WHICH CHANNELS, OR HOW MANY? MATCHED CONTROLS FOR DATA-DERIVED ATTENTION MASKS IN MULTIVARIATE FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Forecasters that mix channels with attention are often given a fixed relational mask estimated from data, and the benefit is usually reported against the unmasked model. Such comparisons change more than the graph: implementations, training runs and machines differ, and a mask also makes attention sparse, so a gain does not show that the estimated edges matter. We study this question for three masks estimated from road-sensor series (partial correlation, Pearson correlation and lead–lag communities, assigned to separate heads of an inverted Transformer), using two controls trained in the same launch as the masked model: an all-ones mask that leaves every parameter and setting unchanged, and a rewiring of each relation that keeps every sensor's degree. We find that the masks lower test error against the all-ones control in all 22 dataset–horizon cells on six road networks, by 13% on average, and that the gain grows with the forecast horizon on every network; on three non-road series they do not help. We find that degree-preserving rewiring keeps only the smaller part of the gain, so most of it requires the estimated edges rather than sparsity alone. We also find that pairing runs from different launches moves the estimated effect by several points, more than the spread between seeds, which is why the controls must be trained together. We release the masks of the rewiring study, estimated without the test split, with their builder, a per-run configuration table that names each run's mask file, and the analysis code.

1 INTRODUCTION

Inverted Transformers treat every channel of a multivariate series as one token and let attention decide which channels exchange information (Liu et al., 2024). On sensor networks it is natural to restrict that choice with a graph. Road-traffic forecasters have long propagated information along a supplied or learned adjacency (Li et al., 2018; Wu et al., 2019; 2020; Bai et al., 2020; Shang et al., 2021), and attention models add fixed graph masks to their heads (Jiang et al., 2023). The usual evidence for such a prior is a lower error than the same family without it. Two facts make that evidence weaker than it looks. First, a graph-free model with node identities matches graph models on the same traffic benchmarks (Shao et al., 2022), so the reference point matters. Second, a mask changes two things at once: it names *which* channels relate, and it makes every head sparser, which regularises by itself.

This work investigates how a fixed relational mask should be tested, using data-derived masks on road-sensor forecasting as the instance. The masks are three graphs over the sensors, partial correlation, Pearson correlation and lead–lag communities; the released build estimates them on the first 70% of each series, which excludes the test split. Each is given to separate heads of an inverted Transformer as an additive gate before the softmax (Section 3). We compare the masked model (FULL) with two controls trained in the same launch, where a launch is one invocation of a script that trains every arm of a comparison on one machine. The first control is an all-ones mask (ONES), which uses the same code path and parameters with dense heads. The second is a degree-preserving rewiring of each relation (RW), which keeps every sensor's number of neighbours and changes only which sensors they are. The command lines of the arms differ only in the mask file, and we reconstruct every run's configuration from the launch scripts and logs.

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.

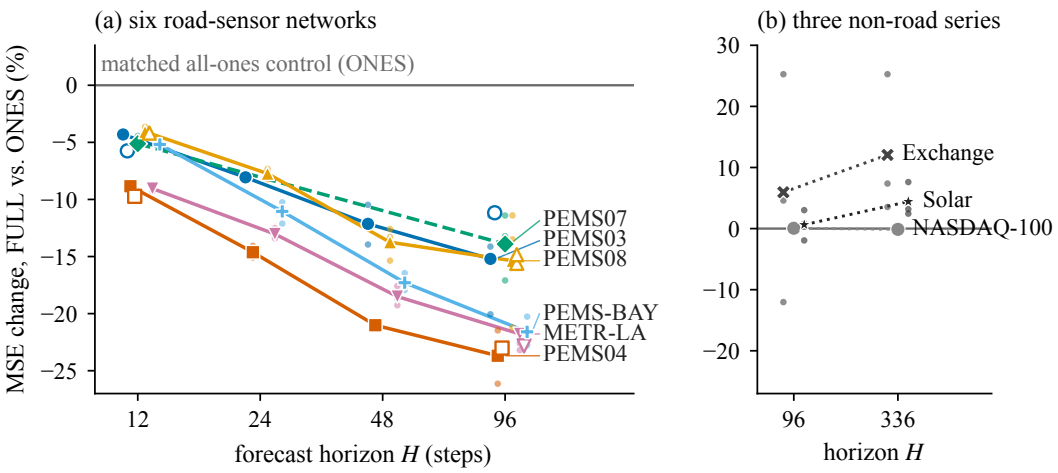

Figure 1: Masked model (FULL) against the all-ones control (ONES) trained in the same launch. Each large marker is the change in mean test MSE over three seeds; small dots are the three paired per-seed differences, in % of the ONES mean. (a) Road-sensor networks; solid lines are the first paired launch (with its completion launch for PEMS-BAY), the dashed line (PEMS07) the rewiring launch, hollow markers other launches of the same cell. (b) Non-road series under the same protocol.

Three findings follow, each a statement about these masks under this protocol. *The masks help road-sensor forecasting against a matched control.* In all 22 dataset–horizon cells on 6 road networks the masked model has lower test error than the all-ones control, by 12.9% on average (range 4.1–23.4%), and every per-seed difference has the same sign (Figure 1a). The reduction grows with the horizon on every network. On exchange rates, stock indices and solar power the same construction does not help (Figure 1b). *Most of the gain needs the estimated edges.* Rewired masks with the same degrees recover 1.0–6.0% of the ONES error, while the true edges add a further 2.2–17.7%, larger than the sparsity part in all 8 cells tested (Section 4.2). *The pairing unit matters.* The same arm trained in two launches can differ by several times its seed spread, and pairing a masked run with a control from another launch moves the estimated effect by up to 4.6 points (Section 4.4). A collection of 208 model–cell results (three seeds each) for 13 models, gathered across separate launches, therefore supports descriptive rankings but not the mask effect, and none of its 16 primary host–mask pairs can be re-specified from released scripts.

This work does not propose a new forecasting architecture, and it does not claim that graph priors are new: fixed graph masks in separate heads and a dense-attention ablation already appear in PDFormer (Jiang et al., 2023). Nor does it rank forecasting models. Its contribution is a protocol that separates the effect of a fixed relational support from implementation, launch and sparsity, applied with paired statistics to released masks, and a measurement of what that protocol finds. The masks, their builder, the per-run configuration table, all logs and the analysis code are in the supplementary material.

