CAUSAL IN TIME, CIRCULAR IN SPACE FOR HIGH-DIMENSIONAL TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

A multivariate time series records many variables over time, and a forecaster must learn relationships along both axes. Modern sensor networks, power grids and web traffic can contain thousands of variables, making attention across variables slow because its cost grows quadratically. A fast global convolution (Li et al., 2021; Poli et al., 2023) is a cheaper alternative, but it naturally wraps the end of an axis back to its beginning. Along time this hands a position trained to forecast what follows it part of its own target, but it is natural for a variable axis with no intrinsic first or last variable. We introduce Time Fluxer (TiF), which replaces attention with a fast, input-dependent global convolution, the fluxer, and gives the two axes different closures, causal along time and circular across variables. The causal closure lets every position be trained to forecast what follows it, so one model serves any history within its window; the circular closure gives no variable an end. On seven TIME-HD datasets (Ni et al., 2025), replacing variable-wise attention (Liu et al., 2024a) with the fluxer reduces error in 26 of 28 settings, and replacing patch-wise attention (Nie et al., 2023) reduces error in 21 of 28. A single model trained causally at every position (Das et al., 2024; Liu et al., 2024b) also outperforms models trained separately for each shorter history in 26 of 35 dataset–history comparisons, whereas circular mixing along time reads its own targets during training and is worse than causal mixing at every shorter history length. Across variables, the fluxer scales as $\mathcal{O}(C \log C)$ rather than the $\mathcal{O}(C^2)$ of attention, and in single precision it is faster than fused attention (Dao et al., 2022) at every measured length. TiF therefore provides a fast and accurate way to model high-dimensional time series while supporting forecasts from any history within its window.

1 INTRODUCTION

A multivariate forecaster maps a lookback window $\mathbf{X} \in \mathbb{R}^{C \times L}$ of C variables over L steps to a horizon $\mathbf{Y} \in \mathbb{R}^{C \times H}$, and every architecture is, at bottom, a choice of how to move information along the two axes of that tensor (Zeng et al., 2023; Nie et al., 2023; Liu et al., 2024a; Wang et al., 2024). The benchmarks that shaped the field carry at most a few hundred variables, so the design effort went into the time axis. Deployments have outgrown them. Traffic networks, power grids and web logs report from thousands to tens of thousands of channels, and the TIME-HD benchmark collects datasets of this kind (Ni et al., 2025). At that scale the variable axis is the binding constraint. Variable-wise attention (Liu et al., 2024a) scores every pair of variables, $\mathcal{O}(C^2)$ operations per layer and per sample, which data parallelism does not reduce. A fused kernel (Dao et al., 2022) spares the memory of the $C \times C$ score matrix but not the operations that fill it.

Spectral mixing is the cheap alternative. Multiplying the Fourier transform of a token sequence by a spectrum gives a global receptive field in $\mathcal{O}(n \log n)$, and it is the computational core of neural operators (Li et al., 2021; Guibas et al., 2022; Kovachki et al., 2023) and of the Fourier branch of forecasting (Zhou et al., 2022b;a; Yi et al., 2023b; Xu et al., 2024; Yang et al., 2024). By the convolution theorem, however, that step is a *circular* convolution. Its matrix is a circulant, so two tokens interact through their index difference *modulo* the transform length, and the end of the axis wraps back to its start. Zero-padding the transform removes the wrap at the same asymptotic cost, so the boundary condition of the convolution, which we call its *closure*, is a choice, and our claim is that the two axes of a multivariate series need different ones.

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.

Along time the circulant is misspecified. Its wrap-around entries let the end of the lookback window write into its start, so it cannot represent a pure delay without an echo. Inside a lookback window this is not a leak of unseen data, since every patch precedes the forecast, and a forecaster that flattens the whole window before predicting loses nothing to it. It becomes a leak once a model is trained the way decoder-style forecasters are, every position predicting what follows it (Das et al., 2024; Liu et al., 2024b), because the wrap-around then hands each position part of its own target. That training is worth having, since the sensors of one network come online at different times and a model trained at every position forecasts from whatever history each has. Padding to twice the length and discarding the tail removes the wrap-around exactly and leaves a lower-triangular Toeplitz matrix, the causal closure.

Across variables we leave the transform unpadded. Padding an axis asserts that it has ends, and on an axis whose labeling is arbitrary that assertion gives the variables stored first and last fewer contributions than the rest; the circular closure has no ends. It is not a set operator either. It imposes a ring on the variables, and once the kernel is read from the tokens, as ours is, the mixer is not equivariant even to a cyclic shift of the labels (Proposition 2). We therefore treat the closure as a choice to be measured, by exchanging it for a padded one and by relabeling the variables (Section M, Section N).

This is the design the title names, *causal in time, circular in space*, where *space* is the variable axis whether or not the variables have locations. Time Fluxer (TiF) runs one token mixer on both axes and sets the closure of each axis to match it; Propositions 1 and 2 fix what each closure is and what it buys. The mixer, the *fluxer*, is a gated long convolution (Poli et al., 2023) whose kernel is read from the input and applied through the transform.[1] Every other family of forecasters treats its two axes alike or leaves one of them unmixed (Table A1). Three results support the design. Replacing only the attention of iTransformer or PatchTST by the fluxer lowers error in 26 of 28 cells on the variable axis and 21 of 28 on time, a cell being one dataset at one horizon. Trained once at every position under the causal closure, one model serves any history within its window, while the same training under the circular closure reads its own targets (Figure 5); across variables, exchanging the closure shows no consistent effect on the error once tokens are held equal (Figure A6), and the ring is the cheaper of the two. In single precision the variable-axis mixing step is faster than fused attention at every measured length, by 268 times at 131,072 tokens, and the whole model by up to 3.4 times. Figure 1 summarizes the fixed-lookback TIME-HD results.

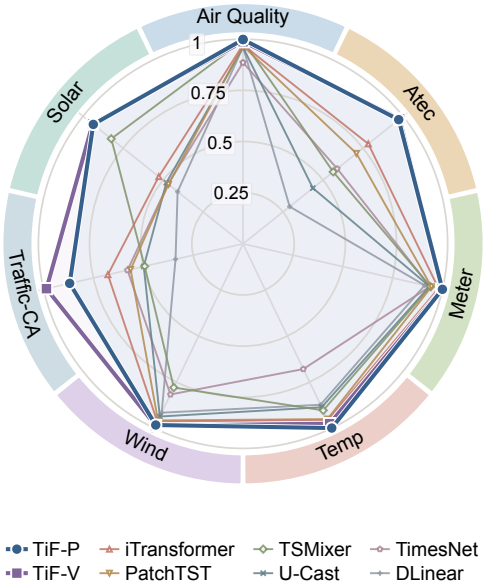

Figure 1: **Horizon-averaged accuracy on the seven TIME-HD datasets.** Each axis is a dataset; the radius is the lowest MSE in a cell divided by that of the model, averaged over the four horizons at $L = 96$, so 1 is best throughout. A fluxer configuration leads on every dataset. TiF-P and TiF-V are the configurations of Figure 2c; the six baselines are those of Table 1.

2 BACKGROUND

Deep forecasters differ mainly in how they treat the two axes. One line keeps the variables apart and spends its capacity along time, through recurrences (Hochreiter & Schmidhuber, 1997; Lin et al., 2026), attention over patches (Nie et al., 2023; Lee et al., 2024; Chen et al., 2024b), large-kernel convolutions (Wu et al., 2023; Luo & Wang, 2024) or linear maps over the whole window (Zeng et al., 2023; Das et al., 2023; Wang et al., 2024); where the variables are weakly coupled it is hard to beat (Zeng et al., 2023; Nie et al., 2023). A second line models the variables jointly, densely (Zhang & Yan, 2023; Liu et al., 2024a; Yu et al., 2024) or through learned groups (Chen et al., 2024a; Qiu et al., 2025), which strongly coupled datasets reward and which costs $\mathcal{O}(C^2)$. SOFTS (Han et al.,

[1]The kernel is in *flux*, generated anew from every input.

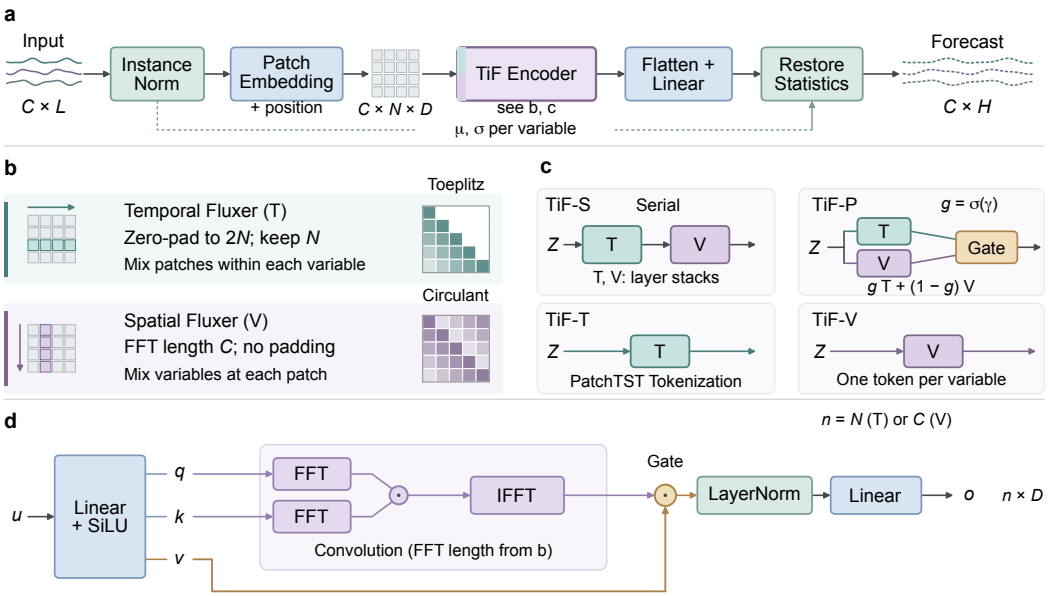

Figure 2: **One fluxer layer on both axes, padded along time and circular across variables. (a)** The pipeline. Each variable is normalized, cut into N patches embedded at width D, encoded, flattened and mapped linearly to the H horizon steps, and denormalized. **(b)** The two closures. Along time the transform is zero-padded to $2N$ and truncated, which leaves a lower-triangular Toeplitz matrix (Proposition 1); across variables it is left unpadded at length C, which keeps the circulant (Proposition 2). The shading is one kernel decaying with the index difference, and the entries the upper matrix discards are exactly the ones the lower one keeps. **(c)** The four encoder configurations we report. The two stages run in sequence (TiF-S) or side by side, joined by a per-channel gate (TiF-P), and each axis runs alone, TiF-T with the tokenization of PatchTST and TiF-V with one token per variable as in iTransformer. **(d)** The fluxer, the token mixer of a fluxer layer. In a spatial layer the kernel $\mathbf{k}$ is added to the learned static kernel of Equation (4) before its transform, which the panel leaves out.

2024) keeps the coupling at a cost linear in C by routing every series through one aggregated core (Section J), and FourierGNN (Yi et al., 2023a) makes every value in the window a node of one graph and mixes them in Fourier space. Patching (Nie et al., 2023) and reversible instance normalization (Kim et al., 2022; Liu et al., 2023b) are common to both lines.

The fluxer belongs to the gated long-convolution family, which replaces attention with a data-dependent convolution evaluated by fast transform, as in structured state-space models and their convolutional descendants (Gu et al., 2022; Poli et al., 2023). Across variables the fluxer adds a static kernel, trained once and applied to every window, which is what a Fourier layer of the forecasting literature learns (Yi et al., 2023b;a) and what a neural operator parameterizes in the spectrum (Li et al., 2021); Section M takes the two parts apart. Mixing tokens with a plain Fourier transform goes back to FNet (Lee-Thorp et al., 2022).

Causality in this family is a settled concern on an ordered axis. Temporal convolutional networks obtain it by padding one side of a local kernel (van den Oord et al., 2016; Bai et al., 2018), and Monarch Mixer (Fu et al., 2023) derives a causal parameterization of its sub-quadratic mixer so that it can be trained autoregressively, which is also how decoder-style forecasters train (Das et al., 2024; Liu et al., 2024b). That work settles how to make one ordered axis causal; which closure each axis of a multivariate series needs does not arise there.

3 Formulation

Setup Let $\mathbf{X} \in \mathbb{R}^{C \times L}$ denote the lookback window of C variables over L steps and $\mathbf{Y} \in \mathbb{R}^{C \times H}$ the H steps to be predicted. After patching with length P and stride Δ, each variable is represented by $N = \lfloor (L - P)/\Delta \rfloor + 1$ tokens of width D, giving a tensor $\mathbf{Z} \in \mathbb{R}^{C \times N \times D}$ (Figure 2a). A token mixer acts on one axis at a time. It takes a sequence $\mathbf{u} \in \mathbb{R}^{n \times d}$ of n tokens, $n = N$ along time and $n = C$ across variables, and returns a sequence of the same shape.

Mixing through the transform is circular convolution Let $\mathcal{F}_m$ denote the unnormalized m-point discrete Fourier transform along the token axis and $\mathcal{F}_m^{-1}$ its inverse. Given a query $\mathbf{q} \in \mathbb{R}^{n\times d}$ and a kernel $\mathbf{k} \in \mathbb{R}^{n\times d}$, the step

$$\mathbf{y} \;=\; \mathcal{F}_n^{-1}(\mathcal{F}_n(\mathbf{q}) \odot \mathcal{F}_n(\mathbf{k})) \tag{1}$$

evaluates, channel by channel, the circular convolution $y_t = \sum_{s=0}^{n-1} q_s\, k_{(t-s) \bmod n}$, in matrix form $\mathbf{y} = \mathbf{C}(\mathbf{k})\,\mathbf{q}$ with the circulant $\mathbf{C}(\mathbf{k})_{ts} = k_{(t-s) \bmod n}$. The two propositions below are statements about $\mathbf{C}$ and hold for any mixer built on Equation (1), whatever its kernel; both are elementary, and Section B proves them.

