{\text{pred}} = \frac{1}{HC} \sum_{t=1}^{H} \sum_{c=1}^{C} \ell_\delta\big(\hat{Y}_{t,c}^{(P)} - Y_{t,c}\big). \tag{6}$$

Because the readout can be applied at any position, we add a *next-block prediction* (NBP) term, the analogue of next-token prediction in a causal language model. With $S = [X; Y]$, the head read at position p is trained on the H steps that follow patch p,

$$\mathcal{L}_{\text{nbp}} = \frac{1}{|\mathcal{P}|\, HC} \sum_{p \in \mathcal{P}} \sum_{t=1}^{H} \sum_{c=1}^{C} \ell_\delta\big(\hat{Y}_{t,c}^{(p)} - S_{pP_s+t,\,c}\big), \tag{7}$$

with read positions tiled at stride H/P_s from the first patch with at least 64 steps of context, $\mathcal{P} = \{p_0,\ p_0 + H/P_s, \ldots\} \cap \{p < P\}$, $p_0 = \lceil 64/P_s \rceil$; a longer lookback thus yields more supervised blocks (one at $L = 96$, seven at $L = 672$). The final loss is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda\, \mathcal{L}_{\text{nbp}}, \tag{8}$$

with scaling factor $\lambda = 1$.

5 Experiments

5.1 Setup

Datasets. Standard long-term benchmarks (ETT, Electricity, Weather) have small channel sets, many variables close to their predictability limit (Wang et al., 2025a), and no metadata relating channels, so a channel set cannot be grown as a coherent sub-network. We therefore use nine spatio-temporal datasets whose channels come with location or topology metadata (Appendix A): a *dense* group on a spatial grid or station layout (ERA5 t2m, d2m, and sp; Air Quality) and a *sparse* group of road-sensor networks (PEMS03/04/07/08, METR-LA). The rollout study (Section 5.4) additionally uses ETTh1, Weather, Solar, ECL, and Traffic for comparison with prior work (Liu et al., 2024a).

Table 1: Channel axis, nine datasets. Left: slope s_C (Eq. 1) averaged over datasets; blue marks a robust model ($s_C < 0$), slopes are not ranked. Right: core MSE with all channels (A) and its average rank; best in bold, second underlined. Mean over three seeds.

	↓ Slope (mean over datasets)		↓ Core MSE with all channels (A)									
			Dense				Sparse					
Model	s_C^{A}	s_C^{B}	t2m	d2m	sp	AirQ.	PEMS03	PEMS04	PEMS07	PEMS08	METR-LA	Avg. rank
DLinear	+0.005	+0.017	0.169	0.227	**0.472**	**1.137**	0.406	0.523	0.585	0.428	1.015	4.89
PatchTST	+0.027	−0.018	0.171	0.224	0.491	1.479	0.361	0.581	0.435	0.402	1.115	5.44
iTransformer	+0.023	−0.120	0.253	0.265	0.573	1.517	1.469	0.558	0.789	0.813	0.955	6.56
TimesNet	−0.008	+0.033	0.155	0.214	0.536	4.557	0.232	0.185	0.169	0.332	0.880	4.22
TiDE	−0.002	−0.033	0.150	0.193	0.637	3.751	0.980	1.461	1.070	1.391	1.360	6.22
TimeFilter	−0.004	−0.043	**0.134**	**0.184**	0.638	1.293	**0.177**	0.162	0.161	**0.237**	0.694	**2.11**
MambaTS	+0.043	+0.014	0.154	0.198	0.655	1.348	0.265	0.414	0.303	0.314	0.873	4.11
STREAM	−0.026	−0.029	0.136	0.193	0.697	1.268	0.179	**0.162**	**0.123**	0.308	**0.671**	2.44

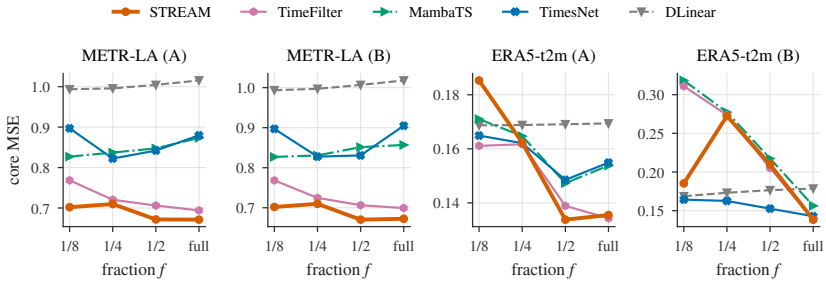


Figure 4: Channel-axis curves under A and B. x: channel fraction f (a separate model per f); y: MSE on the fixed core $V_{1/8}$. Mean over three seeds.

Baselines. The main comparison is with graph-based long-term forecasters: TimeFilter (Hu et al., 2025) and MambaTS (Cai et al., 2024); GG-SSM (Zubić & Scaramuzza, 2025) is compared structurally in Appendix C. For the other families of Section 3.4 we add channel-independent (DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), TiDE (Das et al., 2023)), dense channel mixing (iTransformer (Liu et al., 2024b)), and a temporal-variation model (TimesNet (Wu et al., 2023)). The rollout study adds the autoregressive AutoTimes (Liu et al., 2024a). Every baseline uses the hyperparameters of its official repository (Appendix B).

5.2 Axis 1: robustness to the channel set

Under A, the error of STREAM falls most steadily as informative neighbors are added, and in every dataset group its full-channel error is below that at $f = 1/8$. Channel-independent models stay flat, having no path from a neighbor to the target, while dense mixing (iTransformer) and the fixed scan order of MambaTS get worse, as predicted in Section 3.4. Under B STREAM still does not degrade; the steeper slope of iTransformer reflects a recovery from a poor start at $f = 1/8$ rather than a low error. With all channels, STREAM is best or second on seven of the nine datasets, for the second-best average rank behind TimeFilter. The gap is largest on the sparse road networks, where the relevant neighbors change over time; on the smooth dense grids (ERA5-d2m, ERA5-sp) and on Air Quality a linear model is already competitive, and on ERA5-sp STREAM is behind most baselines. On METR-LA STREAM is lowest at every fraction (Fig. 4). The learned neighborhoods vary smoothly from patch to patch rather than being fixed per window (Appendix D).

Under C (Table 2), STREAM has the steepest spatial-scaling slope on five of the nine datasets and the best average rank for both MSE and MAE: as the monitored area grows, the per-patch graph routes each core channel to surviving neighbors, so a larger network compensates for the failed sensors. Channel-independent models are nearly flat because they never use neighbors, which also shields them from the failures; with few channels, failures remove many of the neighbors STREAM relies on, and its absolute error at $f = 1/8$ is high (Appendix E).

Table 2: Scenario C: spatial-scaling slope s_C^{C} (change of ln MSE per doubling of the monitored area; more negative is better) and its average rank for MSE and MAE. Best in bold, second underlined. Per-fraction errors: Appendix E.

	↓ Spatial-scaling slope s_C^{C} (MSE)									↓ Avg. rank	
Model	t2m	d2m	sp	AirQ.	PEMS03	PEMS04	PEMS07	PEMS08	METR-LA	MSE	MAE
DLinear	+0.00	+0.00	+0.00	**−0.03**	−0.10	−0.07	−0.04	−0.06	+0.00	5.22	4.89
PatchTST	+0.00	+0.00	+0.00	**−0.03**	−0.59	−0.37	−0.73	−0.58	−0.21	4.00	4.11
iTransformer	+0.01	+0.00	+0.00	−0.01	−0.24	+0.09	**−0.74**	+0.02	−0.11	5.44	5.22
TimesNet	**−0.35**	−0.26	+0.01	+0.01	−0.02	−0.56	+0.02	−0.37	−0.14	4.11	4.44
TiDE	−0.29	−0.25	−0.01	−0.01	+0.00	+0.00	+0.00	+0.00	+0.00	5.22	5.56
TimeFilter	−0.30	−0.24	+0.00	+0.05	+0.36	−0.46	+0.66	−0.26	−0.51	5.00	5.56
MambaTS	−0.29	−0.26	**−0.02**	+0.03	+0.36	−0.44	+0.16	−0.48	−0.50	4.33	4.11
STREAM	−0.31	**−0.27**	+0.02	+0.01	**−0.65**	**−0.73**	−0.56	**−0.67**	**−0.69**	**2.67**	**2.11**

Table 3: Lookback axis ($H = 96$, all channels; each model trained and tested at the same L). MSE and rank averaged over the four lookbacks; – : not all lookbacks run. Best in **red bold**, second in blue underline. Per dataset: Appendix F.