2 Related Work

Channel dependence in forecasting. Channel-independent linear and patch models are strong baselines (Zeng et al., 2023; Nie et al., 2023; Han et al., 2024), while other models mix channels explicitly (Zhang & Yan, 2023; Wu et al., 2023; Liu et al., 2024; Wang et al., 2025). Structure has been supplied as leading indicators (Zhao & Shen, 2024), channel clusters (Chen et al., 2024), adaptive embeddings (Liu et al., 2023) or learned graphs (Wu et al., 2020; Bai et al., 2020; Shang et al., 2021). The masks studied here are the simplest member of this family: fixed, estimated once from training data, and entering only through the attention scores.

Graph masks in attention. PDFormer (Jiang et al., 2023) is the closest design. It gives a hop-threshold geographic mask and a DTW-similarity semantic mask to separate heads of spatial self-attention and reports a large gain over a variant in which every node attends to all nodes. That variant is an all-ones control. The present study differs in the relations (conditional, marginal and

delayed statistics of the training series), in the host (channel tokens rather than per-time-step nodes), and above all in the controls: the dense arm is trained in the same launch with the same code, and a degree-preserving rewiring separates edge identity from sparsity.

Controls for graph structure. Degree-preserving edge swaps are the standard null model of network science (Maslov & Sneppen, 2002). In graph learning, rewiring is usually studied as an intervention (Micheli & Tortorella, 2025); as a control, a recent traffic model replaces its learned prior with a random graph of equal edge density (He et al., 2026), and frozen-graph controls separate graph and training effects in graph structure learning (Kim & Jang, 2026). Graph-free baselines (Shao et al., 2022) and local effects (Cini et al., 2023) show how much of a spatio-temporal model's accuracy does not come from the graph. We adopt the degree-preserving null because it holds each sensor's attention budget fixed, which a density-matched graph does not.

Evaluation practice. Uniform re-evaluation changes conclusions about graph models (Shchur et al., 2018; Errica et al., 2020), and run-to-run variance from initialisation, data order and hardware can exceed reported differences (Bouthillier et al., 2021). We add one practical point for forecasting studies that collect results over many machines: the unit of pairing should be the launch, and the configuration of each run should be recoverable from what was executed.

3 SETTING: MASKS, HOSTS AND CONTROLS

3.1 WHERE THE MASK ENTERS

Given a history $\boldsymbol{X} \in \mathbb{R}^{T\times N}$ the task is to predict the next H steps of all N channels. The inverted Transformer normalises each channel, embeds its T values as one d-dimensional token, and applies L encoder layers with h heads over the N tokens (Liu et al., 2024; Vaswani et al., 2017). With a relation r_q assigned to head q, the masked model computes the attention matrix and head output

$$\boldsymbol{A}^{(q)} = \operatorname{softmax}\Big(\tfrac{1}{\sqrt{d_k}}\boldsymbol{Q}^{(q)}{\boldsymbol{K}^{(q)}}^{\top} + \boldsymbol{G}^{(r_q)}\Big), \qquad \boldsymbol{O}^{(q)} = \boldsymbol{A}^{(q)}\boldsymbol{V}^{(q)}, \tag{1}$$

where $G^{(r)}_{ij} = 0$ if $M^{(r)}_{ij} = 1$ and $-\infty$ otherwise, and concatenates and projects the head outputs as usual. The masks $\boldsymbol{M}^{(r)} \in \{0,1\}^{N\times N}$ are non-trainable buffers with unit diagonals, so every query keeps a finite score. Head q uses relation $q \bmod 3$; with the $h = 8$ heads of every paired run, partial correlation and Pearson correlation receive three heads each and lead–lag communities two (Figure 2d). The mask is applied in all $L = 3$ layers. The masked attention has the same four $d \times d$ projections as the host, so the masked model has exactly the host's parameters (1,609,740 at $H = 12$ and 1,631,328 at $H = 96$ for $d = 256$, $d_{\text{ff}} = 512$, $T = 96$; Appendix C). It is not the host's code, however: the masked implementation applies no dropout to the attention probabilities, whereas the host does. A comparison with the unmasked host therefore changes two things, and only ONES, which runs the masked implementation with all-ones masks, changes the mask alone.

Two further hosts appear only in the descriptive comparisons of Section 4.5, and we define them here because their names are easily misread. The *recurrent host* adds, before the masked encoder, one residual block that runs a gated recurrent unit (Cho et al., 2014) along the sequence of channel tokens (1,970,176 extra parameters); its masks enter the same masked attention as above, and it is not a selective state-space model (Gu & Dao, 2023). The *two-stage host* replaces the router-based cross-dimension stage of Crossformer (Zhang & Yan, 2023) with masked full attention over channels in all seven two-stage layers, which removes 1,856,000–1,877,504 parameters relative to Crossformer; its all-ones arm is therefore a router-free dense variant, not Crossformer.

3.2 THE THREE RELATIONS

We describe the builder that produced the released masks, which are the ones behind the rewiring study. Let $\boldsymbol{Z} \in \mathbb{R}^{T_{\text{tr}}\times N}$ be the first 70% of the series, standardised per channel. For the PEMS districts this prefix is the training split (60%) plus the first half of the validation split, which is used for early stopping; it never reaches the test split (the last 20%). For the other datasets the prefix equals the training split. With $\boldsymbol{\Theta} = (\widehat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{I})^{-1}$, $\lambda = 10^{-2}$, the partial-correlation score is $s^{\text{PC}}_{ij} = |\theta_{ij}|/\sqrt{|\theta_{ii}\theta_{jj}|}$, and the Pearson score is $s^{\text{PE}}_{ij} = |\rho_{ij}|$. Both are turned into masks by a