Proposition 1 (Causal closure along time)**.** *For any kernel* $\mathbf{k}$*, the circulant splits uniquely as* $\mathbf{C}(\mathbf{k}) = \mathbf{T}(\mathbf{k}) + \mathbf{U}(\mathbf{k})$ *with*

$$\mathbf{T}(\mathbf{k})_{ts} = k_{t-s}\,\mathbf{1}[s \leqslant t], \qquad \mathbf{U}(\mathbf{k})_{ts} = k_{n+t-s}\,\mathbf{1}[s > t], \tag{2}$$

where $\mathbf{T}$ *is lower-triangular Toeplitz and* $\mathbf{U}$ *is strictly upper-triangular. Replacing the* n*-point transforms in Equation (1) by* $2n$*-point transforms and keeping the first* n *outputs yields exactly* $\mathbf{y} = \mathbf{T}(\mathbf{k})\,\mathbf{q}$*, at unchanged* $\mathcal{O}(n \log n)$ *cost. If* $\mathbf{q}$ *and* $\mathbf{k}$ *are read from the tokens one token at a time,* y_t *then depends on the tokens* $0, \ldots, t$ *alone.*

Remark 1 (What the causal closure buys). The term $\mathbf{U}$ is not a small perturbation of $\mathbf{T}$; its entries are the long-lag taps of the same kernel, so a kernel that reaches far into the past reaches equally far into the future. The two matrices agree in their last row, where $\mathbf{U}$ is empty, so one circular layer and one causal layer compute the same sum at the final token; stacked layers do not, since the second layer reads the earlier positions of the first, and the two are different functions. Neither hands a position its own target when the head flattens the whole lookback; Section L measures what the difference is worth there. What the padding buys is training position by position. Since no position reads a later one, every position can be trained to forecast what follows it, and one model serves any history within the window (Figure 5); gated long-convolution models evaluate their convolutions this way for the same reason (Poli et al., 2023).

Proposition 2 (Circular closure across variables)**.** *Let* $\mathbf{S}$ *be the cyclic shift of the token index,* $(\mathbf{S}\mathbf{q})_t = q_{(t-1) \bmod n}$*. For any kernel* $\mathbf{k}$*, every row of* $\mathbf{C}(\mathbf{k})$ *holds all* n *taps* $k_0, \ldots, k_{n-1}$ *and* $\mathbf{C}(\mathbf{k})\,\mathbf{S} = \mathbf{S}\,\mathbf{C}(\mathbf{k})$*, whereas row* t *of* $\mathbf{T}(\mathbf{k})$ *holds only the* $t+1$ *taps* $k_0, \ldots, k_t$*. If* $\mathbf{q}$ *and* $\mathbf{k}$ *are read from the tokens one token at a time, shifting the tokens by* $\mathbf{S}^m$ *shifts* $\mathbf{y} = \mathbf{C}(\mathbf{k})\,\mathbf{q}$ *by* $\mathbf{S}^{2m}$*.*

Remark 2 (What the circular closure buys). On the variable axis the index carries no order, since relabeling the C variables leaves the data as it is. Padding the axis gives the variables stored first and last fewer contributions than the rest, while under the circular closure every variable receives all C taps and two variables interact through their index difference modulo C, which is as arbitrary as the labeling. The closure is not a symmetry of the model. Under a cyclic relabeling the part of the kernel read from the tokens moves the output by twice the shift and the static part by the shift itself, so the two do not even move together, and the weight that variable s contributes to variable t is the tap at index $t - s$, which the input-dependent part reads from whichever variable is stored there. Unlike attention, the mixer is not a set function of the variables; Section N measures what the dependence on the labeling costs.

4 Architecture

TiF runs the fluxer once per axis inside a patch-based encoder (Figure 2a), with the two stages wired in sequence or side by side (Figure 2c); we report both.

Normalization and patching Each window is normalized per variable and its statistics restored at the output (Kim et al., 2022). Four calendar features of the window (month, day of month, weekday and hour) enter as four extra variables, so the variable axis carries $C + 4$ tokens; Section E records which baselines read them. Each normalized series is cut into N disjoint patches of length $P = 24$ in the two-axis configurations, and into the tokens of its host in TiF-T and TiF-V (Table A5), each embedded to width D with a fixed sinusoidal position code, giving $\mathbf{Z} \in \mathbb{R}^{C\times N\times D}$.

Fluxer layer The fluxer (Figure 2d) reads a sequence $\mathbf{u} \in \mathbb{R}^{n\times D}$ and produces a query, a kernel and a gate,

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathrm{SiLU}(\mathbf{u}\mathbf{W}_{\mathrm{in}})\,, \qquad \mathbf{W}_{\mathrm{in}} \in \mathbb{R}^{D\times 3D}, \tag{3}$$

where $\mathbf{k}$ is read from the tokens one token at a time. Across variables the layer adds to that kernel a learned static kernel $\mathbf{B} \in \mathbb{R}^{n \times D}$, the same one for every input, while along time it uses the kernel read from the tokens alone,

$$\tilde{\mathbf{k}} = \mathbf{k} + \mathbf{B} \ \text{(variable axis)}, \qquad \tilde{\mathbf{k}} = \mathbf{k} \ \text{(time axis)}. \tag{4}$$

The query is then convolved with $\tilde{\mathbf{k}}$ through the transform, channel by channel,

$$\mathbf{y} = \mathcal{P}_n \, \mathcal{F}^{-1}_{n_{\text{pad}}} \Big(\mathcal{F}_{n_{\text{pad}}}(\mathbf{q}) \odot \mathcal{F}_{n_{\text{pad}}}(\tilde{\mathbf{k}}) \Big), \tag{5}$$

where the transforms are taken along the token axis and $\mathcal{P}_n$ keeps the first n positions. By linearity the spatial mixing matrix is the sum of two circulants, $\mathbf{C}(\mathbf{k}) + \mathbf{C}(\mathbf{B})$, one filter that the input writes and one that training leaves fixed; the second is the learned circular filter of the Fourier branch (Li et al., 2021; Yi et al., 2023b), and Section M removes each part in turn. The result is gated by the third projection, normalized and mapped back to width D,

$$\mathbf{o} = \text{LayerNorm}(\mathbf{y} \odot \mathbf{v}) \, \mathbf{W}_{\text{out}}, \qquad \mathbf{W}_{\text{out}} \in \mathbb{R}^{D \times D}. \tag{6}$$

A fluxer layer wraps the fluxer in a residual connection with post-normalization, $\mathbf{u} \leftarrow \text{LayerNorm}(\mathbf{u} + \mathbf{o})$, and follows it with a position-wise feed-forward network of the same width in a second residual connection; Section C gives the layer in code.

What separates the two layers The two layers run the same code and differ in $(n_{\text{pad}}, \mathcal{P}_n)$ and in the static kernel. The temporal layer sets $n_{\text{pad}} = 2n$ and truncates, giving the lower-triangular Toeplitz matrix of Proposition 1; the spatial layer sets $n_{\text{pad}} = n$ and $\mathcal{P}_n = \text{Id}$, giving the circulant of Proposition 2. The code uses orthonormal transforms with no extra factor, so it evaluates $n_{\text{pad}}^{-1/2}$ times the convolution of Equation (1); this scalar leaves the closure, causality and complexity statements unchanged.

The two stages The temporal stage $\mathcal{T}$ stacks ℓ_{T} *temporal fluxer layers*, each treating the N patches of one variable as its tokens, so that the variable axis is carried in the batch and every variable is processed alone. The cross-variable stage $\mathcal{S}$ stacks ℓ_{V} *spatial fluxer layers* and does the mirror image, the C rows at one patch position becoming its tokens and the patch axis carried in the batch. Both map $\mathbb{R}^{C \times N \times D}$ to itself, and each sees only the axis it is named for; a permutation of the other axis passes through it unchanged, which we verify numerically.

Wiring the two axes The *serial* wiring resolves the temporal structure first and mixes variables afterwards, $\mathcal{S}(\mathcal{T}(\mathbf{Z}))$, so the cross-variable stage sees patches that already carry their own history. The *parallel* wiring sends the patch embedding down both axes independently and joins them with a gate,

$$\mathbf{Z}^{\text{out}} = \sigma(\boldsymbol{\gamma}) \odot \mathcal{T}(\mathbf{Z}) + (1 - \sigma(\boldsymbol{\gamma})) \odot \mathcal{S}(\mathbf{Z}), \qquad \boldsymbol{\gamma} \in \mathbb{R}^{D}, \tag{7}$$

where $\boldsymbol{\gamma}$, one parameter per channel, is initialized at zero so both branches start at equal weight. A single projection then flattens the patch axis and maps it linearly to the horizon, the only place the horizon length enters, after which the input statistics are restored.

Training at every position The causal temporal stage admits a second head. One linear map, shared by all positions, reads the representation of patch t and predicts the H steps that follow it; the loss averages over positions, and at inference the forecast is read at the last patch for which data exist, so any whole number of patches up to the window is a valid history. By Proposition 1 the prediction at t does not depend on later patches, which makes this training well posed and is what the circular closure gives up. Patches do not overlap in this mode, and instance normalization is switched off, since its statistics span the window. Table 1 uses the flattened head and Figure 5 this one.

Complexity A temporal fluxer layer costs $\mathcal{O}(ND^2 + ND \log N)$ per variable and a spatial one $\mathcal{O}(CD^2 + CD \log C)$ per patch position, so the cross-variable stage is $\mathcal{O}(NCD^2 + NCD \log C)$ against the $\mathcal{O}(NC^2D)$ of cross-variable attention, linear in C up to the logarithm. The static kernel adds CD parameters to each spatial layer, which is the whole of the parameter count that TiF-V carries above iTransformer (Table A3).

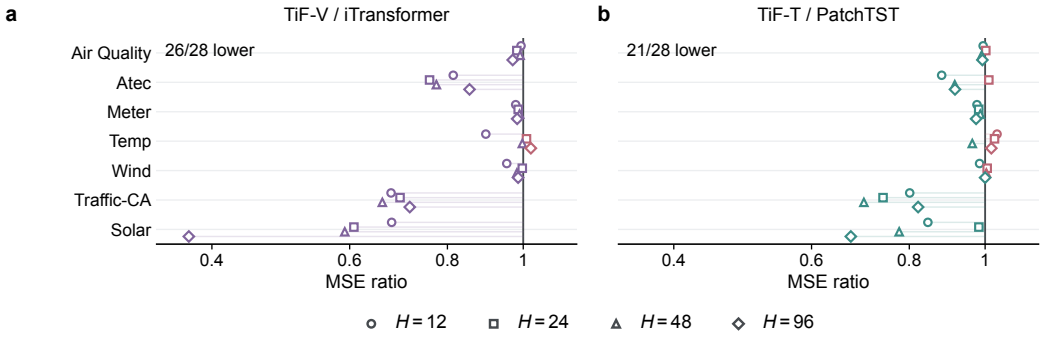

Figure 3: **Substituting the fluxer for attention, cell by cell.** Test MSE of the substituted model divided by that of the host it replaces, one row per dataset and one marker per horizon, at $L = 96$; a point left of the line is a cell where the fluxer layer has the lower error. **(a)** The variable axis, TiF-V against iTransformer. **(b)** The time axis, TiF-T against PatchTST. Table A4 gives the counts per dataset and the places where a variant departs from its host.

5 Experiments

Benchmark and protocol We evaluate on seven datasets of Time-HD (Ni et al., 2025): Air Quality ($C = 1105$), Atec (1569), Meter (2898), Temp (3774), Wind (3696), Solar (5162) and Traffic-CA (7490), which differ widely in what their two axes carry (Figure A2, Section D). Splits, scaling and metrics follow the reference implementation. The split is chronological, $70/10/20$, the standardizer is fitted on training data alone, and Mean Squared Error (MSE) and Mean Absolute Error (MAE) are averaged over horizon and channels, with no batch dropped and the test set evaluated in full. We fix the lookback at $L = 96$, following the long-term forecasting literature rather than the benchmark search over lookbacks, and sweep the horizon $H \in \{12, 24, 48, 96\}$, so that every model fills 28 cells, one per dataset and horizon; Section P repeats the comparison at $L = 720$ on five of the datasets.

Baselines and budget We compare against two channel-independent baselines, DLinear (Zeng et al., 2023) and PatchTST (Nie et al., 2023), and four channel-dependent ones, TSMixer (Chen et al., 2023), TimesNet (Wu et al., 2023), iTransformer (Liu et al., 2024a) and U-Cast (Ni et al., 2025). All seven methods share one pipeline and budget, Adam on MSE at learning rate 10^{-3} with a tenth of warmup and cosine decay, at most twenty epochs, patience five and one seed, each in one untuned configuration at the common width and depth of Table A3 and otherwise the settings of its reference implementation, less the calendar embedding of TimesNet (Section E). This controls tuning effort, but the cells share datasets and are not independent trials, so we read the one-seed grid descriptively and compare each difference with the *run-to-run spread*, the largest gap between repeated runs of one cell (Section L), as a yardstick rather than a test. Batch size is set per dataset and model by memory and recorded with each result (Table A6), and Section F checks our two strongest channel-dependent baselines against the numbers the benchmark reports.

The reported metric Two reference forecasts are computed on the same test windows (Table A8), the training mean (*zero*) and the last value held flat (*persistence*). Datasets at this scale differ enormously in how much of their variance is trivially predictable, and three of the six baselines lose to persistence somewhere in our grid, so we report the fraction of the better reference error that a model removes,

$$\mathcal{S} \;=\; 1 \;-\; \frac{\mathrm{MSE}}{\min(\mathrm{MSE}_{\mathrm{zero}},\; \mathrm{MSE}_{\mathrm{persistence}})}, \tag{8}$$

with $\mathcal{S} = 0$ no better than the trivial forecast and $\mathcal{S} = 1$ exact. The minimum matters because which reference is stronger changes with the horizon, and the score puts the datasets on one scale where raw MSE spans 0.030 to 1.45; Section H carries the raw MSE and MAE of every cell.