Models	STREAM		DLinear		iTransformer		PatchTST		MambaTS		TimeFilter		TimesNet		TiDE	
Metric	MSE	Rank	MSE	Rank	MSE	Rank	MSE	Rank	MSE	Rank	MSE	Rank	MSE	Rank	MSE	Rank
MSE: mean over $L \in \{96, 192, 336, 672\}$; Rank: mean rank among the eight models over the same L																
ERA5 (3)	0.319	3.75	**0.301**	**1.67**	0.345	5.92	0.318	4.08	0.375	7.58	0.340	5.08	0.330	4.25	0.312	3.67
Air Quality	**1.281**	**1.00**	3.946	3.25	4.016	3.50	4.152	4.75	4.624	6.75	4.196	4.75	5.459	8.00	4.031	4.00
PEMS (4)	0.544	3.06	0.703	5.69	0.814	5.94	0.706	5.88	0.580	3.38	0.542	**2.06**	0.677	5.62	**0.495**	4.38
METR-LA	0.904	4.75	0.898	4.75	0.934	6.00	0.982	6.75	0.892	4.25	**0.855**	**2.25**	0.877	2.75	0.969	4.50
All (9)	**0.591**	**3.25**	0.951	3.97	1.027	5.67	0.990	5.25	0.995	5.25	0.915	3.39	1.115	5.11	0.879	4.11

5.3 Axis 2: robustness to lookback length

Across all dataset–lookback pairs, STREAM has the best average rank, ahead of TimeFilter. It is first or second at every lookback on ERA5-t2m, PEMS03 and PEMS07 (Fig. **??**) and keeps improving with longer context on the traffic networks, although its error is not monotone in L on PEMS04, PEMS08 and METR-LA. Dense mixing (iTransformer) degrades with longer context, and TiDE and DLinear have the steepest slopes mainly because they start from a much higher error.

5.4 Axis 3 (extension): error accumulation under autoregressive rollout

For a fair comparison we compare STREAM with the model that is itself trained for autoregressive inference, AutoTimes, under the same two decoding modes (Table 4). STREAM has the lower MSE on four of the five datasets, and on ETTh1, ECL and Traffic its fine-grained rollout, which refreshes the graph and the state after every patch, clearly improves on its own block-wise rollout, whereas fine-grained decoding makes AutoTimes worse on every dataset. On Solar both models degrade under fine-grained decoding and AutoTimes is better. Direct forecasters rolled out block-wise are compared in Appendix **??**.

5.5 Ablation

We ablate the temporal backbone and the training loss on one dataset per domain, with two traffic networks where the channel graph matters most: ETTh1, Solar, PEMS04, PEMS07, and ERA-t2m (Table 5; other datasets in Appendix **??**). Replacing the selective recurrence with self-attention over the same patch tokens degrades every dataset in the decoder-only form, by 5.7% on average and most on the traffic networks, and the encoder-only form is worse on four of five datasets. Reading NBP every two patches instead of every H/P_s patches is worse on all five datasets, and training with MSE instead of the Huber loss costs 2.1% on average.

Table 4: Rollout to $H = 720$ from models trained at $H_0 = 96$ ($L = 96$), against the autoregressive AutoTimes. Block-wise: 96-step blocks; fine-grained: one patch at a time. Best in bold, second underlined, per dataset and metric. Block-wise rollout of every direct forecaster is in Appendix **??**.

	STREAM				AutoTimes			
	block-wise		fine-grained		block-wise		fine-grained	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.565	0.512	**0.535**	0.505	0.541	**0.502**	0.764	0.619
Solar	0.505	0.394	2.532	1.148	**0.410**	**0.361**	2.757	1.228
Weather	**0.356**	**0.350**	0.406	0.387	0.362	0.352	0.391	0.381
ECL	0.350	0.396	**0.210**	**0.305**	0.277	0.348	0.301	0.374
Traffic	0.710	0.422	**0.482**	0.336	0.531	**0.323**	0.593	0.360

Table 5: Ablation: relative MSE change (%) with respect to the full model (positive = worse), $L = H = 96$; the last column averages over the five datasets. ETTh1 and Solar use all channels, the others the core channels of scenario A.

Variant	ETTh1	Solar	PEMS04	PEMS07	ERA-t2m	Avg.
Temporal backbone						
Decoder-only Transformer	+1.6	+1.7	+15.0	+9.9	+0.4	+5.7
Encoder-only Transformer	+5.4	+7.0	+14.2	+6.0	−2.1	+6.1
Loss						
w/o Huber (MSE)	+2.8	+2.8	+0.7	+1.0	+3.0	+2.1
w/o NBP	−0.6	−1.2	+1.0	+1.8	+3.0	+0.8
NBP at every patch	+0.4	+3.4	+3.3	−1.6	−2.4	+0.6
NBP every 2 patches	+0.9	+5.4	+1.3	+0.3	+2.0	+2.0

6 Conclusion

We asked when inter-channel dependency helps multivariate forecasting, and answered it with a protocol that varies what standard benchmarks hold fixed: the channel set (informative, noisy, or failing channels, scored on the same core) and the lookback. STREAM separates the two dependencies: within each patch every channel draws on a fixed budget of k neighbors re-estimated from that patch, and time is carried by a per-channel recurrence linear in the lookback. The results follow this design. Its error falls most as informative neighbors are added, does not grow with noisy channels, and is compensated fastest by a larger network when sensors fail; it has the best average rank across lookbacks and beats the autoregressive AutoTimes under rollout on four of five datasets, whereas channel-independent models cannot use neighbors and dense or window-level graph models degrade as channels or context grow. As limitations, the channel scenarios compare separately trained models; on smooth dense grids a linear model remains competitive; with very few channels, failures hurt STREAM more than channel-independent models; and its rollout does not yet beat the best direct forecasters rolled out block-wise.

Reproducibility Statement

Section 3 defines every stress protocol and metric, and Appendix A–B lists dataset processing, channel selection, hyperparameters, and compute. All numbers are averaged over three seeds $\{1, 2, 3\}$.

Ethics Statement

This work uses publicly available, non-personal spatio-temporal benchmarks (traffic sensors, climate reanalysis, photovoltaic output, air quality) and raises no human-subjects or privacy concerns of which we are aware.

References

Abdul Fatir Ansari et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research (TMLR)*, 2024.

Samy Bengio et al. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Xiuding Cai et al. MambaTS: Improved selective state space models for long-term time series forecasting. *arXiv preprint arXiv:2405.16440*, 2024.

Chao Chen et al. Freeway performance measurement system: Mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001.

Yu Chen, Nathalia Céspedes, and Payam Barnaghi. A closer look at Transformers for time series forecasting: Understanding why they work and where they struggle. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 7763–7780, 2025.

Abhimanyu Das et al. Long-term forecasting with TiDE: Time-series dense encoder. *Transactions on Machine Learning Research (TMLR)*, 2023.

Abhimanyu Das et al. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning (ICML)*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Hans Hersbach et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Yifan Hu et al. TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 24893–24911, 2025.

Taesung Kim et al. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations (ICLR)*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Guokun Lai et al. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

Dongyuan Li et al. DyG-Mamba: Continuous state space modeling on dynamic graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yaguang Li et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Zheng Li, Jerry Cheng, and Huanying Gu. AROpt: An optimization method for autoregressive time series forecasting. *arXiv preprint arXiv:2602.02288*, 2026.

Yong Liu et al. AutoTimes: Autoregressive time series forecasters via large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Yong Liu et al. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024b.

Yong Liu et al. Timer: Generative pre-trained transformers are large time series models. In *International Conference on Machine Learning (ICML)*, 2024c.

Yuqi Nie et al. A time series is worth 64 words: Long-term forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Adam Paszke et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Chao Song et al. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Shiyu Wang et al. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024.

Yuxuan Wang et al. Exploring accuracy law for deep time series forecasters: An empirical study. *arXiv preprint arXiv:2510.02729*, 2025a.

Zihan Wang et al. Is Mamba effective for time series forecasting? *Neurocomputing*, 619: 129178, 2025b. doi: 10.1016/j.neucom.2024.129178.

Haixu Wu et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 22419–22430, 2021.

Haixu Wu et al. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations (ICLR)*, 2023.

Zonghan Wu et al. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

Ailing Zeng et al. Are Transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pp. 11121–11128, 2023.

Jiahui Zhang et al. Enhancing the maximum effective window for long-term time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yu Zheng et al. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 2267–2276, 2015.

Haoyi Zhou et al. Informer: Beyond efficient Transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 11106–11115, 2021.

Nikola Zubić and Davide Scaramuzza. GG-SSMs: Graph-generating state space models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28863–28873, 2025.

AI Use Statement

AI assistants were used for language editing, LaTeX formatting, and help with experiment-management scripts. All research ideas, experimental design, analyses, and claims are the authors' own, and the authors verified every reported number against the underlying run artifacts.

A Datasets and channel-selection protocol

Dataset descriptions. Table 6 summarizes the datasets; they are described below. (1) **ERA5** (Hersbach et al., 2020) is the fifth-generation global atmospheric reanalysis of ECMWF, distributed hourly on a regular $0.25° \times 0.25°$ latitude–longitude grid. We take

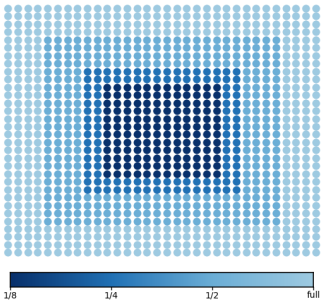
Table 6: Datasets. C: channels; T: time steps. Axis membership: 1 = channel, 2 = lookback, 3 = rollout. Splits are chronological (7:1:2; ETT 12/4/4 months). The general benchmarks have no channel graph and are used only for the rollout study.