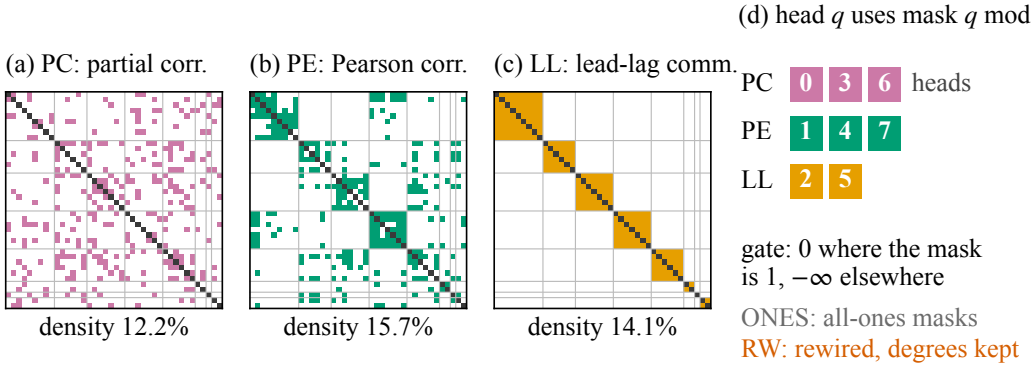

Figure 2: The three released masks on real sensors. (a)–(c) Rows and columns are sensors 0–39 of PEMS08, reordered by lead–lag community (grey lines); dark cells are the diagonal, coloured cells admitted keys. Densities are for all 170 sensors. (d) Head q uses relation $q \bmod 3$, giving 3/3/2 heads to the three relations; the ONES control replaces every mask by all ones, and RW rewires each mask while keeping every sensor's degree.

k-nearest-neighbour rule followed by symmetrisation,

$$M_{ij} = \mathbf{1}\big[\, j \in \mathrm{Top}_k(s,i) \;\vee\; i \in \mathrm{Top}_k(s,j) \;\vee\; i = j \,\big], \qquad k = 16, \tag{2}$$

where $\mathrm{Top}_k(s,i)$ are the k largest off-diagonal entries of row i; every sensor therefore has at least k neighbours. For the lead–lag relation the affinity is the largest lagged cross-covariance of the standardised series over the lags $\Lambda = \{1, 4, 7, 10\}$,

$$a_{ij} = \max_{\tau \in \Lambda} \left| \tfrac{1}{T_{\mathrm{tr}} - \tau} \textstyle\sum_t z_{t+\tau,i}\, z_{t,j} \right|, \tag{3}$$

which is not symmetric; Equation 2 turns a into an unweighted graph $\mathcal{G}$ with total degree $2m$. The partition $\mathcal{P} = \{X_1, \ldots, X_K\}$ minimises two-dimensional structural entropy (Li & Pan, 2016),

$$\mathcal{H}^2(\mathcal{G}; \mathcal{P}) = -\sum_{j=1}^{K} \Big[\sum_{v \in X_j} \tfrac{d_v}{2m} \log_2 \tfrac{d_v}{V_j} + \tfrac{g_j}{2m} \log_2 \tfrac{V_j}{2m} \Big], \tag{4}$$

with d_v the degree of v, V_j the volume of X_j and g_j the number of edges leaving it. The optimiser moves single nodes to the neighbouring community that most decreases $\mathcal{H}^2$, contracts communities and repeats, in the manner of Louvain (Blondel et al., 2008) but on $\mathcal{H}^2$ instead of modularity, and keeps the best of eight restarts with a free number of communities. The builder falls back to greedy modularity (Clauset et al., 2004) only if the optimiser fails or returns at least $0.8N$ communities; the released arrays do not record which branch ran. The lead–lag mask is $M^{\mathrm{LL}}_{ij} = \mathbf{1}[\, i, j \text{ in the same community} \,]$. We verified on the released arrays that every mask is symmetric with a unit diagonal and that M^{LL} is an exact partition, with 8–31 communities. Off-diagonal densities are 2.5–15.7% per relation and the union admits 6.8–29.2% of the pairs on the PEMS districts (Table 1); the partial-correlation and Pearson edge sets overlap little, so the relations are not one sparsity pattern applied three times. The offline cost is $O(|\Lambda|\, T_{\mathrm{tr}} N^2)$ for the lagged covariances, $O(T_{\mathrm{tr}} N^2 + N^3)$ for the covariance and its inverse, and the partition search.

Earlier runs with these hosts used other builds, and we keep them apart. The first paired launch used masks built on its training machine, whose arrays and builder, and hence estimation window, were not retained. The densities logged there for PEMS04 (10.1%, 10.1% and 25.2% including the diagonal) differ from the released build (7.1%, 8.8% and 9.5%), so that launch tests a different mask of the same three relations. The separately collected results of Section 4.5 most likely used an earlier builder (row-wise top-k masks without symmetrisation, default $k = 20$, lags 1–12, Louvain communities, first 60% of each series); neither its arrays nor the k of those runs survive.

3.3 Controls, launches and statistics

Controls. ONES is the all-ones array of the same shape. RW applies $10|E|$ double-edge swaps (Maslov & Sneppen, 2002) to each relation separately, with three independent draws per dataset;

Table 1: Released masks (the build used by the rewiring launch). Densities exclude the diagonal; the union admits a pair if any relation does; last column: Jaccard index of the PC and PE edge sets.

		Off-diagonal density (%)			Union	Communities	Jaccard
Network	N	PC	PE	LL	(%)	(LL)	PC–PE
PEMS03	358	6.0	6.7	6.4	13.0	18	0.19
PEMS04	307	6.8	8.5	9.3	17.1	11	0.18
PEMS07	883	2.5	2.8	4.5	6.8	31	0.21
PEMS08	170	12.2	15.7	14.1	29.2	8	0.20
METR-LA	207	9.8	11.3	11.1	21.4	11	0.18

Table 2: What each comparison protocol holds fixed. Only the last two rows isolate the mask; the paper's findings rest on them.

Comparison	same code	same launch	mask only	config. known	masks released
masked model vs. unmasked host	✗	✗	✗	✗	✗
FULL vs. ONES, different launches	✓	✗	✗	✗	✗
FULL vs. ONES, same launch	✓	✓	✓	✓	rewiring launch
FULL vs. RW, same launch	✓	✓	✓	✓	✓

every draw preserves each sensor's degree in every relation, symmetry and the diagonal (re-verified on the released arrays). A rewired mask keeps 3.7–25.0% of the original edges.