The fluxer against the attention it replaces That one operator serves both axes is tested by running each half against the Transformer built for that half. Each substitution keeps the width, depth, normalization, head and schedule of its host and changes the mixer, along with the minor details

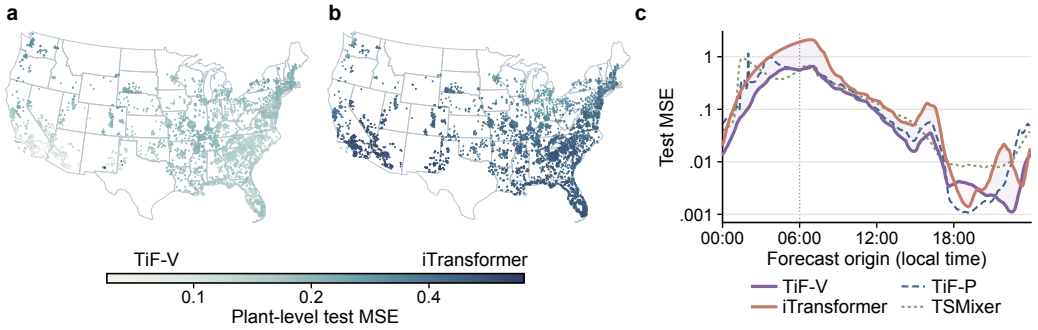

Figure 4: **TiF-V against iTransformer on SOLAR, by plant and by time of day.** TiF-V has the lower error at nearly every plant, and its lead is widest for forecasts that cover the sunrise. **(a, b)** Test MSE of each of the 5162 plants at $L = 96$ and $H = 96$, drawn at the coordinates recorded in its column name on one logarithmic color scale, over the state boundaries of the US Census Bureau; neither model is given the coordinates. **(c)** Test MSE over the whole test set, averaged by the local time at which the forecast starts; the dotted line marks 06:00, the origin at which the two are furthest apart.

Table A4 lists; on TRAFFIC-CA and SOLAR iTransformer also trains at half the batch of TiF-V, the most that fits (Table A6). TiF-V is iTransformer with its attention replaced by the spatial fluxer layer, one token per variable as before, and TiF-T does the same to PatchTST on the time axis, with the patch length and stride it publishes. TiF-V carries the static kernel of its spatial layers, which the host has no counterpart for, so it is the larger of the two (Table A3); the arm that drops it matches the parameter count of iTransformer to within 0.1% at 4.8 M and still takes 21 of 28 cells from it, 12 beyond the run-to-run spread and zero against (Section M).

Both substitutions have the lower error in most cells (Figure 3), 26 of 28 on the variable axis and 21 of 28 on time. On the variable axis the largest gains are on SOLAR, TRAFFIC-CA and ATEC, and the error falls to 0.37 of the host's; on time the largest ratio is 0.67. Figure 4 takes SOLAR plant by plant, where TiF-V has the lower error at 98.1% of the plants and its widest lead on forecasts that cover the sunrise (Section I). Of the lower-error cells, 19 and 14 lie beyond the run-to-run spread, against zero and zero of the higher-error cells, and by MAE the counts are 23 and 21. Averaged over horizons, the substitution is lower on seven of seven datasets on the variable axis and on six on time, the exception being TEMP (Table 1).

Results Whether modeling the variable axis pays depends on its mixer. Against the better of the two channel-independent baselines, iTransformer wins 21 of 28 cells, TSMixer 11 of 28, TimesNet 6 of 28 and U-Cast 2 of 28. As fixed variants, TiF-P leads the best of the six baselines on 25 of 28 cells, TiF-V on 22 and TiF-S on 20. In two of the 28 cells the best baseline leads every variant of ours, AIR QUALITY at $H = 12$ and TEMP at $H = 96$, where TSMixer and PatchTST are ahead by 0.0001 and 0.0006 MSE respectively, a fraction of the run-to-run spread. With SOFTS (Han et al., 2024), the linear-cost alternative along the variable axis (Section J), added to the baselines, the lead is largest on SOLAR and ATEC, 8% to 25% of the MSE of the best baseline, and at most 10% on the other five datasets; TiF-V has lower error than SOFTS in 27 of 28 cells. A per-cell validation choice between our two wirings gives 26 of 28 (Section K), but the baselines receive no comparable architecture choice, so the fixed-variant counts are the primary comparison. TiF-T leads on 7, with PatchTST its proper comparison above. The variable axis alone comes within three cells of the better two-axis wiring at $1/9$ of the memory of a two-axis model (Table A16). On a four-dataset subset fixed before any ablation outcome was seen, removing the static kernel is worse in 15 of 16 cells and removing the input-dependent kernel in 12 of 16; the gate, ablated on the full grid, shows no consistent effect (Section M).

One model for histories within the window Under the protocol of Table 1 the padding shows no consistent effect on the test error; with a head that flattens the whole lookback, removing it moves no cell beyond the run-to-run spread (Section L). What the causal closure buys is the training of Section 4, every position forecasting what follows it (Das et al., 2024; Liu et al., 2024b), so that one TiF-T forecasts from whatever history each variable has. Trained once, it has the lower error in 26 of 35 cases against TiF-T trained separately for each shorter history, at a median of 0.97 of their error (Figure 5a), and its one training run took between $1/13$ and $1/6$ of the time of the six it replaces. It is

Table 1: **Skill against the better trivial forecast on TIME-HD, four horizons at $L = 96$.** $\mathcal{S}$ of Equation (8); higher is better, 0 is no better than the trivial forecast, and the best entry in each row is **shaded and bold** and the second best shaded and underlined, ranked on the underlying MSE rather than on the rounded score, so two cells that print the same value can carry different marks. All four variants are shown and none is selected against the others; TiF-S and TiF-P run both axes, TiF-T the time axis alone and TiF-V the variable axis alone. Every model runs its default configuration, untuned, on all datasets and horizons (Table A3).

Dataset	H	DLinear (2023)	TSMixer (2023)	TimesNet (2023)	PatchTST (2023)	iTransformer (2024a)	U-Cast (2025)	TiF-T (Ours)	TiF-V (Ours)	TiF-S (Ours)	TiF-P (Ours)
Air Quality	12	0.431	**0.467**	0.411	0.441	0.452	0.440	0.445	0.456	0.460	0.467
	24	0.449	0.466	0.370	0.456	0.456	0.447	0.455	0.466	0.463	**0.469**
	48	0.434	0.435	0.377	0.431	0.434	0.430	0.436	0.439	**0.447**	0.444
	96	0.398	0.414	0.358	0.395	0.395	0.392	0.400	0.414	0.412	**0.416**
Atec	12	0.179	0.603	0.488	0.668	0.707	0.446	0.707	0.761	**0.768**	0.759
	24	0.257	0.707	0.689	0.771	0.772	0.573	0.769	0.827	**0.832**	0.828
	48	0.444	0.741	0.774	0.787	0.813	0.640	0.805	0.855	**0.857**	0.855
	96	0.610	0.714	0.786	0.786	0.814	0.709	0.804	0.841	**0.844**	0.839
Meter	12	0.308	0.336	0.315	0.342	0.366	0.335	0.358	0.380	**0.380**	0.380
	24	0.289	0.313	0.309	0.317	0.340	0.311	0.330	**0.350**	0.349	0.349
	48	0.279	0.296	0.297	0.303	0.325	0.294	0.313	0.332	0.330	**0.341**
	96	0.260	0.277	0.294	0.273	0.305	0.268	0.293	0.318	0.310	**0.318**
Temp	12	0.714	0.746	0.570	0.766	0.752	0.735	0.758	0.778	0.777	**0.781**
	24	0.657	0.650	0.550	0.697	0.707	0.666	0.688	0.705	0.700	**0.711**
	48	0.611	0.638	0.517	0.611	0.636	0.609	0.625	0.637	0.627	**0.644**
	96	0.574	0.563	0.502	**0.587**	0.578	0.564	0.579	0.568	0.570	0.586
Wind	12	0.349	0.282	0.184	0.371	0.393	0.367	0.381	**0.422**	0.415	0.417
	24	0.374	0.257	0.282	0.400	0.404	0.381	0.396	0.405	**0.421**	0.414
	48	0.306	0.161	0.239	0.340	0.335	0.322	0.337	0.348	**0.358**	0.341
	96	0.242	0.086	0.228	0.281	0.277	0.262	0.281	0.288	**0.298**	0.282
Traffic-CA	12	0.879	0.940	0.928	0.930	0.943	0.916	0.944	**0.961**	0.960	0.956
	24	0.868	0.894	0.923	0.921	0.937	0.915	0.942	**0.956**	0.956	0.948
	48	0.838	0.899	0.914	0.907	0.924	0.893	0.935	0.950	**0.952**	0.946
	96	0.864	0.873	0.915	0.912	0.922	0.902	0.928	0.944	**0.945**	0.937
Solar	12	0.154	0.542	-0.089	0.345	0.377	0.293	0.447	0.577	0.372	**0.597**
	24	0.247	0.645	0.304	0.407	0.457	0.365	0.418	0.670	0.482	**0.674**
	48	0.359	0.678	0.521	0.396	0.503	0.418	0.531	0.706	0.721	**0.760**
	96	0.352	0.696	0.574	0.240	0.392	0.458	0.488	**0.773**	0.755	0.694

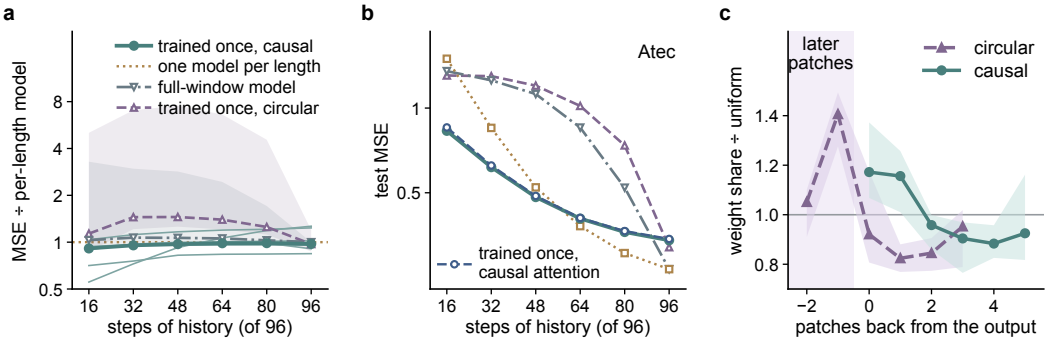

Figure 5: **One model for histories within the window, under each closure of the time axis.** TiF-T at $L = 96$ and $H = 96$, every position trained to forecast what follows it; under the causal closure it forecasts from any history within the window at the level of models trained separately for each length, and under the circular closure the same training reads its own targets. **(a)** Test MSE divided by that of TiF-T trained separately at each history length (dotted line), against the steps of the window that are real data. Under the causal closure, thin lines are the seven datasets and the thick line their median; under the circular closure, and for the full-window TiF-T of Table 1 with the missing steps filled in, the median and range are drawn, leaving out the cases where the latter does worse than the trivial forecast. **(b)** The same comparison in MSE on ATEC, with the same training under causal attention of equal size (blue dashes). **(c)** Share of the mixing weight of the two models at each kernel tap, divided by that of a flat kernel (mean and range over datasets), each tap drawn at the patch it reads in most rows; the shaded taps read later patches, which only a circular transform reaches. Section L gives every dataset.

not better everywhere. On WIND the per-length models win at every length, and at the full window the flattened head is ahead by up to 27%, which is why Table 1 keeps that head. Handing the shorter

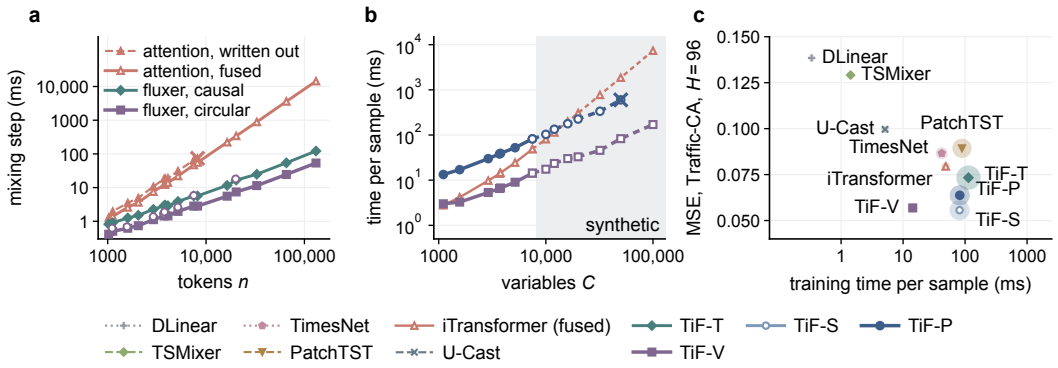

Figure 6: **Training cost in single precision, mixing step and whole model.** The mixing step is faster than fused attention at every measured length, and the whole model on 6 of 7 datasets and beyond them. **(a)** One training step of each mixer at variable-axis token counts, width 512 and batch four; hollow circles are dataset lengths $C + 4$ and a cross the last length fitting an 80 GB accelerator (Section O). **(b)** Whole-model training time per sample. **(c)** Error against time on TRAFFIC-CA at $H = 96$; disc area is peak memory and iTransformer uses fused attention.

history to the model of Table 1 with the missing steps filled in needs no training at all, and is worse in 30 of 35 cases, by a median factor of 1.1. Under the same training, causal multi-head attention of the same size has the higher error in 26 of 42 cases, at a median of 1.01 times ours, and lands within the run-to-run spread in all but one (Section L); the capability comes with the causal closure, which both mixers have, and the fluxer carries it about as well as attention. Under this training the circular transform leaks targets. Every position but the last reads the patch after it, which holds part of its own target and takes the largest mixing weight in all seven datasets (Figure 5c), and at shorter histories its error is a median of 1.6 times that of the causal model.

The closure across variables Across variables the same exchange shows no consistent effect either. With the calendar tokens on the axis, padding it leaves six of the seven datasets inside the run-to-run spread and costs SOLAR 7% to 60%; with them removed, the circular, one-sided and symmetric closures show no consistent difference on ATEC and SOLAR (Section M). The case for the ring is that it needs no first or last variable and that it is the cheaper of the two, not that it lowers the test error. Padding doubles the length an axis transforms, which costs 2.0 to 2.3 times the mixing step at the same token count (Figure 6a), on the axis that carries the most tokens. Its price is that reordering the variables moves our error, by 1.16% against 0.41% for iTransformer, 3 to 14 times less than our lead over it (Section N).