Group	Dataset	C	T	Sampling	Channel graph	Axes
Spatio-Temporal	ERA5 (t2m, d2m, sp)	1,024	21,912	1 h	32×32 lat–lon grid	1, 2
	Air Quality	437	8,760	1 h	station distance, pruned	1, 2
	PEMS03	358	26,208	5 min	road adjacency	1, 2
	PEMS04	307	16,992	5 min	road adjacency	1, 2
	PEMS07	883	28,224	5 min	road adjacency	1, 2
	PEMS08	170	17,856	5 min	road adjacency	1, 2
	METR-LA	207	34,272	5 min	road distance	1, 2
General	ETTh1	7	17,420	1 h	—	3
	Weather	21	52,696	10 min	—	3
	Solar	137	52,560	10 min	—	3
	ECL	321	26,304	1 h	—	3
	Traffic	862	17,544	1 h	—	3

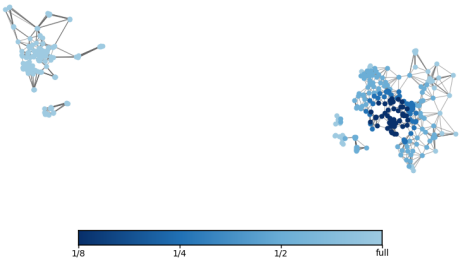
the hourly single-level fields from 2019 to 2023 and use three surface variables, 2 m temperature (t2m), 2 m dewpoint temperature (d2m), and surface pressure (sp), each as a separate dataset. The channels are the 32×32 grid points (about $8° \times 8°$) closest to 37.5°N, 127.5°E, so the channel graph is the grid itself. All three fields are smooth and highly correlated across the grid; t2m and d2m follow a strong daily cycle, whereas sp is driven mainly by passing synoptic systems. (2) **Air Quality** (Zheng et al., 2015) contains hourly PM2.5 concentrations from 437 monitoring stations in 43 Chinese cities over one year (May 2014 to April 2015). Its daily cycle is weak, its stations are only loosely correlated, and it contains a large share of missing values; the channel graph is built from station distances. (3) **PEMS03, PEMS04, PEMS07, and PEMS08** (Song et al., 2020) are traffic-flow records from the Caltrans Performance Measurement System (Chen et al., 2001), collected by loop detectors in four districts of California and aggregated into 5-minute intervals (288 steps per day). They have a clean daily cycle, and the sensors are connected by the road network. (4) **METR-LA** (Li et al., 2018) contains traffic speed from 207 loop detectors on the highways of Los Angeles County from March to June 2012, at 5-minute intervals. Speeds are noisier and less periodic than PEMS flows, and the channel graph follows road-network distance. (5) The **general benchmarks** have no channel graph and enter only the rollout study. ETTh1 (Zhou et al., 2021) records the oil temperature and six load features of an electricity transformer in China, hourly from July 2016 to June 2018. Weather (Wu et al., 2021) contains 21 meteorological indicators recorded every 10 minutes in Germany during 2020. Solar (Lai et al., 2018) contains the power output of 137 photovoltaic plants in Alabama during 2006, every 10 minutes. ECL (Wu et al., 2021) contains the hourly electricity consumption of 321 clients from 2012 to 2014, and Traffic (Wu et al., 2021) the hourly road occupancy measured by 862 sensors on San Francisco Bay Area freeways in 2015 and 2016.

Breadth-first channel fractions. Starting from a center channel, a breadth-first expansion over the channel graph (road adjacency, station distance, or the spatial grid) adds the nearest connected channels first. The visiting order defines the fractions $1/8 \subset 1/4 \subset 1/2 \subset$ full, so each fraction is a spatially contiguous sub-network rather than a random subset (Fig. 5). For ERA the expansion runs over the grid, for the traffic networks over the road adjacency, and for Air Quality over a distance-based kNN graph pruned at a fixed threshold. The B noise channels are drawn from the non-core channels and the C failed set E from all channels, each once from the full set with a fixed generator (Section 3.1). Only the 1/8 core is scored, minus E under C.

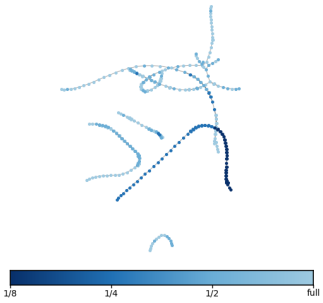
Scoring. We report MSE and MAE on instance-normalized data, averaged over seeds $\{1, 2, 3\}$. Validation and early stopping use the core-channel MSE, which is also the reported metric, and under C the scored target is $V_{1/8} \setminus E$ at every fraction. Since channels are z-scored with training statistics, a B noise channel has the scale of a real channel but carries no information about any other. Every lookback is a multiple of the patch size. Block-wise



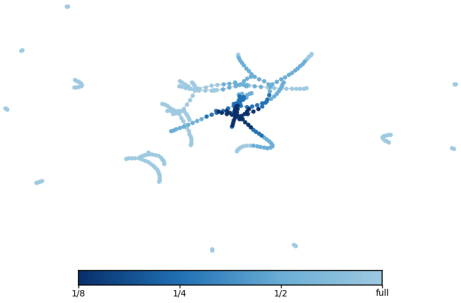
(a) ERA-t2m, d2m, sp



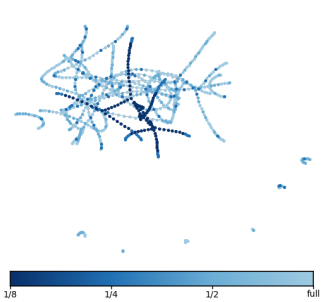
(b) Air Quality



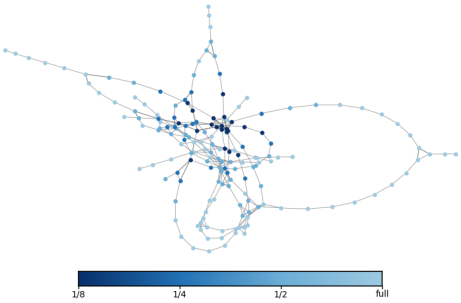
(c) PEMS03



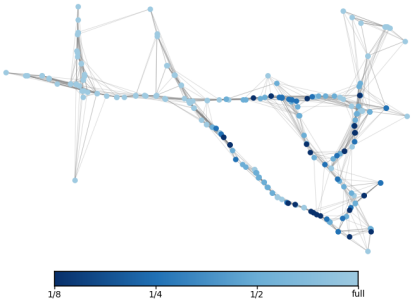
(d) PEMS04



(e) PEMS07



(f) PEMS08



(g) METR-LA

Figure 5: Breadth-first channel fractions. Each panel shows the fraction at which a channel enters. Dark regions denote the 1/8 core that is scored, while light regions denote channels added only at full. Each fraction forms a contiguous region of the underlying graph.

rollout to $H = 720$ uses eight H_0-step blocks, the last truncated to 48 steps; the rollout bin error is $e_b = \mathrm{MAE}\big(\hat{Y}_{(b-1)H_0+1:bH_0}\big)$ for $b = 1, \ldots, 8$.

B Implementation details

STREAM. All experiments are implemented in PyTorch (Paszke et al., 2019) and run on NVIDIA H100 and A100 GPUs. We use the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 3×10^{-4} halved every epoch, dropout 0.1, at most 10 epochs and early stopping with patience 3. The effective batch size is 32 at $L = 96$ and 16 at $L = 672$; when a configuration does not fit in memory the micro-batch is halved and gradients are accumulated, so every channel fraction sees the same optimization. The architecture is shared by all datasets: instance normalization over the whole lookback, two graph layers with 8 heads, a $k = 8$ neighborhood and a feed-forward width of 512, two node-stream Mamba layers (Gu & Dao, 2024) per channel (state 32, expansion 2, convolution 4), one graph-stream Mamba layer, a learned channel embedding without positional embedding, and a readout that concatenates the last $\kappa = \text{round}(P/2)$ states, i.e. half of the lookback. The loss weight is $\lambda = 1$ (Eq. 8) and the Huber threshold is $\delta = 0.5$ (Eq. 5). Table 7 lists the settings per dataset; only the patch size P_s, the width D and the selectivity of Δ change, and none of them changes the set of modules. D follows the size of the full channel set, since $D = 512$ overfits the small general benchmarks (the validation minimum falls at the first epoch). On ERA the step size Δ of the node stream is made input-independent by fixing the weight of its input projection to zero, which leaves the set of parameters unchanged. Within a dataset the configuration is identical at every channel fraction and every lookback, with κ following P. Instance normalization uses the statistics of the whole lookback, so during training the intermediate NBP read positions are normalized with statistics that also cover later lookback steps; the forecast itself uses only the lookback, as in every baseline.

Table 7: STREAM hyperparameters per dataset. κ: readout length at $L = 96$ / $L = 672$. Layers: graph / node stream / graph stream. Δ: input-dependent (dep.) or input-independent (const.). Epochs: maximum, with early stopping.