Launches. Every run uses a 96-step input, three encoder layers, $d = 256$, $d_{\text{ff}} = 512$, eight heads, learning rate 5×10^{-4}, ten epochs with patience three and three repetitions, on the standard benchmark pipeline (Wang et al., 2024). The batch size is 16, except 8 for PEMS-BAY and at $H = 96$ in the first launch, for PEMS07 in the rewiring launch, and at $H = 96$ in the completion launch. Test MSE is computed on the standardised scale. Four launches contain FULL/ONES pairs. The first launch covers PEMS03, 04, 08, METR-LA and PEMS-BAY at four horizons, with a completion launch for three PEMS-BAY cells; the rewiring launch covers PEMS03, 04, 07 and 08 at $H \in \{12, 96\}$ with three RW draws; a third launch repeats two long-horizon cells; and a fourth covers three non-road series at $H \in \{96, 336\}$. The per-run configuration table lists 306 seed-level runs; all 39 FULL/ONES pairs have identical command lines apart from the mask file. The scripts record only a mask path. The rewiring launch read that path at its start to write its rewired draws; all 12 draws regenerate byte-exactly from the released arrays, and none can come from the superseded arrays in the archive, whose lead–lag channel has no off-diagonal edge, so its masked arm used the released masks. The other launches used arrays that were not retained; the lead–lag densities logged on the first launch's machine (PEMS04 25.2%, METR-LA 11.7%, PEMS-BAY 10.1%, with the diagonal; PEMS03 and PEMS08 not logged) are those of communities, not of an identity ($\leq$0.5%).

Statistics. The three repetitions of a launch are paired by position. Per cell we report the mean paired difference ONES − FULL with a t interval and the change relative to the ONES mean, whose denominator is far from zero. Across cells we use a sign test and an exact Wilcoxon signed-rank test, averaging replicate launches of one cell first, and a two-level bootstrap over cells and seeds. We report no ratio of two differences as a headline. The protocol, in the form we recommend for any fixed relational prior, is summarised below.

Testing a fixed relational support. (i) Train the dense control through the same code as the masked model, changing only the mask array. (ii) Train all arms in one launch and pair repetitions by position; do not pair runs across launches. (iii) Add a degree-preserving rewiring of each relation, with several draws, to separate edge identity from sparsity. (iv) Report per-seed paired differences and test across cells rather than comparing means. (v) Record each run's configuration from the executed command, and release the masks with checksums.

Table 3: FULL against ONES trained in the same launch: change in mean test MSE in % of the ONES mean, with the 95% t interval of the paired difference expressed in the same unit (three seeds per arm; †two ONES seeds). Negative is better. Non-road series use $H \in \{96, 336\}$. Per-seed values and absolute MSE are in Appendix A; – marks cells not run in that launch.

Network	$H = 12$	$H = 24$	$H = 48$	$H = 96$
First launch (masks built on the training machine)				
PEMS03	-4.3 [$-5.2, -3.4$]	-8.1 [$-8.6, -7.5$]	-12.1 [$-16.5, -7.8$]	-15.2 [$-26.2, -4.2$]
PEMS04	-8.8 [$-9.8, -7.8$]	-14.6 [$-15.9, -13.3$]	-21.0 [$-21.5, -20.5$]	-23.7 [$-29.5, -17.9$]
PEMS08	-4.1 [$-5.4, -2.8$]	-7.7 [$-8.9, -6.6$]	-13.7 [$-17.3, -10.2$]	-15.4 [$-28.2, -2.5$]
METR-LA	-9.0 [$-9.6, -8.5$]	-13.0 [$-14.2, -11.9$]	-18.5 [$-20.6, -16.4$]	-21.8 [$-26.7, -19.1$]†
PEMS-BAY	-5.2 [$-5.8, -4.6$]	-11.0 [$-13.4, -8.7$]	-17.3 [$-19.2, -15.4$]	-21.6 [$-24.7, -18.5$]
Rewiring launch (released masks)				
PEMS03	-5.7 [$-7.1, -4.4$]	–	–	-11.2 [$-13.0, -9.3$]
PEMS04	-9.7 [$-11.0, -8.5$]	–	–	-23.0 [$-26.1, -19.9$]
PEMS07	-5.1 [$-6.6, -3.7$]	–	–	-13.9 [$-21.1, -6.7$]
PEMS08	-4.2 [$-4.9, -3.5$]	–	–	-15.6 [$-16.2, -15.0$]
Third launch (masks built on its training machine)				
PEMS08	–	–	–	-14.8 [$-22.2, -7.5$]
METR-LA	–	–	–	-22.8 [$-26.5, -19.1$]
Non-road series	$H = 96$	$H = 336$		
Exchange	5.9 [$-40.5, 52.4$]	12.1 [$-16.8, 40.9$]		
NASDAQ-100	0.1 [$-1.1, 1.2$]	-0.1 [$-0.6, 0.4$]		
Solar	0.6 [$-5.5, 6.8$]	4.4 [$-2.5, 11.4$]		

4 RESULTS

4.1 AGAINST A MATCHED CONTROL, THE MASKS HELP ON EVERY ROAD NETWORK

Table 3 gives every paired cell. On the road networks the masked model is better in all 30 launch-level pairs covering 22 cells, all three per-seed differences agree in sign in each pair, and each pair's 95% interval excludes zero. With one-sided paired tests and Holm's correction, 29 of the 29 pairs with three repetitions remain significant at 0.05 (the METR-LA $H = 96$ pair of the first launch has two ONES seeds and is not tested). Across the 22 cells the sign test and the Wilcoxon test both give $p = 4.8 \times 10^{-7}$. The mean reduction is 12.9% (median 13.1%); at the pair level it is 13.1% with a bootstrap 95% interval of [10.9, 15.2]%.

The reduction grows with the horizon, from 6.2% at $H = 12$ to 18.1% at $H = 96$ on average, and it grows monotonically on each of the 5 networks measured at all four horizons (from 4.1–9.0% at $H = 12$ to 15.2–23.7% at $H = 96$). Two of the six networks, METR-LA and PEMS-BAY, come from a different city and collection pipeline than the four PEMS districts (Li et al., 2018), and their reductions at $H = 96$ (21.8% and 21.6%) are as large as on the districts (at most 23.7%). The rewiring launch, with a different mask build and at $H = 96$ a different batch size, gives the same direction in all eight of its cells.

On the three non-road series the same construction does not help: the change against ONES ranges from -0.1% to $+12.1\%$, the masked model is worse in five of six cells, and no interval excludes zero. The masks of these series are weaker tests than the road masks, because the solar mask's lead–lag channel fell back to a copy of its Pearson mask when it was built and the exchange and stock masks come from an unreleased earlier build. We therefore read this panel as the absence of a benefit, not as evidence of harm.