Cost, measured In single precision, the lowest precision available to the transform at these lengths, the circular fluxer is faster than fused attention throughout, by 2.5 times at 1024 tokens and 268 at 131,072 (Figure 6a). This is a same-precision result; with attention in `bfloat16` and the transform in single precision, attention is faster below 4096 tokens and the fluxer above it (Section O). Whole-model overhead narrows the gap, but fused iTransformer still takes 0.9 to 3.4 times as long as TiF-V on the seven datasets and more beyond them (Figure 6b). Fused attention uses less memory than TiF-V because it never stores its scores; the fluxer saves time, not memory. Along time it is no cheaper than attention, since that axis has only eleven patches per variable.

6 CONCLUSION

One gated long convolution, closed differently on each axis, replaces attention on both axes of a multivariate series. It has lower error than variable-wise attention in 26 of 28 cells and than patch-wise attention in 21 of 28, and across variables it scales as $\mathcal{O}(C \log C)$, saving time rather than memory, with an advantage that depends on numerical precision. Along time, twofold padding removes the wrap-around and lets one model trained at every position serve any history within the window, where circular training reads its own targets. Across variables, the circular closure gives no variable an end and is the cheaper of the two, and with equal tokens it shows no consistent difference in error from the padded alternatives. The two axes of a multivariate series need different closures, and one operator supplies both.

AI USE STATEMENT

Large Language Models (LLMs) were used during this project as coding and writing assistants, including to help draft and refactor experiment and analysis scripts, revise prose, and check the consistency of the manuscript with its tables and figures. The empirical results reported in this paper are based on experiment runs conducted and reviewed by the authors, with tables and figures generated primarily from recorded experimental outputs. Citations and LLM-assisted content were reviewed by the authors. The research questions, experimental design, evaluation choices, and interpretation of the results were determined by the authors, who take responsibility for the final content of the paper.

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work trains and evaluates forecasting models on public datasets of the TIME-HD benchmark (Ni et al., 2025), which record infrastructure, energy and meteorological measurements, and it involves no human subjects and no personal data. The only locations it uses are the plant coordinates in the released files of SOLAR and the sensor coordinates in the metadata of LARGEST (Liu et al., 2023a), both public; they serve to describe the data and are never given to a model. The contribution is a token mixer, and it carries no risk beyond that of forecasting such measurements at all, where an error can go unnoticed by whoever acts on the forecast. We therefore report every cell beside two trivial forecasts (Section G) and report the settings in which our models lose alongside those in which they win.

REPRODUCIBILITY STATEMENT

All experiments use the public TIME-HD datasets under the chronological splits and the standardizer of the reference implementation, and Section D describes the seven datasets we use. Section 5, Table A3 and Table A5 give the optimizer, schedule, epoch budget, early-stopping rule and configuration, which are identical for all seven methods, and Table A6 gives the batch of every run at $L = 96$. Section Q describes the per-run records from which every number in the tables is generated, each of which carries the model, dataset, lookback, horizon, seed, parameter count, the number of accelerators and the batch size on each after any out-of-memory reduction, and the two trivial baselines computed on the same test windows. The method is a token mixer that can be dropped into an existing forecaster without changes elsewhere. The code, including the model, the sweep scripts, the causality checks and the scripts that generate the tables from the records, will be released upon acceptance.

REFERENCES

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 3

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 *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. 2

Peng Chen, Yingying ZHANG, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale Transformers with adaptive pathways for time series forecasting. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024b. 2

Si-An Chen, Chun-Liang Li, Nathanael C. Yoder, Sercan O. Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. 6, 8, A6, A17

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with TiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. 2

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of International Conference on Machine Learning (ICML)*, 2024. 1, 2, 3, 7, A9

Daniel Y Fu, Simran Arora, Jessica Grogan, Isys Johnson, Sabri Eyuboglu, Armin W Thomas, Benjamin Spector, Michael Poli, Atri Rudra, and Christopher Ré. Monarch Mixer: A simple sub-quadratic GEMM-based architecture. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022. 3

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive Fourier neural operators: Efficient token mixers for Transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022. 1

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 7, A7

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9: 1735–1780, 1997. 2

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022. 3, 4, A4

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24:1–97, 2023. 1

Seunghan Lee, Taeyoung Park, and Kibok Lee. Learning to embed time series patches independently. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 2

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. FNet: Mixing tokens with Fourier transforms. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2022. 3

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 5, A14

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. SegRNN: Segment recurrent neural network for long-term time-series forecasting. *IEEE Internet of Things Journal*, 13(5):9861–9871, 2026. 2

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. LargeST: A benchmark dataset for large-scale traffic forecasting. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023a. 10, A15

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 *Proceedings of International Conference on Learning Representations (ICLR)*, 2024a. 1, 2, 6, 8, A1, A6, A17

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained Transformers are large time series models. In *Proceedings of International Conference on Machine Learning (ICML)*, 2024b. 1, 2, 3, 7, A9

Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023b. 3

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 2, A1

Juntong Ni, Shiyu Wang, Zewen Liu, Xiaoming Shi, Xinyue Zhong, Zhou Ye, and Wei Jin. U-Cast: Learning hierarchical structures for high-dimensional time series forecasting. *arXiv preprint arXiv:2507.15119*, 2025. 1, 6, 8, 10, A1, A4, A5, A6, A17

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with Transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023. 1, 2, 3, 6, 8, A1, A6, A17

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *Proceedings of International Conference on Machine Learning (ICML)*, 2023. 1, 2, 3, 4

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2025. 2

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 3

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 1, 2, A1

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023. 2, 6, 8, A1, A6, A17

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with $10k$ parameters. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 1, A1

Runze Yang, Longbing Cao, Jianxun Li, and Jie Yang. Rethinking Fourier transform from a basis functions perspective for long-term time series forecasting. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, A1

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023a. 3

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023b. 1, 3, 5, A1

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: learnable decomposition with inter-series dependencies and intra-series variations modeling. In *Proceedings of International Conference on Machine Learning (ICML)*, 2024. 2

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers effective for time series forecasting? In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 1, 2, 6, 8, A1, A6, A17

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023. 2

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved Legendre memory model for long-term time series forecasting. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022a. 1

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed Transformer for long-term series forecasting. In *Proceedings of International Conference on Machine Learning (ICML)*, 2022b. 1

A Token-Mixing Operators by Family

Table A1: **Token-mixing operators of the main forecaster families, as the matrix each applies along the mixed axis.** A circulant is a convolution with periodic boundary and a causal Toeplitz one that reads only the past; N is the number of time tokens and C the number of variables. Representative models are DLinear and TimeMixer (Zeng et al., 2023; Wang et al., 2024) (MLP), TimesNet and ModernTCN (Wu et al., 2023; Luo & Wang, 2024) (CNN), PatchTST and iTransformer (Nie et al., 2023; Liu et al., 2024a) (Transformer), U-Cast (Ni et al., 2025) (latent query), and FreMLP, FITS and FBM (Yi et al., 2023b; Xu et al., 2024; Yang et al., 2024) (Fourier). A Fourier layer is a circulant when it acts on each frequency separately and a dense operator when it maps across frequencies, as the interpolation of FITS does. Only the last row gives the two axes different closures; its variable-axis circulant is the sum of a filter read from the tokens and a learned static one, so the Fourier row is the static half of it.

Family	Time-axis operator	Variable-axis operator	Cost on the variable axis
MLP	dense $N \times N$	none (channel-independent)	–
CNN	banded (local kernel)	none, or a dense map	up to $\mathcal{O}(C^2)$
Transformer	dense attention, unmasked	dense attention	$\mathcal{O}(C^2)$
Latent query	dense attention	attention through C/r latents	$\mathcal{O}(C^2/r)$
Fourier	circulant, or dense across frequencies	the same, or none	$\mathcal{O}(C \log C)$
TiF (ours)	**causal Toeplitz**	**circulant**	$\mathcal{O}(C \log C)$

Table A1 places the fluxer among the token mixers of multivariate forecasters, each written as the matrix it applies along the axis it mixes. Every other family treats its two axes alike or leaves one of them unmixed; the last row is the one that closes them differently.

B Proofs

Proof of Proposition 1. Writing the modular index as $t - s$ when $s \leqslant t$ and $n + t - s$ otherwise separates the two triangles, and the two cases are disjoint and exhaustive, so the decomposition is unique. Padding both sequences with n zeros makes the circular convolution of length $2n$ agree with the linear convolution on its first n entries, which is $\sum_{s \leqslant t} q_s k_{t-s}$; the cost changes by a constant factor. Each term of that sum reads q_s with $s \leqslant t$ and k_{t-s} with $t - s \leqslant t$, so if both are read one token at a time, y_t depends on the tokens $0, \dots, t$ alone. The statement is about unnormalized transforms; an orthonormal pair multiplies the convolution by $1/\sqrt{n_{\text{pad}}}$. □

Proof of Proposition 2. The entry $\mathbf{C}(\mathbf{k})_{ts} = k_{(t-s) \bmod n}$ depends on $t - s$ modulo n alone. As s runs over the n indices, so does $(t - s) \bmod n$, and row t holds every tap once; for the same reason $(\mathbf{S}\,\mathbf{C}(\mathbf{k})\,\mathbf{S}^{-1})_{ts} = \mathbf{C}(\mathbf{k})_{t-1,s-1} = \mathbf{C}(\mathbf{k})_{ts}$. The entry $\mathbf{T}(\mathbf{k})_{ts}$ vanishes for $s > t$, which leaves the $t + 1$ taps $k_0, \dots, k_t$ in row t. If $\mathbf{q}$ and $\mathbf{k}$ are read one token at a time, shifting the tokens by $\mathbf{S}^m$ replaces q_a by q_{a-m} and k_b by k_{b-m}, so that, with indices modulo n, the output becomes $\sum_a q_{a-m} k_{t-a-m} = \sum_{a'} q_{a'} k_{t-2m-a'} = y_{t-2m}$. □

C The Fluxer Layer in Code

Figure A1 gives the fluxer layer of Section 4 in the notation of PyTorch, with dropout left out. The temporal and spatial layers run the same code and differ in the flag `causal` and in whether a static kernel is passed.

D Datasets

The seven datasets come from Time-HD (Ni et al., 2025); Table A2 gives their size and sampling interval.

Figure A2 is what the two axes carry before any model is fitted, measured on the training segment alone and in the standardized units the models read. The variable axis is not the same object from one of these datasets to the next. On Meter, 90% of the variance is spread over 1868 of its 2898 directions, while on Traffic-CA the same share sits in 9 of 7490, and the mean absolute correlation

```
def fluxer(u, causal, B=None):                          # u: (batch, n, D)
    n = u.shape[1]
    q, k, v = silu(W_in(u)).chunk(3, dim=-1)            # query, kernel, gate
    if B is not None:                                   # variable axis only
        k = k + B[:n]                                   # B: (n, D), learned
    m = 2 * n if causal else n                          # pad along time only
    Q = rfft(q, n=m, dim=1, norm='ortho')
    K = rfft(k, n=m, dim=1, norm='ortho')
    y = irfft(Q * K, n=m, dim=1, norm='ortho')[:, :n]
    return W_out(layer_norm(y * v))

def fluxer_layer(u, causal, B=None):
    u = layer_norm(u + fluxer(u, causal, B))
    return layer_norm(u + ffn(u))
```

Figure A1: **The fluxer layer in code.** `W_in` maps the width D to $3D$ and `W_out` back to D; `B` is the learned static kernel of Equation (4), which the spatial layer carries and the temporal one does not; `ffn` is the position-wise feed-forward network. The two layers of Figure 2b differ in `causal` and in `B`.

Table A2: **The seven datasets.** C is the number of channels in the released file and T the number of timesteps, both counted from the file itself. The benchmark paper lists 3850 channels for both TEMP and WIND; the released files hold 3774 and 3696.

Dataset	C	T	Sampling
Air Quality	1105	15,461	6 hours
Atec	1569	8928	10 min
Meter	2898	28,512	30 min
Temp	3774	17,544	1 hour
Wind	3696	17,544	1 hour
Solar	5162	105,120	5 min
Traffic-CA	7490	43,824	1 hour

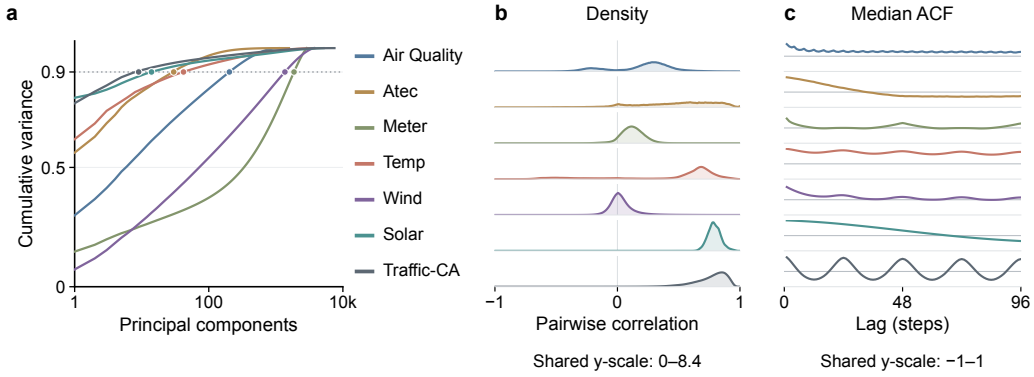

Figure A2: **Structure along the two axes of the seven datasets.** All three panels use the training segment alone, in the standardized units the models read; the variable axis carries very different amounts of structure from one dataset to the next, and the time axis does not follow it. **(a)** Share of the variance held by the leading principal components of the correlation between variables; the dotted line marks 90%. **(b)** Density over pairs of variables of the correlation between them, one row per dataset in the order of the legend. **(c)** Median over variables of the autocorrelation at each lag, out to the lookback of 96 steps, in the rows of (b). A step is a different amount of time in each dataset (Table A2), so the oscillations in (c) sit at different periods and a given lag falls at a different point of the daily cycle.

between two variables runs from 0.07 to 0.79. The time axis varies as well, and independently of it. At a lag of 96 steps the median autocorrelation is 0.81 on TRAFFIC-CA and -0.36 on SOLAR, whose step is five minutes rather than an hour, so the two lags fall at different points of the daily cycle rather than at different strengths of it.