Group	Dataset	P_s	κ	Layers	lr	D	FFN	Δ	Epochs
Spatio-Temporal	ERA5	32	2/10	2/2/1	3e-4	512	512	const.	10
	Air Quality	8	6/42	2/2/1	3e-4	512	512	dep.	10
	PEMS03	8	6/42	2/2/1	3e-4	512	512	dep.	10
	PEMS04	8	6/42	2/2/1	3e-4	256	512	dep.	10
	PEMS07	8	6/42	2/2/1	3e-4	512	512	dep.	10
	PEMS08	8	6/42	2/2/1	3e-4	256	512	dep.	10
	METR-LA	8	6/42	2/2/1	3e-4	256	512	dep.	10
General	ETTh1	16	3/21	2/2/1	3e-4	256	512	dep.	10
	Weather	16	3/21	2/2/1	3e-4	128	512	dep.	10
	Solar	16	3/21	2/2/1	3e-4	512	512	dep.	10
	ECL	16	3/21	2/2/1	3e-4	512	512	dep.	10
	Traffic	4	12/84	2/2/1	3e-4	512	512	dep.	10

Baselines. Each baseline uses the hyperparameters of its official repository for the dataset in question (learning rate, depth, width, patch length, and schedule). Where an official script does not exist for a dataset, the recipe of the closest dataset of the same family is used. TimeFilter ties its patch length to the lookback as in its official code, which fixes its joint graph at C nodes for every L; its earlier fixed-patch configuration runs out of memory at $L = 672$ on PEMS07 and ERA.

C Comparison with graph-aware sequence models

Complexity and rollout cost. Because the graph is local to a patch and patches communicate only through the two causal streams, the model factorizes over patches. After one prefill over the lookback, the recurrent states, the last κ node states, and s_{p-1} are kept in a cache, and extending the forecast by a patch costs one graph step and one recurrent update per stream. Block-wise rollout (Section 3.3) appends the H_0-step output as for every baseline; fine-grained rollout refreshes the graph and the state after every patch. Table 9 lists the per-layer cost.

Table 8: Graph-aware sequence models on a $C \times P$ patch grid.

	STREAM	TimeFilter	MambaTS	GG-SSM
Channel graph	kNN over channels, fixed budget k	kNN over key–query similarity	none at inference; a learned variable order	minimum spanning tree from feature similarity
Graph scope	channels within each patch	joint over all $C \cdot P$ tokens	one chain over all $C \cdot P$ tokens (learned channel order, then time)	one tree over all $C \cdot P$ tokens (temporal tree first, channels unordered)
Update rate	rebuilt every patch	once per window	fixed after training	once per window
Temporal model	selective recurrence across patches, per channel	inside the joint graph (GCN)	Mamba over one channel-first scan	SSM scan along the tree
Patch size	small to moderate (8–32)	one patch per window ($P_s = L$) officially; 48–96 when fixed	large (48–96)	large (48–96)
Rollout	patch-wise, constant per patch	re-run on the full window	re-run on the full window	re-run on the full window

Table 9: Per-layer cost on a $C \times P$ patch grid ($P = L/P_s$; d: head width, D: model width, k: neighborhood size).

Component	Compute	Memory	Per rollout patch
STREAM graph (within patch)	$O(P \cdot C^2 \cdot d)$	$O(P \cdot C \cdot k)$	$O(C^2 \cdot d)$
STREAM node stream (per channel)	$O(C \cdot P \cdot D)$	$O(C \cdot D)$ state	$O(C \cdot D)$
STREAM graph stream	$O(P \cdot D)$	$O(D)$ state	$O(D)$
Joint graph, fixed P_s (Hu et al., 2025)	$O((C \cdot P)^2 \cdot d)$	$O((C \cdot P)^2)$	re-run on the window
Joint graph, $P_s = L$ (Hu et al., 2025)	$O(C^2 \cdot d)$	$O(C^2)$	re-run on the window

D Temporal variation of the per-patch graph

Because STREAM rebuilds its graph for every patch, we measure how the selected neighborhoods change within one window. For a pair of patches $(p, p+s)$ we compute two quantities per layer, averaged over all pairs at distance s and over several test windows. *Kept* is the fraction of the k keys that a query selects at patch p and selects again at patch $p + s$, computed for each head's own top-k and averaged over heads. *Score rank* is the median percentile of a pair selected at p within the query's score row at $p + s$ (100: still ranked first; 50: unrelated). Figure 6 shows that adjacent patches keep a large share of each other's neighbors and still rank them highly, and that both quantities decrease as the distance between patches grows: the graph is not a single structure per window but varies smoothly along it.

Graph scores and context correlation. Following the analysis of Liu et al. (2024b), Figure 7 compares the channel correlation of the context window with the pre-softmax score maps of both graph layers (head mean, averaged over patches). In every case we examined, the score map is more strongly related to the correlation of the context than to that of the future window, and the relation to the future does not grow with depth. The graph is thus estimated from the context itself, while predicting the future is left to the recurrence, unlike a direct forecaster whose attention drifts toward future correlations in deeper layers (Liu et al., 2024b).

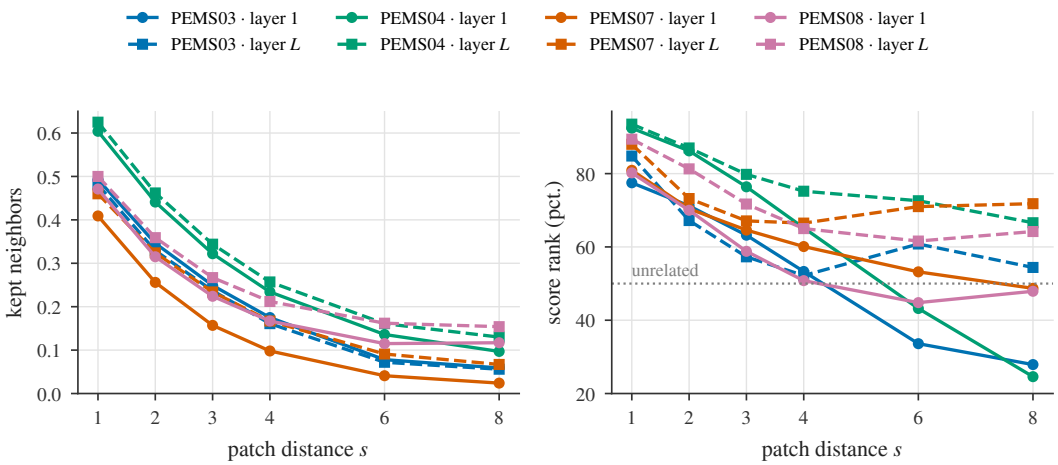


Figure 6: Per-patch graphs drift smoothly within a window. Left: fraction of a query's selected neighbors that are selected again s patches later. Right: median score percentile, s patches later, of the previously selected pairs; dotted line: unrelated.

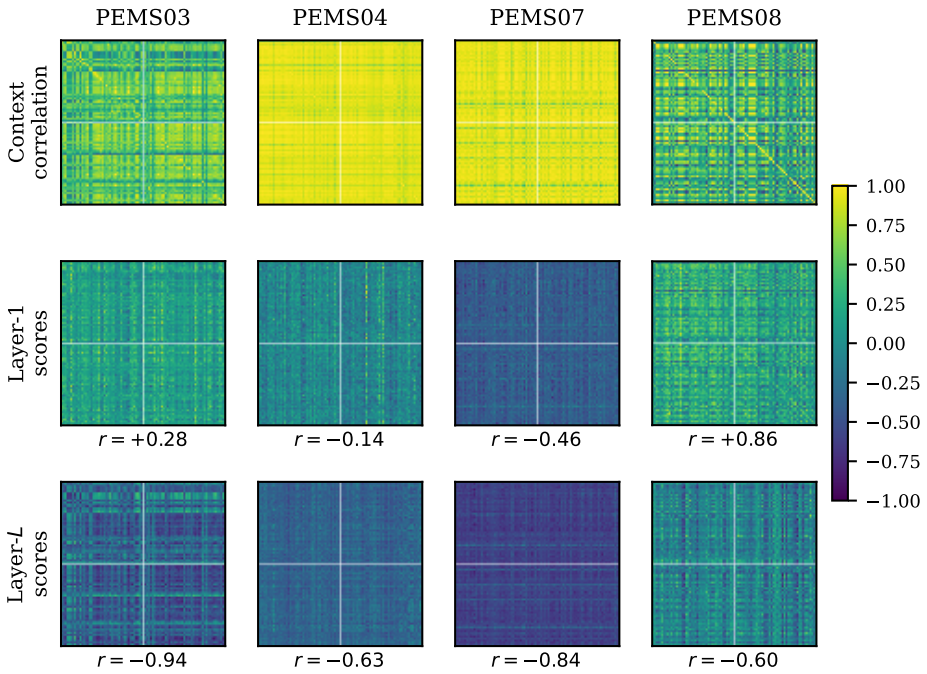


Figure 7: Context correlation (top) and the score maps of the first and last (L-th) graph layers for one test window per dataset; r is the Pearson correlation with the context correlation over off-diagonal entries. Scores are scaled by their maximum magnitude.

E Full channel-axis results

Table 10: Channel axis, scenario A (informative channels), every dataset: core MSE/MAE per channel fraction f (mean over three seeds). Best in bold.