4.2 MOST OF THE GAIN REQUIRES THE ESTIMATED EDGES

A mask that keeps every sensor's degree but scrambles its neighbours isolates the value of the particular edges. In all 8 cells of the rewiring launch the ordering is ONES $>$ RW $>$ FULL in mean test MSE, and every rewired run is worse than the masked run of the same repetition (Figure 3, Table 4). Both parts of the gain are positive in every cell with intervals that exclude zero: sparsity at fixed

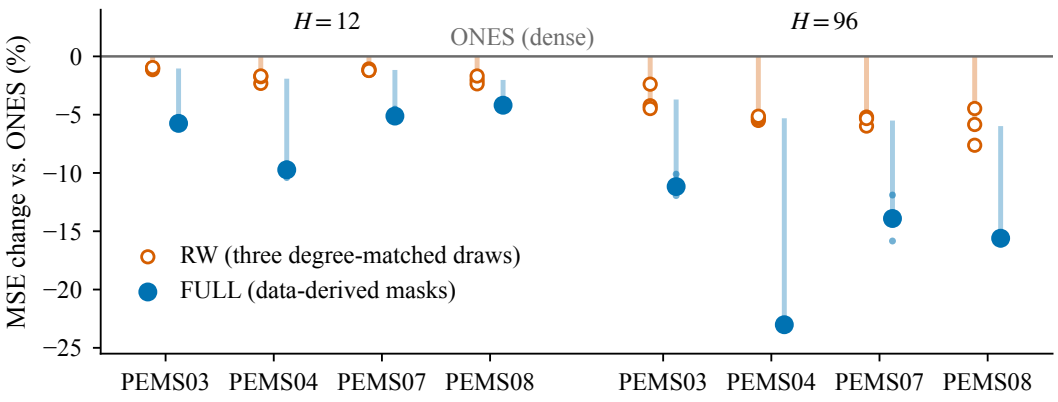

Figure 3: Degree-preserving rewiring in the rewiring launch. Hollow markers: mean test MSE of each of three rewired draws; filled: masked model (small dots are its three seeds); both in % of the ONES mean. The orange bar from zero is the sparsity part (ONES to mean RW), the blue bar the identity part (mean RW to FULL).

Table 4: Decomposition of the gain in the rewiring launch (mean of three seeds; RW averaged over three draws). Reductions are in % of the ONES mean with 95% t intervals over the three paired repetitions; draw spread is the range of the three RW draw means ($\times 10^{-3}$ MSE).

		MSE (mean of 3 seeds)			Reduction, % of ONES (95% CI)		Draw
Network	H	ONES	RW	FULL	sparsity (ONES−RW)	identity (RW−FULL)	spread
PEMS03	12	0.069	0.068	0.065	1.0 [0.3, 1.8]	4.7 [4.1, 5.3]	0.1
PEMS03	96	0.270	0.260	0.240	3.7 [0.9, 6.5]	7.5 [5.0, 9.9]	5.6
PEMS04	12	0.089	0.088	0.081	1.9 [1.5, 2.4]	7.8 [6.4, 9.2]	0.6
PEMS04	96	0.318	0.302	0.245	5.3 [2.4, 8.2]	17.7 [16.6, 18.9]	1.1
PEMS07	12	0.066	0.065	0.063	1.2 [0.7, 1.7]	4.0 [2.0, 5.9]	0.1
PEMS07	96	0.226	0.213	0.194	5.5 [2.6, 8.4]	8.4 [2.9, 13.9]	1.7
PEMS08	12	0.081	0.079	0.078	2.0 [0.9, 3.1]	2.2 [1.5, 2.8]	0.5
PEMS08	96	0.390	0.366	0.329	6.0 [0.8, 11.1]	9.6 [4.6, 14.6]	12.3

degree removes 1.0–6.0% of the ONES error (mean 3.3%), and the estimated edges remove a further 2.2–17.7% (mean 7.7%). The identity part is the larger in all 8 cells and exceeds the spread between the three rewired draws in all 8 (sign test over cells, $p = 0.008$). It is largest on PEMS04 at $H = 96$ and smallest on PEMS08 at $H = 12$, where the two parts are nearly equal. The launch script also states an acceptance criterion in its header. Edge identity counts if FULL beats every rewired draw in at least six of the eight cells and the mean identity gain exceeds the mean spread between draws; the sparsity-only reading holds if RW − FULL is within the draw spread in at least five cells. Applied to the logged results, the first condition holds in 8 of 8 cells (mean identity gain 7.7% against a mean draw spread of 1.0%) and the second in 0 of 8.

4.3 One query sensor under the three arms

Figure 4 shows what the arms change for one sensor. We take PEMS08 and the sensor whose union neighbourhood has the median size across sensors (lowest index among ties; sensor 6, 49 neighbours). Under ONES it may attend to all 170 sensors. Under FULL its partial-correlation and Pearson heads admit two partly overlapping sets scattered across communities, and its lead–lag heads admit exactly its own community. Under RW each head type keeps the same number of keys but only 3 of 17, 5 of 22 and 3 of 21 of the original ones. Panel (b) gives the outcome for this network at $H = 96$: the three rewired draws fall between the dense and the masked arms, as in every cell of Table 4. Per-run forecasts were deleted after logging, so the figure shows the supports each arm gives this sensor, not its forecasts; a sensor-level error analysis is left for future runs.

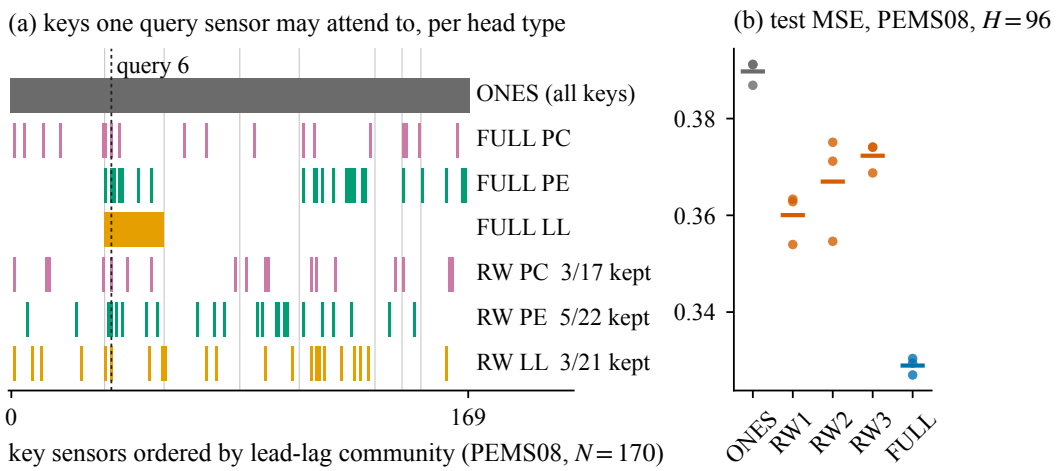

Figure 4: A case study under an identical view. (a) Keys that PEMS08 sensor 6 (median union-neighbourhood size; selection rule stated in the text) may attend to in each head type, for ONES, FULL and the first rewired draw; sensors are ordered by lead–lag community, and the counts give how many of the original keys a rewired head keeps. (b) Test MSE of the five arms on PEMS08 at $H = 96$ in the rewiring launch (dots: seeds; bars: means). No per-sensor forecasts were retained.