The seven datasets therefore do not ask the same thing of a variable-axis operator, and the ones that ask the most are the ones where the substitution of Figure 3 gains the most. The three datasets it

gains most on, SOLAR, TRAFFIC-CA and ATEC, are the three whose variance is most concentrated, TRAFFIC-CA, SOLAR and ATEC; the next most concentrated, TEMP, is the exception, the dataset where the substitution gains least of all seven, at 1.00 of the error of the host. We report the coincidence rather than a relationship, which seven datasets could not establish.

E CONFIGURATIONS

Table A3: **What each model is given.** Width is the embedding size, and a dash means the model mixes in the input space without one. Depth counts blocks along each mixed axis. The feed-forward width is the hidden size of the position-wise network inside one block, which for TSMixer means its two mixing networks and for TimesNet its inception layers. Params is the range of trainable parameters over the seven datasets and four horizons, since it changes with the number of variables and with the horizon.

Model	Width	Depth	FFN width	Dropout	Params (M)	Mixing along
DLinear	–	–	–	–	0.002–0.019	nothing; one linear map per channel
TSMixer	–	3	512	0.1	3.7–23	both, an MLP over time steps and then one over variables
TimesNet	512	3	512	0.1	452–465	time, inception over the 5 dominant periods; variables only in the embedding
PatchTST	512	3	512	0.1	4.8–5.3	time, attention over patches of 16 at stride 8
iTransformer	512	3	512	0.1	4.8	variables, attention over C tokens
U-Cast	512	3	512	0.1	6.7–6.9	variables, attention through $C/16$ latents
TiF-T	512	3	512	0.1	4.8–5.3	time, fluxer, patches of 16 at stride 8
TiF-V	512	3	512	0.1	6.5–16	variables, fluxer, one token per variable
TiF-S	512	3 + 3	512	0.1	11–21	both in sequence, fluxer, patches of 24 without overlap
TiF-P	512	3 + 3	512	0.1	11–21	both side by side and joined by a gate, fluxer, patches of 24 without overlap

Table A4: **Replacing attention with the fluxer layer, and holding the rest fixed.** Cells won out of those completed, at four horizons per dataset. TiF-V matches iTransformer in token count, width, depth, normalization and head, differing only in the mixer and in two details that carry no information, a bias in the embedding and a positional encoding that is a constant when there is one token per variable. TiF-T matches the patch length and stride of PatchTST and its depth, but forms one patch fewer than PatchTST, which appends one more by replication; it also uses our own unbiased patch projection and a fixed rather than learned positional encoding. Batches are those of Table A6, which on TRAFFIC-CA and SOLAR give iTransformer half the batch of TiF-V. Both variants are complete on every cell.

	Time axis		Variable axis	
Dataset	PatchTST	TiF-T	iTransformer	TiF-V
Air Quality	1	**3**	0	**4**
Atec	1	**3**	0	**4**
Meter	0	**4**	0	**4**
Temp	**3**	1	2	2
Wind	2	2	0	**4**
Solar	0	**4**	0	**4**
Traffic-CA	0	**4**	0	**4**
All	7	**21**	2	**26**

Table A3 lists what every model is given. The two substitution variants are matched to the host each replaces, TiF-T to PatchTST down to the patch length and stride, TiF-V to iTransformer down to the one-token-per-variable embedding, which is what makes Table A4 a comparison of operators. The rule is otherwise one width and one depth per mixed axis for all seven methods, so that the tables compare operators at a common width and depth. TiF mixes two axes and receives on each the depth a single-axis baseline receives on its one; that is a choice, and it is why TiF carries six blocks where the baselines carry three, TSMixer included, whose three blocks each mix both axes.

Two entries stay unequal. Equal width is not equal size, and at the width of 512 that its reference implementation also uses on Traffic, the inception layers of TimesNet, six kernels from 1×1 to 11×11, give it at least 19 times as many parameters as any other model in Table A3. PatchTST patches with length 16 and stride 8, as its authors publish, against our length 24 without overlap, so it

forms three times as many tokens along time as we do; the asymmetry runs against us and we leave it, since shortening its patches would move it away from its published configuration.

Four calendar features (month, day of month, weekday and hour) are read only by models that keep one token or channel per variable, so that they are never folded into a compressed representation of the variables. iTransformer reads them as four extra tokens, as its reference implementation does, and TiF-V, TiF-S and TiF-P read them as four extra channels; TiF-T concatenates them too, but without a variable stage they never reach a forecast variable. TSMixer and U-Cast pass the variables through a narrower representation, a channel-mixing network of width 512 and latent queries that keep one sixteenth of them per level, and neither reference implementation reads the features. TimesNet embeds the C variables around each step into a single vector of width 512, and its reference implementation adds an embedding of the features to that vector; we leave that embedding out, which makes TimesNet the one model that reads the features in its reference implementation but not in ours. DLinear and PatchTST treat the variables independently, and their reference implementations do not read the features.

Table A5: **Default configuration of the four variants of TiF.** Every entry is the same on all seven datasets and all four horizons, and none was tuned; the training settings are those of every model in Section 5, and the batch of each run is in Table A6. TiF-T appends the calendar features as well, but without a variable stage they never reach a forecast variable.

Setting	TiF-T	TiF-V	TiF-S	TiF-P
Lookback L	96	96	96	96
Patch length and stride	16, 8	96, 96	24, 24	24, 24
Tokens per variable N	11	1	4	4
Width D	512	512	512	512
Temporal fluxer layers	3	–	3	3
Spatial fluxer layers	–	3	3	3
Static kernel per spatial layer	–	$(C{+}4) \times 512$	$(C{+}4) \times 512$	$(C{+}4) \times 512$
Feed-forward width	512	512	512	512
Dropout	0.1	0.1	0.1	0.1
Wiring of the two stages	–	–	serial	parallel
Calendar features	four extra variables: month, day of month, weekday, hour			
Normalization	reversible instance normalization per variable (Kim et al., 2022)			
Head	all patches flattened, one linear map to the H steps			
Parameters (M)	4.8–5.3	6.5–16	11–21	11–21
Optimizer and loss	Adam on MSE			
Learning rate	10^{-3}, a tenth of the steps as warmup, cosine decay			
Epochs	at most 20, early stopping with patience 5			
Seed	42			

Table A6: **The batch each model trained with, per accelerator times the number of accelerators, at $L = 96$.** Batch size is the one setting the protocol leaves to memory; the starting value is set per dataset and model, halved whenever a run does not fit, and recorded with the run.

Dataset	DLinear	TSMixer	TimesNet	PatchTST	iTransformer	U-Cast	TiF-T	TiF-V	TiF-S	TiF-P
Air Quality	16×2	32×1	32×1	16×2	16×2	16×2	16×2	16×2	16×2	16×2
Atec	16×2	32×1	32×1	8×2	16×2	16×2	8×2	16×2	16×2	16×2
Meter	16×2	32×1	32×1	8×2	16×2	16×2	8×2	16×2	16×2	16×2
Temp	16×2	32×1	32×1	8×2	16×2	16×2	8×2	16×2	8×2	8×2
Wind	16×2	32×1	32×1	8×2	16×2	16×2	8×2	16×2	8×2	8×2
Traffic-CA	8×2	16×1	16×1	4×2	4×2	8×2	4×2	8×2	4×2	4×2
Solar	12×2	24×1	24×1	6×2	6×2	12×2	6×2	12×2	8×2	8×2

F Our Baselines against the Numbers the Benchmark Reports

In Table 1 U-Cast, the method released with the benchmark, trails iTransformer on most cells, which is the reverse of what the benchmark paper reports (Ni et al., 2025). Table A7 places our runs of the two models beside the numbers of that paper to show which side the reversal comes from. The protocols differ. The benchmark sets the horizon S by the sampling interval, searches the lookback in $\{3S, 4S, 5S\}$ and the learning rate in $\{10^{-2}, 10^{-3}, 10^{-4}\}$ per model, and averages five seeds, whereas we fix the lookback, the learning rate and the seed. We therefore compare at the nearest cell of ours,

Table A7: **Our two strongest channel-dependent baselines beside the numbers the benchmark reports.** Left, Table 4 of Ni et al. (2025) at the horizon S of the benchmark; right, our record at the nearest cell, in MSE. Our U-Cast reproduces the reported numbers, and our iTransformer has 4.2 times lower error than the reported one on TRAFFIC-CA.

Dataset	reported by the benchmark			ours, nearest cell			
	S	U-Cast	iTransformer	L	H	U-Cast	iTransformer
Air Quality	28	0.446	0.447	96	24	0.422	0.416
Atec	336	0.287	0.345	720	336	0.295	0.288
Meter	336	0.943	0.949	720	336	0.966	0.959
Temp	168	0.262	0.265	720	192	0.269	0.279
Traffic-CA	168	0.061	0.271	720	192	0.065	0.065

$L = 720$ with the horizon closest to S where we have it; AIR QUALITY is the exception, since its S is shorter than the shortest horizon we run at $L = 720$, so its nearest cell is $L = 96$ with $H = 24$. WIND and SOLAR are not part of the $L = 720$ grid (Section P), and the nearest cell we have for them, $L = 96$ with $H = 96$, is too far from S to compare, so the table leaves them out.

Our U-Cast lands within 7% of the reported one on every dataset in the table, the largest gap being on TRAFFIC-CA, at horizons that differ by up to a seventh. The reversal comes from iTransformer. Ours reaches 0.065 on TRAFFIC-CA where the benchmark reports 0.271, lower by a factor of 4.2, while on the other datasets the two differ by at most 17%. The attention baseline we substitute against is therefore stronger than the one the benchmark was evaluated with on TRAFFIC-CA, not weaker, and close to it on the other datasets.

G TRIVIAL REFERENCE FORECASTS

Table A8: **Trivial reference forecasts, MSE.** Computed on the same test windows as Table 1 in the same pass.

Dataset	Reference	H=12	H=24	H=48	H=96
Air Quality	*zero*	0.790	0.791	0.792	0.795
	persistence	0.672	0.763	0.831	0.877
Atec	*zero*	1.354	1.357	1.361	1.368
	persistence	0.538	0.946	1.636	2.244
Meter	*zero*	1.398	1.398	1.399	1.401
	persistence	2.009	2.354	2.389	2.436
Temp	*zero*	0.956	0.953	0.946	0.934
	persistence	0.413	0.436	0.499	0.572
Wind	*zero*	1.590	1.591	1.592	1.590
	persistence	1.292	1.549	1.812	1.994
Traffic-CA	*zero*	1.017	1.017	1.017	1.017
	persistence	1.843	1.745	1.760	1.774
Solar	*zero*	0.710	0.710	0.710	0.711
	persistence	0.075	0.178	0.438	0.941

Neither reference depends on the model, so each would be constant along a row of Table 1; we collect them here rather than adding two columns to it. *Zero* predicts the training mean and therefore measures how much of the test variance is in the mean; *persistence* holds the last observed value and measures how much is in the most recent sample. Two datasets pull the references furthest apart, in opposite directions. On TRAFFIC-CA persistence is worse than the mean at every horizon (1.84 against 1.02 at $H = 12$), and on SOLAR at $H = 12$ it is 9.5 times better (0.075 against 0.710). They are the two datasets that ask a forecaster the least similar questions, and no single number would have shown it.

H FULL RESULTS AS RAW ERRORS

Table 1 reports the skill score of Equation (8) alone, so that the columns can be read across datasets, and Figure A3 plots the same runs against the horizon. Table A9 carries the same runs as raw Mean Squared Error (MSE) and Mean Absolute Error (MAE), in the form the forecasting literature usually reports.

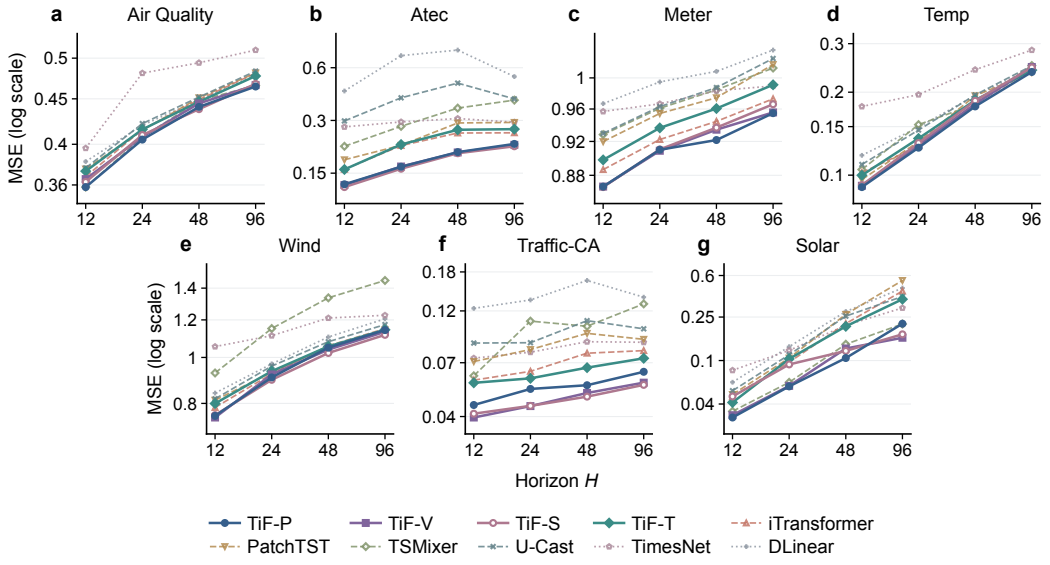

Figure A3: **Error against horizon for every model of Table 1.** MSE against the horizon at $L = 96$, on logarithmic axes, for **(a)** AIR QUALITY, **(b)** ATEC, **(c)** METER, **(d)** TEMP, **(e)** WIND, **(f)** TRAFFIC-CA and **(g)** SOLAR; the four TiF configurations are the solid lines. The margins of Table 1 hold at every horizon rather than at one.

Table A9: **All results at $L = 96$ as MSE and MAE.** Conventions as in Table 1.