Models		STREAM		DLinear		iTransformer		PatchTST		MambaTS		TimeFilter		TimesNet		TiDE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ERA5-t2m	1/8	0.185	0.310	0.169	0.292	0.244	0.365	0.170	0.294	0.171	0.295	**0.161**	**0.289**	0.165	0.294	0.173	0.296
	1/4	**0.162**	0.291	0.169	0.292	0.253	0.375	0.170	0.294	0.165	**0.287**	**0.162**	0.290	**0.162**	0.295	0.166	0.289
	1/2	**0.134**	**0.264**	0.169	0.292	0.253	0.375	0.171	0.294	0.147	0.270	0.139	0.267	0.148	0.282	0.146	0.271
	full	0.136	0.270	0.169	0.293	0.253	0.375	0.171	0.294	0.154	0.283	**0.134**	**0.269**	0.155	0.288	0.150	0.281
	Avg	0.154	0.284	0.169	0.293	0.251	0.373	0.170	0.294	0.159	0.284	**0.149**	**0.279**	0.158	0.290	0.159	0.284
ERA5-d2m	1/8	0.239	0.346	0.225	**0.334**	0.266	0.367	**0.224**	**0.334**	0.239	0.345	0.229	0.338	0.235	0.342	0.232	0.338
	1/4	0.221	0.334	0.226	0.335	0.264	0.366	0.224	0.333	0.225	0.335	**0.213**	**0.328**	0.230	0.336	0.218	0.329
	1/2	0.193	0.317	0.226	0.334	0.261	0.365	0.223	0.332	0.202	0.323	**0.190**	**0.316**	0.232	0.336	0.196	0.319
	full	0.193	0.326	0.227	0.334	0.265	0.368	0.224	0.332	0.198	0.328	**0.184**	**0.319**	0.214	0.328	0.193	0.325
	Avg	0.211	0.331	0.226	0.334	0.264	0.367	0.224	0.333	0.216	0.333	**0.204**	**0.325**	0.228	0.335	0.210	0.328
ERA5-sp	1/8	0.506	0.552	**0.473**	**0.526**	0.572	0.598	0.491	0.537	0.512	0.552	0.487	0.535	0.503	0.543	0.504	0.549
	1/4	0.539	0.572	**0.473**	**0.526**	0.572	0.598	0.492	0.538	0.526	0.561	0.499	0.544	0.511	0.557	0.516	0.557
	1/2	0.590	0.591	**0.472**	**0.526**	0.572	0.598	0.492	0.539	0.581	0.590	0.555	0.574	0.505	0.551	0.563	0.582
	full	0.697	0.640	**0.472**	**0.526**	0.573	0.598	0.491	0.539	0.655	0.624	0.638	0.611	0.536	0.565	0.637	0.616
	Avg	0.583	0.589	**0.472**	**0.526**	0.572	0.598	0.492	0.538	0.569	0.581	0.545	0.566	0.514	0.554	0.555	0.576
Air Quality	1/8	1.362	0.663	**1.124**	**0.614**	1.306	0.641	1.319	0.636	1.381	0.646	1.301	0.626	4.522	1.057	3.834	0.930
	1/4	1.402	0.668	**1.123**	**0.616**	1.552	0.678	1.478	0.660	1.433	0.656	1.279	0.627	4.580	1.007	3.801	0.922
	1/2	1.359	0.662	**1.126**	0.619	1.564	0.667	1.572	0.670	1.358	0.645	1.259	**0.616**	4.408	0.989	3.728	0.913
	full	1.268	0.626	**1.137**	**0.609**	1.517	0.658	1.479	0.648	1.348	0.646	1.293	0.624	4.557	1.047	3.751	0.918
	Avg	1.348	0.655	**1.128**	**0.614**	1.484	0.661	1.462	0.654	1.380	0.648	1.283	0.623	4.517	1.025	3.779	0.921
PEMS03	1/8	**0.178**	**0.297**	0.398	0.491	0.545	0.562	0.287	0.369	0.235	0.345	0.196	0.316	0.241	0.347	0.966	0.761
	1/4	0.226	0.332	0.406	0.495	0.370	0.447	0.326	0.401	0.235	0.345	**0.182**	**0.303**	0.227	0.329	0.972	0.763
	1/2	0.187	0.306	0.403	0.493	1.185	0.863	0.347	0.413	0.243	0.352	**0.173**	**0.296**	0.224	0.328	0.971	0.763
	full	0.179	**0.298**	0.406	0.494	1.469	0.836	0.361	0.421	0.265	0.368	**0.177**	0.300	0.232	0.337	0.980	0.768
	Avg	0.193	0.309	0.403	0.493	0.892	0.677	0.330	0.401	0.244	0.353	**0.182**	**0.304**	0.231	0.335	0.972	0.764
PEMS04	1/8	**0.135**	**0.250**	0.513	0.521	0.708	0.641	0.469	0.478	0.266	0.361	0.146	0.266	0.206	0.325	1.450	0.968
	1/4	**0.129**	**0.242**	0.515	0.519	0.561	0.554	0.495	0.495	0.292	0.378	0.145	0.266	0.185	0.304	1.456	0.965
	1/2	**0.149**	**0.258**	0.522	0.520	0.543	0.536	0.538	0.512	0.352	0.410	0.153	0.275	0.179	0.301	1.466	0.959
	full	**0.162**	**0.271**	0.523	0.521	0.558	0.551	0.581	0.530	0.414	0.443	**0.162**	0.284	0.185	0.309	1.461	0.958
	Avg	**0.143**	**0.255**	0.518	0.520	0.592	0.570	0.521	0.504	0.331	0.398	0.151	0.273	0.189	0.310	1.458	0.963
PEMS07	1/8	**0.134**	**0.236**	0.558	0.533	1.467	0.975	0.394	0.425	0.227	0.323	0.162	0.267	0.175	0.273	1.058	0.771
	1/4	**0.127**	**0.230**	0.566	0.534	1.688	1.067	0.402	0.431	0.243	0.335	0.147	0.251	0.166	0.265	1.061	0.772
	1/2	**0.128**	**0.233**	0.578	0.537	0.965	0.738	0.421	0.444	0.274	0.357	0.153	0.258	0.167	0.266	1.066	0.775
	full	**0.123**	**0.229**	0.585	0.540	0.789	0.669	0.435	0.452	0.303	0.378	0.161	0.268	0.169	0.271	1.070	0.776
	Avg	**0.128**	**0.232**	0.572	0.536	1.227	0.862	0.413	0.438	0.262	0.348	0.156	0.261	0.169	0.269	1.064	0.773
PEMS08	1/8	0.700	0.490	0.425	0.494	1.645	1.006	0.454	0.451	0.281	0.362	**0.229**	**0.321**	0.313	0.378	1.382	0.948
	1/4	0.384	0.388	0.424	0.493	1.206	0.847	0.424	0.443	0.306	0.367	**0.253**	**0.334**	0.327	0.385	1.386	0.951
	1/2	0.346	0.379	0.425	0.494	1.045	0.780	0.423	0.445	0.317	0.374	**0.275**	**0.347**	0.327	0.389	1.391	0.953
	full	0.308	0.361	0.428	0.496	0.813	0.674	0.402	0.436	0.314	0.377	**0.237**	**0.328**	0.332	0.396	1.391	0.952
	Avg	0.435	0.404	0.425	0.494	1.177	0.827	0.426	0.444	0.305	0.370	**0.248**	**0.332**	0.325	0.387	1.387	0.951
METR-LA	1/8	**0.702**	**0.486**	0.994	0.650	1.336	0.717	1.031	0.623	0.827	0.566	0.768	0.521	0.897	0.570	1.366	0.752
	1/4	**0.710**	**0.486**	0.996	0.650	1.244	0.690	1.035	0.616	0.837	0.567	0.720	0.501	0.822	0.538	1.365	0.746
	1/2	**0.672**	**0.472**	1.004	0.652	1.080	0.624	1.059	0.615	0.847	0.571	0.706	0.492	0.842	0.541	1.368	0.749
	full	**0.671**	**0.464**	1.015	0.657	0.955	0.550	1.115	0.602	0.873	0.577	0.694	0.490	0.880	0.549	1.360	0.747
	Avg	**0.689**	**0.477**	1.002	0.653	1.154	0.645	1.060	0.614	0.846	0.570	0.722	0.501	0.860	0.550	1.365	0.749
# 1st		**30**		16		0		2		1		26		1		0	

Table 11: Channel axis, scenario B (noisy channels), every dataset: core MSE/MAE per channel fraction f (mean over three seeds). Best in bold.