4.4 Why the controls must share a launch

Seven cells were trained in more than one launch (Figure 5a). Between launches the same arm moves by 0.4 to 4.1 times its largest within-launch seed deviation, because the launches differ in machine and, depending on the cell, in the build of the masks or the batch size. Yet the relative gain measured inside each launch is stable: for the same cell the within-launch gains differ by a median of 0.9 and at most 4.0 points. Pairing a masked run with the control of another launch deviates from the masked run's own within-launch gain by a median of 1.1 and up to 4.6 points, and by up to 6.1 points when the control comes from the separately collected results used in Section 4.5 (Figure 5b). These shifts are as large as the short-horizon effects themselves (4.1–9.0%); a control that shares the launch removes them by construction, so the paired design, not the number of seeds, is what makes the short-horizon results interpretable.

4.5 What the separately collected results can support

Besides the paired launches, per-seed test errors exist for 13 models on the four PEMS districts at four horizons, gathered across separate launches on several machines; they include the unmasked hosts, seven common baselines and three masked hosts. Their per-seed values regenerate deterministically, but their configurations do not. Of the 208 model–cell entries, 129 carry no configuration record, only 4 can be matched to a raw log line and none to a released launch script; none of the 16 primary pairs of masked model and unmasked host is fully specified. We therefore use these results descriptively and never pool them with the paired launches (Appendix E). Within them, the masked model has lower error than the unmasked host in 16 of 16 cells (3.8–21.5%), a comparison that also changes the attention dropout and the launch. Its FULL runs are lower than ONES runs from other launches in 16/16 cells (mean 9.7%), as are the recurrent host's (16/16, mean 13.8%); for the two-stage host FULL is lower in only 5/16 cells (mean change -4.9%), and its FULL runs used batch sizes from 4 to 32. These directions agree with the paired results where both exist, but Section 4.4 shows why their magnitudes are not estimates of the mask effect, and the two-stage result in particular cannot be read as a property of that architecture.

5 Discussion and limitations

The paired design answers a narrower question than the usual comparison, and answers it more firmly: in these road networks, restricting each head of a channel-token Transformer to a data-

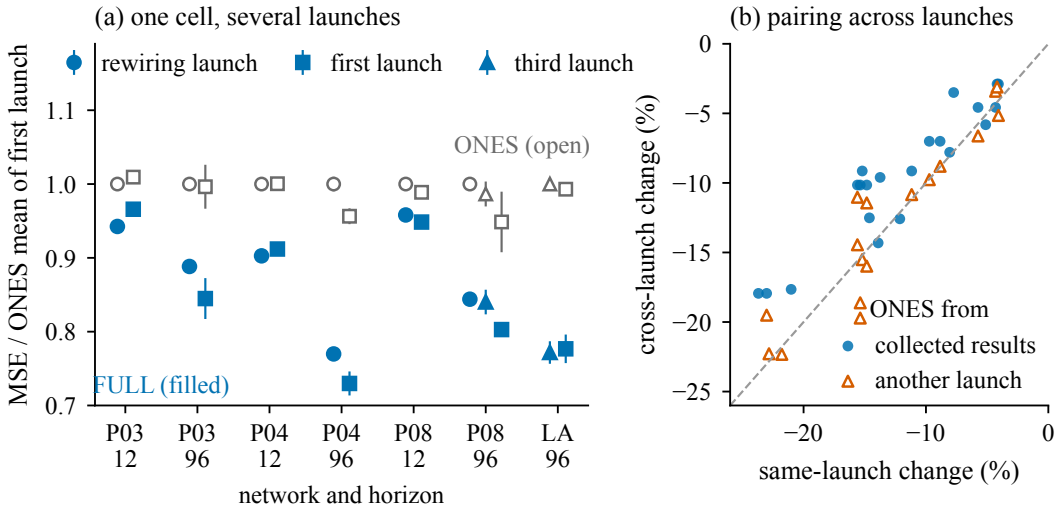

Figure 5: Launch effects. (a) Cells trained in two or three launches: mean test MSE of FULL (filled) and ONES (open) per launch with seed standard deviations, normalised by the first launch's ONES mean. (b) Change of FULL against a control from its own launch (x) and from another launch (y); points off the diagonal are the error that cross-launch pairing adds.

derived neighbourhood lowers error at every horizon, and most of the reduction comes from which sensors are admitted. The growth of the gain with horizon suggests that cross-sensor structure matters more once each sensor's own recent past stops being informative, but we have not tested this reading, for instance by stratifying sensors by lag or distance.

Several controls were not run and remain open. The design assigns relations to separate heads, but no run applies the union of the three masks to every head, so the evidence supports structured sparsity with the estimated edges, not the claim that separate heads beat a shared support. With eight heads the allocation is 3/3/2, and no balanced or rotated allocation was tested. Single-relation arms exist only in the separately collected results. Every cell has three seeds, and effects are claimed only where all per-seed differences agree. All six benefiting networks are road-sensor networks; the non-road panel uses weaker masks. The support is fixed after estimation, so a relation that changes after the estimation window is not tracked. For the PEMS districts the estimation window overlaps the first half of the validation split, which selects the stopping epoch; the test split is never used. Neither the masks of the first launch nor their builder were retained, so we cannot show that they excluded the test split; the rewiring launch, whose masks exclude it, gives gains within 4.0 points of the first launch's in the six cells both cover. Errors are reported on the standardised scale because de-normalised errors were not logged.

6 Conclusion

In this work, we tested data-derived relational masks in an inverted Transformer against two controls trained in the same launch: an all-ones mask that changes nothing else, and a degree-preserving rewiring that changes only which sensors are neighbours. The masks lower test error on every road-network cell measured and the gain grows with horizon; rewiring shows that most of the gain needs the estimated edges; on non-road series there is no gain. Controls from other launches shift the estimate by amounts comparable to the short-horizon effect, so the pairing unit, and a record of what each run executed, matter as much as the number of seeds. We hope the protocol, the released masks and the per-run configuration table make the next claim about a relational prior easier to test, starting with the union-mask and balanced-head controls that this study leaves open.