Dataset	H	DLinear (2023)		TSMixer (2023)		TimesNet (2023)		PatchTST (2023)		iTransformer (2024a)		U-Cast (2025)		TiF-T (Ours)		TiF-V (Ours)		TiF-S (Ours)		TiF-P (Ours)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Air Quality	12	0.382	0.399	**0.358**	0.387	0.396	0.400	0.375	0.385	0.368	0.376	0.376	0.389	0.373	0.383	0.365	0.374	0.363	0.374	0.358	**0.369**
	24	0.420	0.427	0.407	0.423	0.481	0.449	0.415	0.409	0.416	0.410	0.422	0.416	0.416	0.406	0.408	**0.403**	0.410	0.404	**0.405**	0.404
	48	0.449	0.447	0.448	0.445	0.494	0.455	0.451	0.432	0.449	0.427	0.452	0.432	0.447	0.431	0.445	0.424	**0.438**	0.423	0.441	**0.421**
	96	0.478	0.471	0.466	0.453	0.511	0.465	0.481	0.450	0.481	0.448	0.484	0.452	0.477	0.447	0.466	**0.435**	0.468	0.442	**0.465**	0.438
Atec	12	0.442	0.403	0.214	0.274	0.276	0.248	0.179	0.212	0.158	0.193	0.298	0.308	0.157	0.196	0.128	0.173	**0.125**	**0.169**	0.130	0.173
	24	0.703	0.547	0.278	0.327	0.294	0.256	0.216	0.236	0.216	0.233	0.404	0.375	0.219	0.230	0.164	0.201	**0.159**	**0.198**	0.163	0.204
	48	0.757	0.574	0.353	0.377	0.308	0.263	0.290	0.288	0.254	0.257	0.490	0.440	0.265	0.257	0.197	0.225	**0.195**	**0.223**	0.198	0.224
	96	0.534	0.427	0.392	0.375	0.293	0.264	0.292	0.279	0.255	0.248	0.398	0.355	0.268	0.257	0.218	0.224	**0.214**	**0.223**	0.220	0.234
Meter	12	0.967	0.563	0.928	0.552	0.957	0.555	0.920	0.528	0.887	0.510	0.930	0.536	0.898	0.517	0.867	0.513	**0.866**	**0.504**	0.867	0.506
	24	0.995	0.571	0.961	0.565	0.967	0.556	0.954	0.543	0.922	0.526	0.963	0.556	0.937	0.531	**0.909**	0.528	0.910	0.532	0.910	**0.524**
	48	1.009	0.576	0.985	0.572	0.983	0.561	0.974	0.551	0.944	0.536	0.987	0.559	0.961	0.540	0.934	0.536	0.937	0.538	**0.922**	**0.532**
	96	1.037	0.587	1.013	0.590	0.989	0.563	1.018	0.564	0.973	0.549	1.026	0.572	0.991	0.552	0.956	**0.545**	0.966	0.547	**0.955**	0.547
Temp	12	0.118	0.232	0.105	0.224	0.178	0.303	0.096	0.202	0.102	0.209	0.109	0.219	0.100	0.206	0.091	**0.195**	0.092	0.195	**0.090**	0.196
	24	0.149	0.269	0.152	0.285	0.196	0.321	0.132	0.248	0.128	**0.240**	0.146	0.264	0.136	0.251	0.129	0.243	0.131	0.246	**0.126**	0.242
	48	0.194	0.316	0.181	0.313	0.241	0.361	0.194	0.313	0.182	0.298	0.195	0.315	0.187	0.305	0.181	0.300	0.186	0.303	**0.178**	**0.298**
	96	0.243	0.361	0.250	0.381	0.284	0.396	**0.236**	0.354	0.241	0.356	0.249	0.364	0.240	0.357	0.247	0.361	0.246	0.359	0.237	**0.354**
Wind	12	0.841	0.578	0.927	0.606	1.054	0.656	0.813	0.573	0.784	0.560	0.818	0.574	0.799	0.564	**0.747**	0.540	0.756	0.543	0.753	**0.540**
	24	0.969	0.628	1.151	0.676	1.112	0.686	0.930	0.620	0.923	0.617	0.959	0.628	0.936	0.621	0.921	0.614	**0.897**	**0.606**	0.908	0.609
	48	1.104	0.669	1.336	0.731	1.211	0.719	1.051	0.668	1.058	0.669	1.079	0.680	1.055	0.669	1.039	0.660	**1.022**	**0.657**	1.049	0.663
	96	1.206	**0.695**	1.453	0.787	1.227	0.728	1.144	0.703	1.150	0.706	1.173	0.712	1.144	0.701	1.132	0.702	**1.116**	0.697	1.142	0.702
Traffic-CA	12	0.123	0.224	0.061	0.164	0.073	0.165	0.071	0.173	0.058	0.154	0.086	0.194	0.057	0.160	**0.040**	**0.117**	0.041	0.126	0.045	0.138
	24	0.134	0.238	0.108	0.232	0.078	0.170	0.080	0.182	0.064	0.156	0.086	0.192	0.059	0.159	**0.045**	**0.118**	0.045	0.124	0.053	0.147
	48	0.165	0.269	0.102	0.215	0.087	0.185	0.095	0.205	0.077	0.176	0.108	0.221	0.066	0.165	0.051	0.128	**0.049**	**0.125**	0.055	0.141
	96	0.138	0.232	0.129	0.242	0.087	0.181	0.089	0.188	0.079	0.173	0.100	0.204	0.073	0.165	0.057	0.134	**0.056**	**0.133**	0.064	0.155
Solar	12	0.063	0.153	0.034	0.091	0.081	0.149	0.049	0.104	0.047	0.098	0.053	0.107	0.041	0.095	0.032	0.075	0.047	0.100	**0.030**	**0.074**
	24	0.134	0.249	0.063	0.130	0.124	0.191	0.105	0.149	0.097	0.158	0.113	0.165	0.103	0.187	0.059	**0.110**	0.092	0.147	**0.058**	0.122
	48	0.281	0.394	0.141	0.229	0.210	0.269	0.265	0.254	0.218	0.239	0.255	0.254	0.205	0.273	0.129	0.180	0.123	0.189	**0.105**	**0.172**
	96	0.461	0.537	0.216	0.289	0.303	0.320	0.541	0.410	0.432	0.343	0.385	0.412	0.364	0.365	**0.162**	**0.206**	0.174	0.225	0.217	0.265

I SOLAR, PLANT BY PLANT

SOLAR is the one dataset we use whose files record where each variable is. Each of its 5162 series is a photovoltaic plant whose column name gives its latitude, longitude and capacity, and Figure 4 uses the coordinates to look at the variable-axis substitution of Table A4 one plant at a time, at $L = 96$ and $H = 96$, which on this dataset is eight hours of lookback and eight hours of forecast at a five-minute step. The models are those behind Table 1, drawn from the checkpoints those runs saved; re-scored on the test set, each reproduces its record to the last digit of both the test MSE and the best validation loss.

TiF-V has lower error than iTransformer at 98.1% of the plants, by a median factor of 2.7, with 1.3 at the 5th percentile and 6.7 at the 95th. The two maps also run in opposite directions. The logarithm of the error of TiF-V correlates with latitude at +0.50, so it grows toward the north, while that of iTransformer correlates at −0.66 and grows toward the south. Figure 4c places the difference in the day. It is largest for forecasts that start at 06:00, where iTransformer reaches 2.04 against 0.61, and it reverses only at the 3 starting hours 18:00, 19:00 and 23:00, whose horizons are mostly night.

The morning forecasts are hard for a reason. At 06:00 the lookback is dark at every plant, so nothing in the input series says when the sun will rise, and the calendar features, which both models read through the same one-token-per-variable embedding, are the only cue. That origin is also where the host is weakest rather than typical. On SOLAR iTransformer is the fourth of the six baselines here, at 0.432 against 0.216 for TSMixer, which reads no calendar at all, 0.317 for SOFTS (Section J) and 0.162 for TiF-V. The substitution of Table A4 is against the host by construction; read against the best baseline instead, the lead on this dataset is 8% to 25% (Table 1), and with the calendar removed from both models the error of TiF-V is 0.46 to 0.64 of that of iTransformer (Section M).

J A VARIABLE MIXER LINEAR IN C

Table A10: **Three mixers along the variable axis, MSE at $L = 96$.** Attention (iTransformer), an aggregated core linear in C (SOFTS) and the spatial fluxer layer (TiF-V), each with one token per variable and the protocol of Section 5; bold marks the lowest error in each row.

Dataset	H	SOFTS	iTransformer	TiF-V
Air Quality	12	0.372	0.368	**0.365**
	24	0.413	0.416	**0.408**
	48	0.449	0.449	**0.445**
	96	0.473	0.481	**0.466**
Atec	12	0.153	0.158	**0.128**
	24	0.208	0.216	**0.164**
	48	0.249	0.254	**0.197**
	96	0.254	0.255	**0.218**
Meter	12	0.884	0.887	**0.867**
	24	0.919	0.922	**0.909**
	48	0.941	0.944	**0.934**
	96	0.972	0.973	**0.956**
Temp	12	0.098	0.102	**0.091**
	24	0.133	**0.128**	0.129
	48	0.187	0.182	**0.181**
	96	0.246	**0.241**	0.247
Wind	12	0.792	0.784	**0.747**
	24	0.935	0.923	**0.921**
	48	1.058	1.058	**1.039**
	96	1.145	1.150	**1.132**
Traffic-CA	12	**0.040**	0.058	**0.040**
	24	0.046	0.064	**0.045**
	48	0.054	0.077	**0.051**
	96	0.058	0.079	**0.057**
Solar	12	0.052	0.047	**0.032**
	24	0.106	0.097	**0.059**
	48	0.244	0.218	**0.129**
	96	0.317	0.432	**0.162**

SOFTS (Han et al., 2024) replaces variable-wise attention by one aggregated core through which every series passes, at a cost linear in C, and is the sub-quadratic alternative closest to the spatial fluxer layer. We ran it on the grid of Table 1 under the same protocol, with its default configuration at the width and depth of Table A3. Table A10 sets it beside iTransformer and TiF-V, the three models that mix the variables with one token per variable and differ in the mixer. SOFTS has lower error than iTransformer in 18 of 28 cells, and TiF-V has lower error than SOFTS in 27 of 28, 14 of them

beyond the run-to-run spread and zero of the lost ones, and, once the horizons are averaged, on all seven datasets.

K SERIAL OR PARALLEL

Table A11: **The two wirings of the two stages, cell by cell.** MSE for the serial and parallel wirings of Section 4; **bold** marks the one validation loss would choose, which is not always the one with the lower test error, and Table 1 reports both. The choice is largely consistent within a dataset and differs between datasets.

Dataset	H	Serial	Parallel	Selected
Air Quality	12	0.363	**0.358**	parallel
	24	0.410	**0.405**	parallel
	48	**0.438**	0.441	serial
	96	0.468	**0.465**	parallel
Atec	12	**0.125**	0.130	serial
	24	**0.159**	0.163	serial
	48	0.195	**0.198**	parallel
	96	**0.214**	0.220	serial
Meter	12	0.866	**0.867**	parallel
	24	0.910	**0.910**	parallel
	48	0.937	**0.922**	parallel
	96	0.966	**0.955**	parallel
Temp	12	0.092	**0.090**	parallel
	24	0.131	**0.126**	parallel
	48	0.186	**0.178**	parallel
	96	0.246	**0.237**	parallel
Wind	12	**0.756**	0.753	serial
	24	0.897	**0.908**	parallel
	48	1.022	**1.049**	parallel
	96	**1.116**	1.142	serial
Traffic-CA	12	**0.041**	0.045	serial
	24	**0.045**	0.053	serial
	48	**0.049**	0.055	serial
	96	**0.056**	0.064	serial
Solar	12	0.047	**0.030**	parallel
	24	0.092	**0.058**	parallel
	48	0.123	**0.105**	parallel
	96	**0.174**	0.217	serial

The two stages admit two arrangements, and we report both rather than choosing one. TiF-S runs them in sequence, so that the cross-variable stage sees patches that already carry their own history; TiF-P runs them side by side from the same embedding and joins them with the gate. They share every hyperparameter, every layer count and the same fluxer, and differ only in whether one axis reads the output of the other, which makes Table A11 a clean comparison.

Neither wins outright, and the preference belongs to the dataset rather than to the horizon. Validation loss selects the parallel wiring at every horizon of METER and TEMP and at three of four on AIR QUALITY and SOLAR, and the serial wiring at every horizon of TRAFFIC-CA and at three of four on ATEC; WIND splits evenly. In 21 of the 28 cells the two lie within the run-to-run spread of 0.0122 MSE, so in most cells the arrangement moves the error by no more than a rerun does.

SOLAR, the dataset with the finest sampling in the benchmark, is where the two separate most; the parallel wiring is better by 36%, 37% and 14% at $H = 12$, $H = 24$ and $H = 48$ and worse by 25% at $H = 96$. The margin narrows as the horizon grows and changes sign at the last one, so even the dataset with the widest gap does not settle the arrangement. We therefore carry both wirings; Section 5 counts each on its own, and the per-cell choice that validation loss makes, the last column of Table A11, is reported there beside those counts and not in place of them.

L THE CLOSURE OF THE TIME AXIS

Proposition 1 is a statement about the operator, not about accuracy. This section measures the accuracy twice, because the answer differs between the two ways of training. Under the protocol of Section 5 the padding shows no consistent effect on the test error; under the training of a decoder-style forecaster it is what lets one model serve any history within the window. Figure 5 is the summary, and what follows is every dataset on its own.

Table A12: **Removing the twofold padding from the time axis.** TiF-T with the circular transform minus TiF-T as reported, in MSE; positive means the circular variant is worse. The run-to-run spread is 0.0122 at $L = 96$ and 0.0293 at $L = 720$, and no entry exceeds it. The $L = 720$ rows are the datasets of Table A17 on which TiF-T fits an accelerator.