Models		STREAM		DLinear		iTransformer		PatchTST		MambaTS		TimeFilter		TimesNet		TiDE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ERA5-t2m	1/8	0.185	0.310	0.169	**0.292**	0.244	0.365	0.170	0.294	0.318	0.384	0.311	0.380	**0.164**	0.294	0.329	0.392
	1/4	0.272	0.356	0.173	0.297	0.259	0.381	0.170	0.294	0.277	0.357	0.274	0.357	**0.163**	**0.293**	0.286	0.364
	1/2	0.210	0.310	0.176	0.301	0.219	0.344	0.172	0.295	0.217	0.313	0.205	0.305	**0.153**	**0.285**	0.226	0.322
	full	**0.138**	**0.270**	0.179	0.303	0.200	0.327	0.172	0.296	0.156	0.284	0.139	**0.270**	0.143	0.277	0.163	0.292
	Avg	0.201	0.311	0.174	0.299	0.231	0.354	0.171	0.295	0.242	0.334	0.232	0.328	**0.156**	**0.287**	0.251	0.342
ERA5-d2m	1/8	0.239	0.346	0.225	**0.334**	0.266	0.367	**0.224**	**0.334**	0.372	0.424	0.368	0.419	0.231	0.338	0.373	0.423
	1/4	0.405	0.420	0.234	0.344	0.260	0.364	**0.223**	**0.333**	0.326	0.396	0.318	0.389	0.228	0.338	0.328	0.397
	1/2	0.257	0.355	0.237	0.346	0.259	0.363	**0.223**	**0.333**	0.268	0.363	0.262	0.357	0.265	0.364	0.269	0.365
	full	0.194	**0.323**	0.234	0.345	0.258	0.362	0.224	0.333	0.204	0.332	0.195	0.326	**0.192**	0.326	0.202	0.334
	Avg	0.274	0.361	0.233	0.342	0.261	0.364	**0.224**	**0.333**	0.293	0.379	0.286	0.373	0.229	0.341	0.293	0.380
ERA5-sp	1/8	0.506	0.552	**0.473**	**0.526**	0.572	0.598	0.491	0.537	0.601	0.597	0.588	0.587	0.601	0.597	0.610	0.603
	1/4	0.602	0.600	**0.484**	**0.537**	0.572	0.598	0.497	0.539	0.591	0.594	0.576	0.583	0.592	0.595	0.595	0.598
	1/2	0.615	0.604	**0.488**	0.540	0.560	0.590	0.495	**0.539**	0.620	0.609	0.597	0.595	0.627	0.615	0.614	0.609
	full	0.669	0.625	**0.489**	0.541	0.573	0.597	0.494	**0.539**	0.651	0.621	0.634	0.611	0.681	0.638	0.647	0.623
	Avg	0.598	0.595	**0.484**	**0.536**	0.569	0.596	0.494	0.538	0.616	0.605	0.599	0.594	0.626	0.611	0.616	0.608
Air Quality	1/8	1.362	0.663	**1.173**	0.638	1.394	0.692	1.430	0.692	1.381	0.646	1.301	**0.626**	4.499	1.043	3.834	0.930
	1/4	1.398	0.670	**1.191**	0.626	1.451	0.662	1.384	0.639	1.327	0.630	1.267	**0.625**	4.861	1.060	3.791	0.922
	1/2	1.367	0.667	**1.195**	0.628	1.331	0.637	1.272	0.619	1.323	0.635	1.238	**0.612**	4.403	1.014	3.713	0.913
	full	1.373	0.649	1.356	0.640	1.266	0.625	**1.215**	**0.610**	1.304	0.632	1.276	0.618	4.264	0.993	3.732	0.918
	Avg	1.375	0.662	**1.229**	0.633	1.361	0.654	1.325	0.640	1.334	0.636	1.270	**0.620**	4.507	1.027	3.767	0.921
PEMS03	1/8	**0.178**	**0.297**	0.398	0.491	1.060	0.711	1.484	0.597	0.235	0.345	0.196	0.316	0.229	0.337	0.966	0.761
	1/4	0.228	0.333	0.406	0.497	0.456	0.493	0.322	0.389	0.239	0.347	**0.193**	**0.312**	0.227	0.330	0.971	0.765
	1/2	0.195	0.311	0.405	0.497	0.360	0.440	0.330	0.395	0.252	0.356	**0.180**	**0.301**	0.229	0.332	0.976	0.769
	full	0.257	0.350	0.408	0.500	0.403	0.470	0.345	0.403	0.267	0.367	**0.169**	**0.292**	0.237	0.341	0.989	0.777
	Avg	0.214	0.323	0.404	0.496	0.570	0.528	0.620	0.446	0.249	0.354	**0.184**	**0.305**	0.230	0.335	0.976	0.768
PEMS04	1/8	**0.135**	**0.250**	0.513	0.521	0.708	0.641	0.469	0.478	0.266	0.361	0.146	0.266	0.206	0.325	1.450	0.968
	1/4	**0.129**	**0.242**	0.521	0.525	0.494	0.507	0.498	0.495	0.299	0.381	0.142	0.263	0.188	0.303	1.456	0.966
	1/2	**0.137**	**0.249**	0.533	0.529	0.435	0.468	0.530	0.508	0.373	0.421	0.144	0.266	0.195	0.311	1.464	0.960
	full	**0.142**	**0.258**	0.538	0.533	0.352	0.418	0.564	0.522	0.442	0.453	0.153	0.275	0.338	0.421	1.460	0.959
	Avg	**0.136**	**0.250**	0.526	0.527	0.497	0.509	0.515	0.501	0.345	0.404	0.146	0.267	0.232	0.340	1.457	0.963
PEMS07	1/8	**0.134**	**0.236**	0.558	0.533	1.467	0.975	0.400	0.428	0.228	0.323	0.162	0.267	0.177	0.274	1.058	0.771
	1/4	0.155	0.255	0.571	0.541	0.889	0.713	0.404	0.439	0.242	0.335	**0.144**	**0.248**	0.173	0.270	1.066	0.779
	1/2	**0.121**	**0.225**	0.585	0.546	0.333	0.406	0.424	0.453	0.270	0.356	0.144	0.249	0.174	0.271	1.076	0.786
	full	**0.124**	**0.229**	0.597	0.552	0.342	0.407	0.440	0.463	0.305	0.381	0.146	0.251	0.170	0.271	1.083	0.790
	Avg	**0.133**	**0.236**	0.578	0.543	0.758	0.625	0.417	0.446	0.261	0.349	0.149	0.254	0.173	0.271	1.071	0.782
PEMS08	1/8	0.700	0.490	0.425	0.494	1.645	1.006	0.454	0.451	0.281	0.362	**0.229**	**0.321**	0.313	0.378	1.382	0.948
	1/4	0.411	0.398	0.427	0.497	0.545	0.528	0.465	0.457	0.318	0.378	**0.226**	**0.320**	0.308	0.370	1.387	0.951
	1/2	0.308	0.373	0.430	0.500	0.440	0.463	0.457	0.460	0.311	0.375	**0.279**	**0.343**	0.310	0.378	1.391	0.954
	full	0.275	0.345	0.435	0.504	0.393	0.433	0.428	0.448	0.316	0.379	**0.232**	**0.326**	0.475	0.481	1.392	0.954
	Avg	0.423	0.401	0.429	0.499	0.756	0.607	0.451	0.454	0.307	0.374	**0.241**	**0.327**	0.352	0.402	1.388	0.952
METR-LA	1/8	**0.702**	**0.486**	0.994	0.650	1.336	0.717	1.031	0.623	0.827	0.566	0.768	0.521	0.897	0.570	1.366	0.752
	1/4	**0.710**	**0.485**	0.997	0.653	1.186	0.670	1.039	0.619	0.831	0.566	0.725	0.503	0.828	0.540	1.362	0.745
	1/2	**0.670**	**0.471**	1.007	0.655	0.974	0.580	1.100	0.621	0.851	0.568	0.706	0.491	0.831	0.537	1.365	0.748
	full	**0.672**	**0.463**	1.018	0.660	0.950	0.550	1.107	0.604	0.857	0.568	0.699	0.490	0.905	0.553	1.360	0.748
	Avg	**0.689**	**0.476**	1.004	0.654	1.111	0.629	1.069	0.617	0.842	0.567	0.725	0.501	0.865	0.550	1.364	0.748
# 1st		**27**		11		0		10		0		20		6		0	

Table 12: Channel axis, scenario C (test-time channel drop), every dataset: core MSE/MAE per channel fraction f (mean over three seeds). Best in bold.