REPRODUCIBILITY STATEMENT

The supplementary material contains the model code of the three hosts and the patches that apply the masks, the mask builder, every launch script and raw log behind the paired launches, the released mask arrays (the relation masks and all rewired draws), the per-run configuration table reconstructed from the scripts and logs together with its coverage report, the collection of per-seed results used in Section 4.5, and the scripts that regenerate every number, table and figure from those files. Section 3.3 and Appendix A state which runs enter each cell. The datasets are public and not included. The pipeline was not run in a bit-reproducible mode, and per-run forecasts and checkpoints were not retained, so re-running a configuration reproduces its distribution, not its exact errors.

AI USE STATEMENT

We used generative AI tools on the following tasks from ICLR's required-disclosure list: implementing methods (parts of the model, mask-building, launch and analysis code), giving feedback on experimental design, and helping to interpret results. We did *not* use generative AI tools to generate data reported as measurements, to clean or reformat data sets, for translation, or for qualitative data analysis. From the recommended-disclosure list we additionally used generative AI tools to draft and edit parts of this paper, to create figures and tables, to identify and summarise related literature, to format references, to reconstruct the per-run configuration table from launch scripts and logs, and to audit the reported numbers against the run records.

We have reviewed all AI-assisted work. Every numerical value reported in this paper is regenerated from the released run records by the released analysis scripts and compared against the text, and every reference was checked against a bibliographic record. We take responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, 2020.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, pp. P10008, 2008.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, 2021.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *Advances in Neural Information Processing Systems*, 2024.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. Taming local effects in graph-based spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, 2023.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *International Conference on Learning Representations*, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Xuanmian He, Can Li, and Wanjing Ma. Structure-guided spatiotemporal attention graph neural network for traffic flow prediction. *arXiv preprint arXiv:2608.14177*, 2026.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *AAAI Conference on Artificial Intelligence*, 2023.

Minkyoung Kim and Beakcheol Jang. Bilevel graph structure learning, revisited: Inner-channel origins of the reported gain. *arXiv preprint arXiv:2605.07577*, 2026.

Angsheng Li and Yicheng Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339, 2016. doi: 10.1109/TIT.2016.2555904.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. STAEformer: Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting. In *ACM International Conference on Information and Knowledge Management*, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002. doi: 10.1126/science.1065103.

Alessio Micheli and Domenico Tortorella. An empirical evaluation of rewiring approaches in graph neural networks. *Pattern Recognition Letters*, 196:134–141, 2025. doi: 10.1016/j.patrec.2025.05.021.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *ACM International Conference on Information and Knowledge Management*, 2022.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is Mamba effective for time series forecasting? *Neurocomputing*, 2025.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence*, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2023.

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.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *International Conference on Learning Representations*, 2024.

A PER-SEED MATCHED RESULTS

Table 5 lists every FULL/ONES pair with absolute test MSE (mean $\pm$ standard deviation over seeds), the mean paired difference with its 95% t interval, the relative change, and whether all per-seed differences favour FULL. Superscripts give the numbers of FULL and ONES seeds when they are not three. The per-seed values themselves are in the configuration table of the supplementary material. The PEMS-BAY cells at $H \in \{24, 48, 96\}$ of the first launch had one or two seeds per arm and are superseded by the completion launch; they agree in sign (second-to-last block of Table 5). The first launch has two ONES seeds at METR-LA, $H = 96$; the third launch repeats that cell with three seeds per arm and gives a change of -22.8%.

Table 5: All paired cells. MSE on the standardised scale; differences in 10^{-3} MSE.

Network	H	ONES MSE	FULL MSE	diff. (10^{-3})	95% CI	Δ (%)	3/3
First launch, masks built on its training machine (PEMS-BAY $H \geq 24$: completion launch)							
PEMS03	12	0.070±.000	0.067±.000	3.0	$[2.4, 3.6]$	-4.3	✓
	24	0.101±.001	0.093±.001	8.1	$[7.6, 8.7]$	-8.1	✓
	48	0.166±.001	0.146±.002	20.2	$[13.0, 27.4]$	-12.1	✓
	96	0.269±.008	0.228±.007	40.9	$[11.4, 70.4]$	-15.2	✓
PEMS04	12	0.089±.000	0.081±.000	7.9	$[7.0, 8.8]$	-8.8	✓
	24	0.126±.000	0.108±.001	18.5	$[16.8, 20.1]$	-14.6	✓
	48	0.203±.001	0.160±.001	42.6	$[41.6, 43.5]$	-21.0	✓
	96	0.305±.003	0.232±.005	72.1	$[54.4, 89.9]$	-23.7	✓
PEMS08	12	0.080±.000	0.077±.000	3.3	$[2.2, 4.3]$	-4.1	✓
	24	0.119±.001	0.109±.000	9.2	$[7.8, 10.5]$	-7.7	✓
	48	0.208±.005	0.179±.003	28.6	$[21.2, 36.0]$	-13.7	✓
	96	0.370±.016	0.313±.003	56.8	$[9.3, 104.4]$	-15.4	✓
METR-LA	12	0.438±.001	0.399±.001	39.7	$[37.4, 41.9]$	-9.0	✓
	24	0.580±.001	0.505±.002	75.6	$[68.9, 82.4]$	-13.0	✓
	48	0.753±.001	0.614±.006	139.4	$[123.3, 155.4]$	-18.5	✓
	96	0.908±.002	0.711±.018	208.0	$[173.9, 242.2]$	$-21.8^{3/2}$	✓
PEMS-BAY	12	0.297±.001	0.282±.002	15.4	$[13.6, 17.2]$	-5.2	✓
	24	0.447±.000	0.398±.004	49.4	$[38.7, 60.1]$	-11.0	✓
	48	0.645±.003	0.534±.002	111.6	$[99.4, 123.8]$	-17.3	✓
	96	0.820±.006	0.643±.011	177.0	$[151.4, 202.7]$	-21.6	✓
Rewiring launch (released masks)							
PEMS03	12	0.069±.000	0.065±.000	4.0	$[3.0, 4.9]$	-5.7	✓
PEMS03	96	0.270±.002	0.240±.003	30.1	$[25.2, 35.1]$	-11.2	✓
PEMS04	12	0.089±.000	0.081±.001	8.7	$[7.6, 9.8]$	-9.7	✓
PEMS04	96	0.318±.003	0.245±.001	73.3	$[63.4, 83.2]$	-23.0	✓
PEMS07	12	0.066±.000	0.063±.000	3.4	$[2.4, 4.4]$	-5.1	✓
PEMS07	96	0.226±.002	0.194±.004	31.4	$[15.1, 47.7]$	-13.9	✓
PEMS08	12	0.081±.000	0.078±.000	3.4	$[2.8, 3.9]$	-4.2	✓
PEMS08	96	0.390±.002	0.329±.002	60.8	$[58.6, 63.0]$	-15.6	✓
Third launch (masks built on its training machine)							
PEMS08	96	0.384±.007	0.327±.007	57.1	$[28.8, 85.3]$	-14.8	✓
METR-LA	96	0.915±.002	0.706±.014	208.8	$[174.9, 242.7]$	-22.8	✓
First launch, PEMS-BAY partial cells (superseded by the completion launch)							
PEMS-BAY	24	0.448±.001	0.396±.004	51.1	$[18.2, 84.1]$	$-11.4^{2/2}$	✓
PEMS-BAY	48	0.641±.001	0.533±.006	107.4	$[48.4, 166.5]$	$-16.8^{2/2}$	✓
PEMS-BAY	96	0.816±nan	0.650±nan	166.1	n/a	$-20.4^{1/1}$	✓
Non-road series, same protocol							
Exchange	96	0.100±.010	0.106±.009	-6.0	$[-52.6, 40.6]$	$+5.9$	
Exchange	336	0.336±.007	0.377±.032	-40.6	$[-137.5, 56.4]$	$+12.1$	
NASDAQ-100	96	0.027±.000	0.027±.000	-0.0	$[-0.3, 0.3]$	$+0.1$	
NASDAQ-100	336	0.102±.000	0.102±.000	0.1	$[-0.4, 0.6]$	-0.1	
Solar	96	0.215±.008	0.217±.003	-1.4	$[-14.6, 11.9]$	$+0.6$	
Solar	336	0.264±.004	0.276±.004	-11.7	$[-30.0, 6.7]$	$+4.4$	