Lookback	Dataset	ΔMSE (circular − causal)			
		$H = 12$	$H = 24$	$H = 48$	$H = 96$
$L = 96$	Air Quality	$+0.0009$	-0.0017	-0.0004	-0.0010
	Atec	-0.0013	$+0.0091$	-0.0116	-0.0086
	Meter	-0.0012	$+0.0009$	$+0.0003$	$+0.0028$
	Temp	$+0.0001$	$+0.0003$	$+0.0003$	-0.0000
	Wind	-0.0019	$+0.0023$	$+0.0040$	$+0.0061$
	Traffic-CA	-0.0052	$+0.0015$	$+0.0003$	-0.0050
	Solar	-0.0024	-0.0065	$+0.0020$	-0.0096
		$H = 96$	$H = 192$	$H = 336$	$H = 720$
$L = 720$	Air Quality	-0.0001	$+0.0002$	-0.0030	$+0.0020$
	Atec	$+0.0220$	-0.0080	$+0.0080$	$+0.0057$
	Meter	-0.0025	-0.0009	-0.0030	$+0.0083$
	Temp	-0.0017	-0.0069	$+0.0012$	$+0.0039$

Under the flattened head We take TiF-T, which is the time axis on its own, and remove the twofold zero-padding from its transform, leaving the circular convolution of Equation (1); the seed, the batch size, the schedule and every other setting are those of the cell it is compared against. Table A12 gives the change at both lookbacks. At $L = 96$ the circular variant is worse in 14 of 28 cells with a median change of $+0.0000$ MSE; at $L = 720$, where each variable carries eighty-nine patches instead of eleven so the wrap-around reaches correspondingly further, it is worse in 8 of 16 with a median of $+0.0001$. Read against the run-to-run spread of the same lookback, 0.0122 from 12 repeat pairs at $L = 96$ and 0.0293 from 8 at $L = 720$, no cell moves beyond it in either direction. At $L = 96$ the spread is the largest gap between the seed-42 run of a cell and a rerun with another seed, over the 12 pairs of TiF-V and iTransformer on AIR QUALITY, ATEC and TRAFFIC-CA at $H = 96$; relative to the MSE it is 5.6% on ATEC and at most 2.7% elsewhere, so a single absolute value is lenient on the datasets with the smallest errors. We use it as a yardstick for single differences, not as a test, and a difference inside it does not show that two models are equal. The substitution of Table A4 is untouched as well, since TiF-T takes 21 of 28 cells from PatchTST with the padding and 21 without it.

The head of Section 4 is the reason to expect as much. It flattens every patch of the lookback and maps them jointly to the horizon, so every patch precedes the forecast and an operator that lets patch t read patch $t + k$ reaches nothing the head did not already hold. The two encoders are still different functions, since a stack of circular layers mixes the window in an order a causal stack cannot reproduce; what the protocol does not do is put that difference under load, and in this one-seed grid it does not show up beyond the spread.

One model for histories within the window A forecaster that flattens its lookback is trained for one history length. Decoder-style forecasters avoid that by training every position rather than the last one (Das et al., 2024; Liu et al., 2024b); each patch predicts the steps that follow it, the loss runs over all positions, and at inference the forecast is read from the position at the end of whatever history is available. That training is valid only if no position reads a later one, which is what the causal closure guarantees. We train TiF-T that way at $L = 96$ and $H = 96$ on seven datasets and evaluate it on histories shorter than the training window, padded on the right to the trained length rather than truncated, since with the orthonormal transforms of Equation (5) the

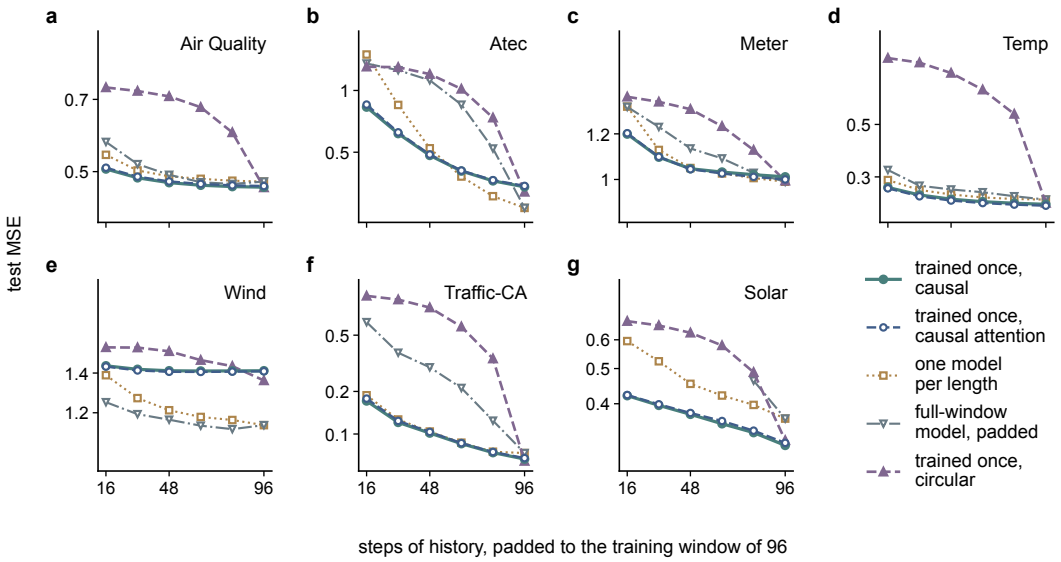

Figure A4: **One model for histories within the window, dataset by dataset.** Test MSE at $L = 96$ and $H = 96$ against the number of leading steps of the window that are real data, the rest being padded. *Trained once*: every position forecasts the steps that follow it, under the causal closure of Proposition 1 or the circular one, which differ in nothing else. *One model per length*: TiF-T with the flattened head of Section 4, trained at that lookback, one model per point. *Causal attention*: the same training with causal multi-head attention of equal size in place of the fluxer, on all seven datasets. *Full-window model*: the TiF-T of Table 1 handed the last steps of the window, the missing ones filled in by whichever of four rules gives it the lowest error, and not drawn where it does worse than the better trivial forecast (SOLAR, short histories; Table A13 has the values). Trained once under the causal closure, TiF-T forecasts from any history within the window at the level of the models trained separately for each length; under the circular closure it does not. At the full window the two arms come together, since the forecast is read at the last position, whose target lies wholly after the window and so beyond the reach of either stack.

convolution carries a factor $1/\sqrt{2n}$ and a shorter transform would rescale it. By Proposition 1 the forecast at a position does not depend on what follows it, and the code agrees; padded and in-window outputs agree to within 8×10^{-4} on every dataset except SOLAR, where they differ by up to 2×10^{-2}, and run in double precision the same check agrees to machine precision on all of them, which places the residual in single-precision arithmetic rather than in the operator (`scripts/test_decoder_causality.py` in the code release).

Figure A4 and Table A13 set that one model against the alternatives. The first is one model per length, TiF-T with the flattened head trained separately at lookbacks of 16 to 80 steps and scored on the same forecast origins, except the $96-c$ at either end of the test range that a history of c steps displaces. Over the 35 pairs of dataset and shorter history, the single model has the lower error in 26, with a ratio to the dedicated model between 0.55 and 1.21 and a median of 0.97; on WIND the per-length models win at every length. At the full window, where the flattened head reads all patches jointly, the single model is the better of the two on 4 of 7 datasets and behind by up to 27% on ATEC, which is why Table 1 keeps the flattened head. Training once is also cheaper. On the same two accelerators and budget, the single model took between $1/13$ and $1/6$ of the wall-clock time of the five per-length models and the full-window one together, a median of $1/8$ (`sources/context/train_cost.json` in the code release). The saving comes from the training, which any causal mixer admits, not from the operator.

The second alternative needs no training. The TiF-T of Table 1, trained on full windows, is handed the last c steps with the missing ones filled in by the training mean, by the earliest observed value or by the mean of the observed part, the last also with its instance normalization computed on the observed part alone, and we keep whichever filling gives it the lowest error at each length. The single model has the lower error in 30 of 35 pairs, by a median factor of 1.1 that reaches 3.6 on TRAFFIC-CA; on WIND, where the flattened head is far ahead at the full window, the filled-in model stays ahead at every length. In four pairs, all on SOLAR with 64 steps of history or fewer, the filled-in model does worse than the better trivial forecast under every filling. We keep those pairs in the counts and in Table A13 but leave them out of the plots and of the largest factor, which would otherwise measure a failure of the reference rather than an advantage of ours.

Table A13: **One causal model against one model per history length.** Test MSE of TiF-T at $H = 96$ by the number of steps of history, the rows as in Figure A4; the last column of the per-length row is the TiF-T of Table 1. Bold marks the lowest error in each column of a dataset.

Dataset	Model	steps of history					
		16	32	48	64	80	96
Air Quality	trained once, causal	**0.506**	**0.486**	**0.475**	**0.469**	**0.466**	0.465
	trained once, causal attention	0.509	0.488	0.478	0.472	0.469	0.468
	one model per length	0.541	0.503	0.489	0.484	0.479	0.477
	full-window model, padded	0.574	0.518	0.493	0.476	0.472	0.477
	trained once, circular	0.740	0.728	0.710	0.675	0.600	**0.464**
Atec	trained once, causal	**0.830**	**0.616**	**0.482**	0.405	0.363	0.339
	trained once, causal attention	0.854	0.625	0.487	0.407	0.365	0.343
	one model per length	1.497	0.850	0.523	**0.381**	**0.305**	**0.268**
	full-window model, padded	1.353	1.255	1.122	0.853	0.521	0.268
	trained once, circular	1.305	1.298	1.202	1.018	0.737	0.320
Meter	trained once, causal	**1.199**	**1.093**	1.044	1.030	1.019	1.011
	trained once, causal attention	1.203	1.095	**1.042**	**1.024**	1.011	1.000
	one model per length	1.338	1.125	1.047	1.024	**1.006**	0.991
	full-window model, padded	1.337	1.234	1.132	1.088	1.027	**0.991**
	trained once, circular	1.392	1.364	1.324	1.237	1.125	0.995
Temp	trained once, causal	0.271	0.252	0.241	0.236	0.232	0.230
	trained once, causal attention	**0.268**	**0.248**	**0.238**	**0.232**	**0.229**	**0.227**
	one model per length	0.291	0.264	0.253	0.246	0.242	0.240
	full-window model, padded	0.321	0.275	0.265	0.258	0.249	0.240
	trained once, circular	0.954	0.913	0.825	0.701	0.554	0.233
Wind	trained once, causal	1.440	1.420	1.413	1.411	1.412	1.413
	trained once, causal attention	1.436	1.414	1.407	1.406	1.408	1.409
	one model per length	1.388	1.270	1.212	1.181	1.167	1.144
	full-window model, padded	**1.250**	**1.194**	**1.168**	**1.141**	**1.126**	**1.144**
	trained once, circular	1.547	1.545	1.523	1.472	1.438	1.360
Traffic-CA	trained once, causal	**0.171**	**0.121**	**0.102**	**0.085**	**0.074**	0.067
	trained once, causal attention	0.178	0.124	0.103	0.086	0.075	0.068
	one model per length	0.188	0.127	0.104	0.087	0.075	0.073
	full-window model, padded	0.618	0.376	0.296	0.212	0.124	0.073
	trained once, circular	0.948	0.892	0.784	0.574	0.341	**0.064**
Solar	trained once, causal	**0.421**	**0.396**	**0.374**	**0.352**	**0.333**	**0.307**
	trained once, causal attention	0.423	0.399	0.377	0.359	0.337	0.312
	one model per length	0.595	0.524	0.454	0.421	0.397	0.364
	full-window model, padded	11.820	11.461	2.062	0.745	0.463	0.364
	trained once, circular	0.675	0.657	0.627	0.579	0.488	0.317

The third alternative keeps the training and the closure and replaces the mixer. The three temporal layers use causal multi-head attention with eight heads, as in the PatchTST configuration, with query, key, value and output projections that hold the same $4D^2$ weights as the input and output projections of the fluxer, and everything else unchanged. Run on all seven datasets, it has the higher error in 26 of 42 pairs of dataset and history, by a factor between 0.99 and 1.04 with a median of 1.01, and the difference exceeds the run-to-run spread in one of them. Under this training the two causal operators therefore land within a factor of 1.04 of each other; the fluxer does not buy the capability, and it carries it about as well as attention does.

The last alternative is the same position-by-position training under the circular closure. It is worse at every history shorter than the window, in all 35 cells, by a median factor of 1.6 that reaches 7.7 on TRAFFIC-CA. The reason is visible in the training loss, which the circular arm drives far below the causal one, since a position whose transform wraps reads the patches after it, and those patches are its own target. At the full window the two arms come together. The forecast is then read at the last position, whose target lies wholly after the window, so no layer of either stack can reach any part of it and the leak has nothing to feed on there. The two remain different functions, since Proposition 1 equates one circular and one causal layer only in their last row and the upper layers of the circular stack read positions that did wrap, and what the table reports is how far apart they land.

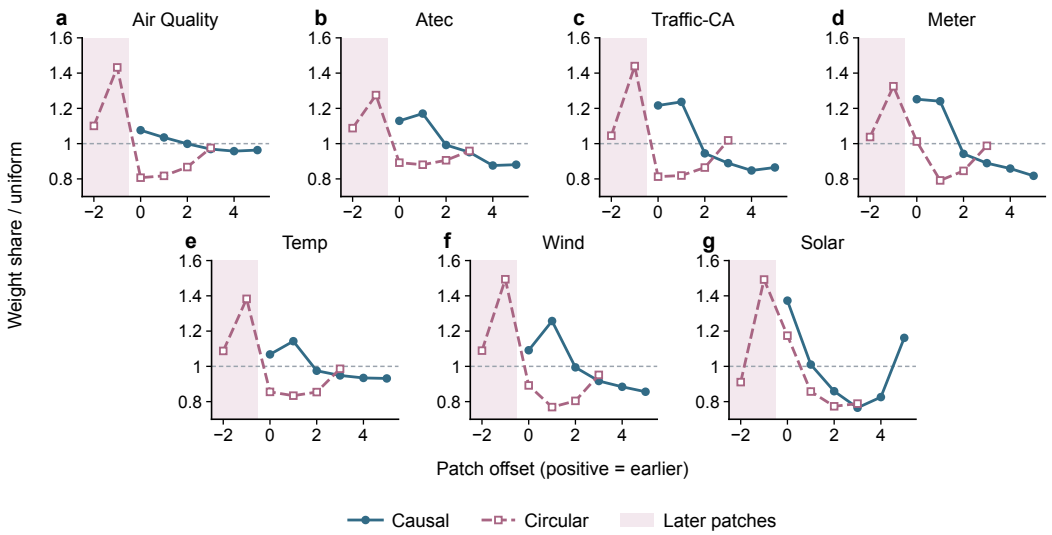

Figure A5: **Mixing weight by kernel tap under the two closures of the time axis.** The share of the mixing weight the trained fluxer applies at each tap of its kernel, divided by the share a flat kernel would give, averaged over the test set and the three layers of TiF-T trained at every position at $L = 96$ and $H = 96$; the window holds six patches. A causal tap d reads d patches back in the rows $t \geqslant d$ and is not used in the others. A circular tap d reads d patches back in the rows $t \geqslant d$ and wraps to $n - d$ patches ahead in the d rows $t < d$, and each is drawn at the patch it reads in most rows, positive being earlier; the shaded band holds the taps that read a later patch in most rows, where the causal transform has no weight at all. Since k is read from the tokens, the fluxer places weight on a tap through the activation at the corresponding index of the window, and what is plotted is the weight the convolution applies. The circular arm spends its largest weight on the tap that reads the next patch, part of the target it is trying to forecast, in every row but the last. The two arms are those of Figure A4.