Models		STREAM		DLinear		iTransformer		PatchTST		MambaTS		TimeFilter		TimesNet		TiDE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ERA5-t2m	1/8	0.454	0.447	**0.169**	**0.292**	0.244	0.366	**0.169**	0.293	0.390	0.414	0.384	0.412	0.466	0.439	0.381	0.409
	1/4	0.449	0.429	**0.169**	**0.292**	0.253	0.375	0.170	0.294	0.366	0.397	0.345	0.393	0.405	0.408	0.356	0.393
	1/2	0.373	0.370	**0.169**	**0.293**	0.253	0.375	0.171	0.294	0.274	0.342	0.299	0.352	0.338	0.361	0.266	0.337
	full	0.169	0.291	0.170	0.293	0.253	0.375	0.171	0.294	0.163	0.289	**0.149**	**0.280**	0.156	0.286	0.161	0.287
	Avg	0.361	0.384	**0.169**	**0.293**	0.250	0.372	0.170	0.294	0.298	0.360	0.294	0.359	0.341	0.374	0.291	0.357
ERA5-d2m	1/8	0.450	0.457	**0.225**	**0.334**	0.266	0.367	**0.225**	0.335	0.448	0.452	0.406	0.436	0.471	0.460	0.426	0.442
	1/4	0.477	0.453	**0.225**	**0.334**	0.264	0.366	**0.225**	**0.334**	0.417	0.434	0.379	0.419	0.418	0.434	0.394	0.424
	1/2	0.340	0.387	0.226	0.334	0.261	0.365	**0.224**	**0.333**	0.322	0.386	0.302	0.377	0.317	0.384	0.307	0.379
	full	0.208	0.330	0.226	0.334	0.265	0.368	0.225	0.333	0.208	0.333	**0.195**	**0.326**	0.216	0.339	0.202	0.331
	Avg	0.369	0.407	0.226	**0.334**	0.264	0.366	**0.225**	**0.334**	0.349	0.401	0.321	0.389	0.356	0.404	0.332	0.394
ERA5-sp	1/8	0.640	0.613	**0.474**	**0.527**	0.574	0.599	0.494	0.538	0.703	0.634	0.651	0.608	0.670	0.623	0.660	0.619
	1/4	0.653	0.622	**0.474**	**0.527**	0.574	0.599	0.494	0.539	0.701	0.636	0.629	0.604	0.658	0.623	0.657	0.620
	1/2	0.651	0.623	**0.474**	**0.527**	0.574	0.599	0.494	0.540	0.688	0.636	0.634	0.609	0.691	0.640	0.646	0.618
	full	0.688	0.642	**0.474**	**0.527**	0.574	0.599	0.493	0.539	0.666	0.628	0.644	0.613	0.680	0.639	0.645	0.619
	Avg	0.658	0.625	**0.474**	**0.527**	0.574	0.599	0.494	0.539	0.690	0.634	0.640	0.608	0.675	0.631	0.652	0.619
Air Quality	1/8	1.553	0.725	**1.224**	0.663	1.312	0.649	1.310	0.631	1.381	0.646	1.301	**0.626**	3.841	1.014	3.341	0.908
	1/4	1.635	0.683	**1.226**	**0.630**	1.370	0.649	1.334	0.637	1.641	0.679	1.513	0.659	3.912	0.975	3.316	0.900
	1/2	1.611	0.689	**1.134**	0.628	1.342	0.639	1.272	**0.619**	1.575	0.665	1.520	0.677	3.768	0.970	3.270	0.892
	full	1.633	0.683	**1.132**	**0.608**	1.261	0.624	1.215	0.610	1.531	0.663	1.522	0.664	3.970	1.030	3.285	0.897
	Avg	1.608	0.695	**1.179**	0.632	1.321	0.640	1.283	**0.624**	1.532	0.663	1.464	0.656	3.873	0.997	3.303	0.899
PEMS03	1/8	3.846	1.207	0.561	0.572	2.565	1.084	2.490	0.805	0.235	0.345	**0.196**	**0.316**	0.332	0.387	0.956	0.754
	1/4	0.557	0.526	0.398	0.495	0.463	0.493	0.323	0.390	0.496	0.476	0.308	0.403	**0.280**	**0.371**	0.962	0.757
	1/2	0.408	0.446	0.397	0.495	0.358	0.438	0.330	0.394	0.537	0.511	0.507	0.505	**0.260**	**0.354**	0.960	0.757
	full	0.490	0.482	0.400	0.497	1.273	0.736	0.344	0.404	0.761	0.615	0.552	0.527	**0.319**	**0.388**	0.968	0.761
	Avg	1.325	0.666	0.439	0.515	1.165	0.688	0.872	0.498	0.507	0.487	0.391	0.438	**0.298**	**0.375**	0.962	0.757
PEMS04	1/8	2.655	0.823	**0.652**	**0.590**	2.449	1.126	1.913	0.819	2.454	0.860	3.081	0.857	2.197	0.756	1.438	0.965
	1/4	0.265	0.360	0.514	0.518	1.810	0.892	0.468	0.485	0.391	0.430	0.471	0.500	**0.247**	**0.358**	1.458	0.968
	1/2	**0.216**	**0.321**	0.520	0.518	2.466	1.076	0.503	0.500	0.498	0.478	0.533	0.527	0.259	0.368	1.464	0.962
	full	**0.246**	**0.343**	0.521	0.520	2.982	1.213	0.543	0.516	0.516	0.497	0.645	0.577	0.333	0.425	1.460	0.961
	Avg	0.845	**0.462**	**0.552**	0.536	2.426	1.077	0.857	0.580	0.965	0.566	1.183	0.615	0.759	0.477	1.455	0.964
PEMS07	1/8	3.455	0.975	0.699	0.608	2.377	1.246	5.089	1.142	0.227	0.323	**0.157**	**0.258**	0.202	0.304	1.059	0.775
	1/4	**0.213**	0.314	0.578	0.549	0.888	0.718	0.402	0.431	0.299	0.381	0.758	0.569	0.214	**0.311**	1.062	0.777
	1/2	**0.174**	**0.285**	0.593	0.555	0.330	0.406	0.421	0.444	0.344	0.408	1.048	0.630	0.237	0.326	1.067	0.779
	full	0.571	0.505	0.604	0.559	0.283	0.373	0.435	0.452	0.365	0.423	1.277	0.686	**0.210**	**0.318**	1.071	0.781
	Avg	1.103	0.520	0.618	0.568	0.970	0.686	1.587	0.617	0.309	0.384	0.810	0.536	**0.216**	**0.315**	1.065	0.778
PEMS08	1/8	4.280	1.322	**0.542**	**0.550**	2.511	1.250	2.783	0.879	4.588	1.248	2.392	0.908	1.506	0.691	1.412	0.958
	1/4	1.615	0.821	0.432	0.495	2.140	1.139	0.431	0.441	1.463	0.713	1.046	0.675	**0.384**	**0.427**	1.421	0.964
	1/2	0.778	0.614	0.433	0.497	2.270	1.153	**0.429**	**0.443**	1.007	0.605	1.258	0.753	0.442	0.461	1.426	0.967
	full	0.593	0.548	0.436	0.499	2.616	1.204	**0.407**	**0.434**	1.043	0.630	0.940	0.672	0.420	0.451	1.425	0.965
	Avg	1.817	0.826	**0.461**	0.510	2.384	1.186	1.012	0.549	2.025	0.799	1.409	0.752	0.688	**0.508**	1.421	0.964
METR-LA	1/8	7.216	1.091	**1.013**	**0.674**	1.333	0.746	2.267	0.904	4.886	1.188	4.139	0.914	1.580	0.714	1.357	0.772
	1/4	**0.731**	**0.489**	1.003	0.645	1.261	0.694	1.044	0.615	1.020	0.666	0.780	0.516	0.880	0.546	1.368	0.741
	1/2	**0.733**	**0.487**	1.012	0.647	1.094	0.625	1.069	0.616	0.958	0.630	0.784	0.510	0.903	0.551	1.372	0.743
	full	**0.717**	**0.471**	1.022	0.651	0.971	0.552	1.122	0.601	0.953	0.613	0.766	0.508	0.985	0.570	1.363	0.741
	Avg	2.349	0.635	**1.013**	0.654	1.165	0.654	1.375	0.684	1.954	0.774	1.617	0.612	1.087	**0.595**	1.365	0.749
# 1st		13		**30**		0		11		0		9		13		0	

F Full lookback-axis results

Table 13: Lookback axis, every dataset ($H = 96$, all channels; mean over three seeds). Best in bold. – : not run.