B MASK STATISTICS

Table 1 in Section 3.2 summarises the released masks. Rewired draws keep each sensor's degree in every relation exactly; they retain 3.7–25.0% of the original edges (per relation and draw). The archive also contains three superseded arrays whose lead–lag channel is the identity (a label-conversion error in an earlier build). They were built for side runs on imputation that are not reported, and the configuration table assigns none of its runs to them; Section 3.3 gives the evidence that the rewiring launch, the only paired launch with released masks, used the corrected arrays.

C PARAMETER COUNTS AND WHERE EACH HOST APPLIES THE MASK

Counts follow from the layer definitions of the released code with $T = 96$, $d = 256$, $d_{\text{ff}} = 512$, $L = 3$, $h = 8$. *Masked model and unmasked host:* input embedding $Td + d$; per layer four $d \times d$ projections with biases, two position-wise convolutions ($dd_{\text{ff}} + d_{\text{ff}}$ and $d_{\text{ff}}d + d$) and two layer norms; a final layer norm; output projection $dH + H$. This gives 1,609,740 parameters at $H = 12$ and 1,631,328 at $H = 96$ for both; the masks add $3N^2$ non-trainable entries. *Recurrent host:* the masked model plus one residual block (input projection $d \times 2d$, a gated recurrent unit of width $2d$, gate projection $d \times 2d$, output projection $2d \times d$, layer norm), which adds 1,970,176 parameters; the masks enter the masked encoder, exactly as in Equation 1. *Two-stage host:* in each of the three encoder and four decoder two-stage layers, the cross-dimension stage of Crossformer (two attention layers and a router of $3 \times d$ vectors per segment) is replaced by one masked attention over channels; with three routers per segment this removes 1,856,000 ($H = 12$) to 1,877,504 ($H = 96$) parameters.

D COVERAGE OF THE CONFIGURATION TABLE

The table has one row per seed-level run and records the launch, the log line, dataset, loader, horizon, arm, repetition, test MSE, the mask path written in the script, the archive file and checksum of the array it loaded where released (keyed by its full path in the archive, since file names repeat across folders) with the evidence for that identification, and every value on the command line (resolved from the script's own lookup tables and asserted against the script text). Hyperparameters not set on the command line (for example dropout 0.1) are marked as defaults. Across the 306 runs, every FULL/ONES pair has identical command lines apart from the mask file. For the separately collected results: 0 of 208 entries are fully specified, 4 match a raw log line without a released script, 204 exist only as recorded per-seed values, and 79 carry a free-text configuration record; 9 of the 16 primary pairs carry records for both arms and 0 are fully specified.

E SEPARATELY COLLECTED RESULTS

Table 6: Changes (%) of FULL in the separately collected results, per network and horizon. Row "ONES": against an all-ones run of the same host from another launch. Row "ref.": against the reference model of that row, also from another launch; for the attention host this is the unmasked inverted Transformer (which also differs in attention dropout), for the recurrent host it is S-Mamba (Wang et al., 2025) and for the two-stage host Crossformer, neither of which is that host without masks. Descriptive only (Section 4.5).

		PEMS03				PEMS04				PEMS07				PEMS08			
Host	vs.	12	24	48	96	12	24	48	96	12	24	48	96	12	24	48	96
attention	ONES	−5	−8	−13	−9	−7	−13	−18	−18	−6	−8	−12	−14	−3	−4	−10	−10
	ref.	−4	−7	−11	−9	−9	−15	−21	−22	−6	−8	−12	−15	−4	−6	−13	−14
recurrent	ONES	−4	−8	−15	−20	−9	−14	−21	−27	−5	−8	−16	−21	−5	−10	−19	−19
	ref.	−3	−6	−9	−12	−4	−7	−11	−12	−3	−5	−9	−4	0	+2	0	+12
two-stage	ONES	+5	+3	+4	+2	+1	+2	+1	−2	+22	+22	+33	+4	−7	−2	−6	−3
	ref.	+8	−23	−49	−29	+1	0	−42	−17	+6	−34	−36	−48	−3	−2	−2	0

Unlike the paired launches, these results do not show a gain that grows with horizon on every network: against the unmasked host, the gain on PEMS03 is 11% at $H = 48$ and 9% at $H = 96$. The monotone growth reported in Section 4.1 is a property of the paired launches.