Models		STREAM		DLinear		iTransformer		PatchTST		MambaTS		TimeFilter		TimesNet		TiDE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ERA5-t2m	96	0.136	0.270	0.146	0.275	0.143	0.278	0.147	0.278	0.154	0.283	**0.132**	**0.267**	0.136	0.274	0.150	0.281
	192	**0.137**	**0.276**	0.145	**0.276**	0.156	0.287	0.145	0.277	0.165	0.291	0.154	0.285	0.138	0.280	0.148	0.279
	336	**0.140**	0.274	**0.140**	**0.271**	0.155	0.289	0.142	0.276	0.172	0.298	0.157	0.283	0.141	0.278	0.143	0.275
	672	0.135	0.274	**0.129**	**0.263**	0.141	0.274	0.146	0.283	0.153	0.287	0.152	0.286	0.156	0.296	0.137	0.271
	Avg	**0.137**	0.274	0.140	**0.271**	0.148	0.282	0.145	0.278	0.161	0.290	0.149	0.280	0.143	0.282	0.145	0.277
ERA5-d2m	96	0.193	0.326	0.187	0.322	0.217	0.343	0.189	0.323	0.198	0.327	0.184	**0.319**	**0.183**	**0.319**	0.193	0.325
	192	0.194	0.329	**0.184**	**0.320**	0.211	0.341	0.191	0.327	0.237	0.355	0.195	0.328	0.188	0.326	0.188	0.324
	336	0.188	0.325	**0.180**	**0.315**	0.212	0.343	0.182	0.317	0.222	0.344	0.205	0.329	0.182	0.320	0.184	0.317
	672	0.198	0.332	**0.170**	**0.304**	0.192	0.330	0.194	0.329	0.204	0.334	0.219	0.342	0.193	0.326	0.179	0.313
	Avg	0.193	0.328	**0.180**	**0.315**	0.208	0.339	0.189	0.324	0.215	0.340	0.201	0.329	0.186	0.323	0.186	0.320
ERA5-sp	96	0.637	0.610	**0.590**	**0.587**	0.647	0.625	0.640	0.613	0.659	0.625	0.634	0.610	0.677	0.638	0.637	0.616
	192	0.619	0.609	**0.573**	**0.582**	0.642	0.626	0.639	0.615	0.789	0.686	0.608	0.602	0.654	0.636	0.598	0.600
	336	0.634	0.613	**0.588**	**0.589**	0.700	0.653	0.593	0.599	0.791	0.688	0.683	0.641	0.654	0.633	0.594	0.598
	672	0.622	0.610	**0.576**	**0.582**	0.723	0.665	0.612	0.613	0.751	0.671	0.752	0.674	0.654	0.636	0.596	0.601
	Avg	0.628	0.611	**0.582**	**0.585**	0.678	0.642	0.621	0.610	0.748	0.667	0.669	0.632	0.660	0.636	0.606	0.604
Air Quality	96	**1.326**	**0.635**	3.685	0.948	4.197	1.256	3.745	0.923	4.197	1.008	3.991	0.968	4.215	0.994	3.750	0.918
	192	**1.261**	**0.630**	4.560	1.054	3.910	1.126	4.101	0.983	4.511	1.025	4.075	0.979	5.045	1.075	4.018	0.970
	336	**1.240**	**0.627**	3.783	0.974	3.970	1.124	4.400	1.055	5.050	1.126	4.112	0.979	6.069	1.178	4.273	1.035
	672	**1.298**	**0.648**	3.757	1.002	3.986	1.116	4.362	1.106	4.739	1.094	4.604	1.082	6.507	1.250	4.082	1.052
	Avg	**1.281**	**0.635**	3.946	0.994	4.016	1.156	4.152	1.017	4.624	1.063	4.196	1.002	5.459	1.124	4.031	0.994
PEMS03	96	0.172	0.291	0.406	0.495	0.568	0.592	0.471	0.530	0.242	0.349	**0.151**	**0.276**	0.234	0.338	0.980	0.768
	192	0.133	0.253	0.347	0.445	0.966	0.828	0.227	0.367	0.146	0.269	**0.125**	**0.249**	0.153	0.274	0.388	0.446
	336	0.109	0.225	0.179	0.297	0.683	0.665	0.171	0.313	0.119	0.232	**0.107**	**0.224**	0.144	0.255	0.179	0.281
	672	**0.106**	**0.223**	0.187	0.307	0.666	0.612	0.155	0.292	0.113	0.225	0.115	0.225	0.147	0.258	0.175	0.279
	Avg	0.130	0.248	0.280	0.386	0.721	0.674	0.256	0.376	0.155	0.269	**0.125**	**0.243**	0.170	0.281	0.430	0.444
PEMS04	96	**0.698**	**0.464**	0.987	0.633	1.098	0.598	1.075	0.582	0.988	0.552	0.868	0.531	1.121	0.605	1.265	0.847
	192	1.034	0.572	0.946	0.606	0.969	0.541	0.962	0.533	0.835	**0.479**	0.841	0.505	1.057	0.561	**0.526**	0.500
	336	0.923	0.546	0.920	0.589	0.919	0.515	0.978	0.541	0.800	0.464	0.803	0.487	0.963	0.530	**0.202**	**0.285**
	672	0.581	0.408	0.903	0.584	0.871	0.507	0.948	0.540	0.815	0.483	0.753	0.467	0.953	0.558	**0.202**	**0.287**
	Avg	0.809	0.498	0.939	0.603	0.964	0.540	0.991	0.549	0.859	0.494	0.816	0.498	1.024	0.564	**0.549**	**0.480**
PEMS07	96	**0.127**	**0.232**	0.582	0.541	0.149	0.257	0.605	0.567	0.297	0.373	0.128	0.235	0.172	0.275	1.070	0.776
	192	0.105	0.203	0.470	0.466	0.155	0.254	0.269	0.370	0.168	0.268	**0.089**	**0.186**	0.126	0.224	0.447	0.465
	336	0.090	0.181	0.212	0.300	0.412	0.417	0.168	0.284	0.100	0.193	**0.084**	**0.175**	0.126	0.214	0.210	0.282
	672	0.089	0.182	0.210	0.298	0.694	0.611	0.165	0.278	0.098	0.190	**0.086**	**0.176**	0.143	0.225	0.208	0.284
	Avg	0.103	0.200	0.369	0.401	0.353	0.384	0.301	0.375	0.166	0.256	**0.097**	**0.193**	0.142	0.234	0.484	0.452
PEMS08	96	**1.032**	**0.542**	1.293	0.673	1.344	0.614	1.418	0.629	1.293	0.586	1.286	0.581	1.679	0.741	1.160	0.825
	192	1.305	0.611	1.248	0.643	1.219	0.559	1.233	0.558	1.099	0.505	1.115	0.529	1.296	0.601	**0.464**	**0.488**
	336	1.311	0.630	1.207	0.621	1.149	0.539	1.234	0.573	1.073	0.504	1.058	0.513	1.266	0.582	**0.221**	**0.315**
	672	0.883	0.489	1.152	0.603	1.167	0.579	1.220	0.602	1.086	0.528	1.062	0.529	1.250	0.601	**0.220**	**0.318**
	Avg	1.133	0.568	1.225	0.635	1.220	0.573	1.276	0.591	1.138	0.531	1.130	0.538	1.373	0.631	**0.516**	**0.486**
METR-LA	96	0.897	0.592	0.967	0.656	1.015	0.613	1.104	0.657	0.960	0.606	0.945	0.611	**0.896**	**0.568**	1.287	0.696
	192	0.885	0.596	0.907	0.627	0.886	0.570	1.039	0.648	0.902	0.583	0.846	0.569	**0.838**	**0.550**	1.091	0.641
	336	0.879	0.584	0.861	0.605	0.889	0.586	0.904	0.595	0.878	0.573	0.828	0.564	0.837	0.555	**0.754**	**0.531**
	672	0.956	0.633	0.855	0.603	0.946	0.640	0.883	0.592	0.828	0.552	0.801	0.560	0.936	0.611	**0.743**	**0.532**
	Avg	0.904	0.601	0.898	0.623	0.934	0.602	0.982	0.623	0.892	0.579	**0.855**	0.576	0.877	**0.571**	0.969	0.600
# 1st		**19**		**19**		0		0		1		15		6		15	

G Efficiency

We measure the latency of a rollout to $H = 720$ for every model on one GPU (batch 32, 128 channels, fp32), using the same rollout code as the evaluation and CUDA graphs where the model allows them. Baselines re-encode the lookback window at every step; STREAM encodes it once and then advances its cached recurrent state.

State cache. On the same network, caching the state makes fine-grained rollout 9–25× faster than re-encoding the window at every step, and the gap widens with the lookback (Figure 8a).

Fine-grained rollout at the cost of block-wise rollout. Because a cached step costs the same whether it emits one patch or a block, fine-grained rollout of STREAM takes only 1.2× the time of block-wise rollout, whereas for every baseline it takes 5.8–15× as long (Figure 8b). The accuracy gains of fine-grained rollout (Section 5.4) therefore come almost for free.

Scaling with the lookback. When the lookback grows from 96 to 1440 steps, the rollout latency of STREAM grows 2.9×, the least among models that keep the temporal resolution of the lookback (Figure 8c); PatchTST grows 46× and MambaTS 17×. Variate-token models (iTransformer and TimeFilter with one patch per window) grow even less because they compress the whole lookback into one token per channel, which also prevents them from refining a forecast patch by patch.

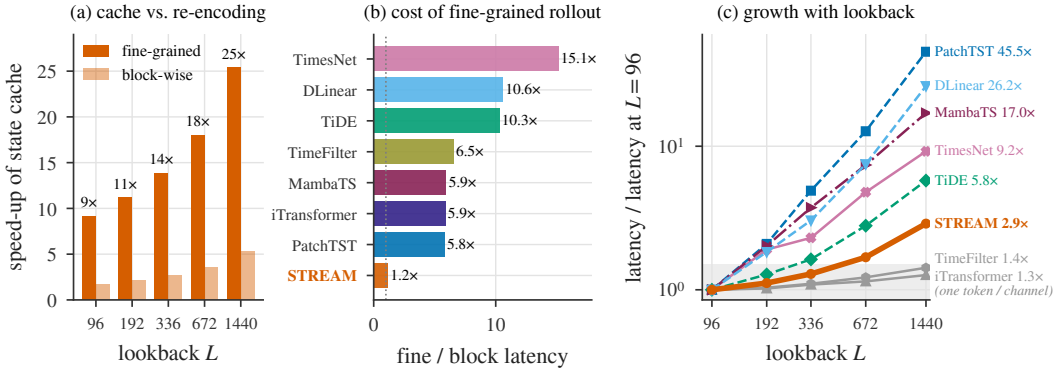


Figure 8: Rollout efficiency ($H = 720$, batch 32). (a) Speed-up from the state cache on the same network. (b) Latency of fine-grained relative to block-wise rollout at $L = 672$. (c) Fine-grained rollout latency relative to $L = 96$ (log scale); grey: models that compress the lookback into one token per channel.