Figure A5 measures the leak rather than inferring it. The fluxer applies the weight $k_{(t-s) \bmod n}$ to position s when it produces position t. Under the causal closure tap d reads d patches back in the rows $t \geqslant d$ and is not used in the others; under the circular closure it reads d patches back in the rows $t \geqslant d$ and wraps to $n - d$ patches ahead in the d rows $t < d$, so what a tap reaches depends on the row, and the tap $d = n - 1$ reads the next patch in every row but the last. Averaged over the test set and the three layers, the circular arm places its largest weight on that tap in all seven datasets, 23.9% of the weight at the median and from 21.3% to 24.9%, against 16.7% for a flat kernel. Counted row by row, the entries of the circulant that read a later patch carry 46.6% of its weight, from 44.1% to 47.4% across datasets, against 41.7% for a flat kernel, whose share $(n - 1)/2n$ is already large because the wrap reaches ahead in every row but the last. It takes that weight from the recent past; taps 0 and 1, which read the current patch and, in every row but the first, the previous one, hold 39% of the weight under the causal closure and 28% under the circular one, where a flat kernel would give 33%. The causal arm cannot reach the later positions at all, and its largest weight falls on the current or the previous patch in all seven datasets. These are weights of the convolution rather than a measure of what the trained network does with them; the error of Figure A4 is that measure.

M The Variable Axis, Part by Part

TiF-V is the variable axis on its own, and here we change it one part at a time, the closure of its transform, the gate $\mathbf{v}$ that multiplies the output of Equation (5), and the kernel itself. Each changed model is trained with the seed, batch size, schedule and every other setting of the cell it is compared against, and Table A14 gives the changed model minus its reference in MSE, so a positive entry means the change made the model worse. The run-to-run spread at this lookback is 0.0122, and Figure A6 draws the first of the three cell by cell.

The closure The closure of the variable axis can be exchanged in two ways. One-sided padding gives it the closure of the time axis, the transform padded to $2C$ and truncated, so that the mixing matrix is lower-triangular and each variable reads only the variables stored before it. Symmetric padding keeps both directions and removes only the wrap, which leaves the circulant with every entry

Table A14: **TiF-V with one part changed at a time.** TiF-V at $L = 96$ with one part changed, minus its reference, in MSE; the median over four horizons and the number of horizons at which the change is worse. The padded arms, run without the calendar tokens and the static kernel, use a circular reference with the same tokens; the other arms use TiF-V as reported. The two kernel-removal arms were run on ATEC, METER, WIND and TEMP, a four-dataset subset fixed before any ablation outcome was observed, and the mean arm on ATEC and SOLAR. The run-to-run spread is 0.0122, and a dash marks a cell not run. With the tokens held equal, neither the closure nor the gate moves the test error consistently, whereas removing either part of the kernel usually increases it.

Dataset	one-sided padding		symmetric padding		no gate		no static kernel		static kernel only		mean of the tokens	
	median Δ	worse	median Δ	worse	median Δ	worse	median Δ	worse	median Δ	worse	median Δ	worse
Air Quality	–	–	–	–	+0.0011	2 of 4	–	–	–	–	–	–
Atec	+0.0037	3 of 4	−0.0009	1 of 4	+0.0079	3 of 4	+0.0168	4 of 4	+0.0081	3 of 4	+0.0423	4 of 4
Meter	–	–	–	–	−0.0056	1 of 4	+0.0155	4 of 4	+0.0092	4 of 4	–	–
Temp	–	–	–	–	−0.0020	0 of 4	+0.0073	3 of 4	−0.0016	2 of 4	–	–
Wind	–	–	–	–	+0.0021	2 of 4	+0.0125	4 of 4	+0.0210	3 of 4	–	–
Traffic-CA	–	–	–	–	+0.0006	3 of 4	–	–	–	–	–	–
Solar	−0.0018	0 of 4	−0.0002	2 of 4	+0.0033	2 of 4	–	–	–	–	+0.0020	3 of 4
All	−0.0010	3 of 8	−0.0003	3 of 8	−0.0002	13 of 28	+0.0125	15 of 16	+0.0068	12 of 16	+0.0222	7 of 8

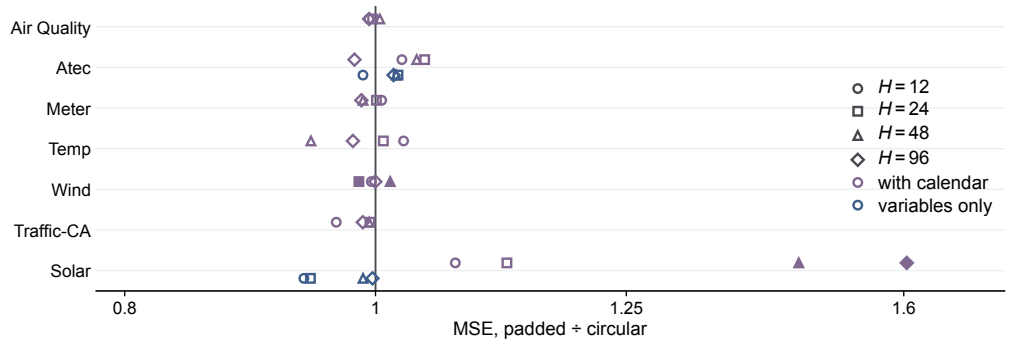

Figure A6: **Padding the variable axis, with and without the calendar tokens on it.** Test MSE of TiF-V with a one-sided padded variable axis divided by that with the circular one, one marker per horizon, filled beyond the run-to-run spread. Upper row of each dataset: the model as reported, with the calendar tokens on the axis and stored first in both arms. Lower row: the variables alone and without the static kernel, on ATEC and SOLAR; Section M repeats it with the static kernel kept. Padding costs SOLAR up to 60% in the model as reported and shows no consistent cost once the calendar tokens are off the axis.

at a distance of $C/2$ or more set to zero. All three arms of this comparison drop the calendar tokens and the static kernel, so they read the same tokens and differ in the closure alone; we run them on ATEC and SOLAR. One-sided padding is worse in 3 of 8 cells with a median change of -0.0010, and crosses the spread in zero against it and zero for it; symmetric padding is worse in 3 of 8 with a median of -0.0003, and crosses it in zero and zero. With the tokens held equal, this one-seed grid shows no consistent difference between the three closures. The same holds with the static kernel kept: on the six datasets where we ran the three arms with it, one-sided padding is worse in 14 of 24 cells with a median change of $+0.0008$ and crosses the spread in one cell against it and zero for it, and symmetric padding is worse in 15 of 24 with a median of $+0.0004$, crossing it in zero and zero.

The model as reported does tell them apart, on one dataset, and Figure A6 draws that comparison beside the one above. TiF appends its four calendar features to the variables as tokens, and under one-sided padding a token stored last is read by no variable, so both arms of this second comparison store the calendar tokens first. The move shows no consistent cost on its own; the circular arm with the calendar first is worse than TiF-V as reported in 12 of 28 cells, with a median change of -0.0003, and beyond the spread in one. Padding is then worse in 13 of 28 cells with a median change of -0.0002; on six of the seven datasets the median stays inside the spread, and outside SOLAR it crosses the spread in one cell against it and one for it. On SOLAR the padded arm is worse at every horizon, by 7% at $H = 12$ and by more at each longer horizon, up to 60% at $H = 96$, where the two circular runs reach 0.1616 and 0.1621 and the padded arm 0.2601, a gap 8 times the spread. The SOLAR row of Table A14 says what that gap is not, since with the calendar tokens off the axis both padded arms sit inside the spread there as well, and so does the one-sided arm with the static kernel kept, which moves SOLAR by -2.8% to $+0.5\%$. That pair differs from the upper pair of Figure A6 in the calendar tokens and in one switch besides: its padded arm rescales the padded transform to the

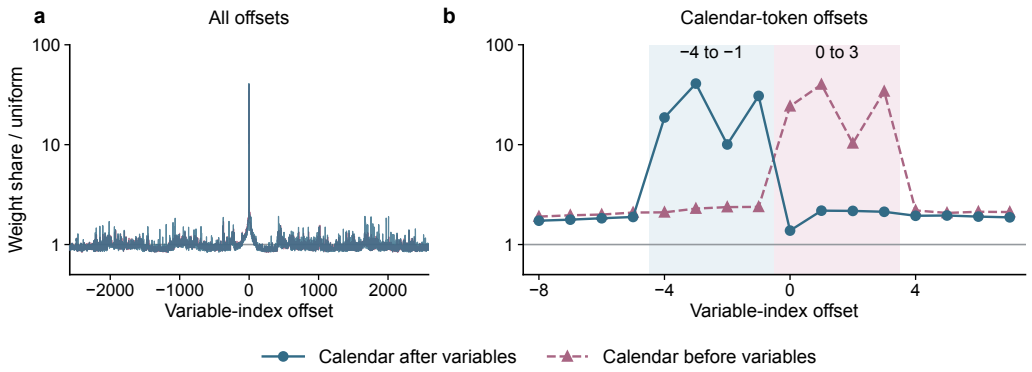

Figure A7: **Mixing weight by offset across variables on SOLAR.** The share of the mixing weight that the spatial layers of TiF-V apply at each offset between the variable read and the variable produced, divided by the share of a flat kernel and averaged over the test set of SOLAR at $L = 96$ and $H = 96$ and over the three layers. The two curves are the same configuration with the 5162 variables stored before and after the four calendar tokens. **(a)** All 5166 offsets; the axis is circular, so offsets are drawn from $-5166/2$ to $5166/2$. **(b)** The offsets nearest zero. The mixing is close to flat except at the four offsets whose taps are read from the calendar tokens, and those taps move when the tokens do.

norm of the circular one, which the upper pair does not, but a constant factor on $\mathbf{y}$ is removed by the LayerNorm that follows it, up to its ϵ. The difference on SOLAR therefore belongs to the calendar tokens on a padded axis and not to the closure of the variables or to the static kernel, and we count it as evidence for the circular closure in one respect only: under the ring, where the calendar tokens are stored made no consistent difference in these runs, whereas under one-sided padding a token stored last is read by no variable.

The gate Removing the third projection $\mathbf{v}$ and the modulation it applies is worse in 13 of 28 cells with a median change of -0.0002, and it crosses the spread in one cell against it and two for it. The gate is a third of the input projection and shows no consistent effect on the test error at this lookback.

The kernel Three arms change where the kernel comes from. The first two remove one part at a time and were run on ATEC, METER, WIND and TEMP, a four-dataset subset fixed before any ablation outcome was observed; the other three datasets were not run for this comparison. Without the static part, 15 of 16 cells are worse, with median change $+0.0125$; separately, the all-dataset parameter-matched substitution of Table A4, which also has no static kernel, still takes 21 of 28 cells from iTransformer. Keeping only the static kernel applies one learned circular filter to every window, as a Fourier Neural Operator (FNO) does (Li et al., 2021); it is worse in 12 of 16 cells with median change $+0.0068$. Replacing the convolution by the token mean gives every pair the same weight; on the two datasets run, it is worse in 7 of 8 cells. Both parts of the kernel therefore earn their place, and a flat aggregator, which weights every pair alike, loses what the learned filter buys.

Where the calendar enters Figure A7 shows what the variable axis learns. Averaged over the test set and the three layers, the kernel of TiF-V on SOLAR gives no offset more than 2.4 times the share of a flat kernel except four, and on those four of its 5166 offsets it places 1.9% of its mixing weight, 25 times their share under a flat kernel, and 4.4% in the first layer alone. The kernel is read from the tokens, so the tap at offset d takes its value from the token stored at index d, and these four are the indices of the calendar tokens, -4 to -1 of the circular axis when they are stored after the variables, as in the model we report, and 0 to 3 when they are stored before them, where the four largest taps move with them and take 2.1%. What the calendar sets, then, is not a weight on itself but how strongly each variable mixes with the variables stored next to it, and that coupling changes with the calendar. The accuracy shows no consistent difference between the two placements, within the spread in all but one cell. Nor is the calendar where the lead over iTransformer on SOLAR comes from; with the calendar tokens removed from both models, the error of TiF-V is 0.46 to 0.64 of that of iTransformer over the four horizons, against 0.37 to 0.68 with them.

N ORDER ON AN UNORDERED AXIS

Table A15: **Test error under three random relabelings of the variables.** MSE under the original variable order and under three random permutations π_1, π_2, π_3, each applied identically to the training, validation and test segments. The last two columns give the standard deviation over the four orderings and, for comparison, over three training seeds at the original order. iTransformer is the reference, since it embeds each variable as a token with no positional encoding and a shared head, so that its operator is permutation-equivariant and only its dropout, whose mask is drawn by token position, can respond to a reordering.

Cell	Model	original	π_1	π_2	π_3	spread(π)	spread(seed)
AIR QUALITY L=96 H=96	TiF-V	0.4663	0.4703	0.4683	0.4653	0.0019	0.0008
	iTransformer	0.4810	0.4771	0.4756	0.4783	0.0020	0.0025
ATEC L=96 H=96	TiF-V	0.2177	0.2132	0.2097	0.2096	0.0033	0.0054
	iTransformer	0.2550	0.2496	0.2465	0.2467	0.0034	0.0043
TRAFFIC-CA L=96 H=96	TiF-V	0.0569	0.0582	0.0585	0.0584	0.0007	0.0003
	iTransformer	0.0794	0.0799	0.0795	0.0796	0.0002	0.0009

An operator that reads an order the axis does not have The circulant across variables couples two indices through their difference, so unlike attention it is not equivariant to relabeling the variables. On an axis whose order is arbitrary that is the obvious objection to the design, and Table A15 measures what it costs, by training and testing TiF-V and its host under three random permutations of the variables, each applied identically to every split.

What the released order contains The permutation destroys something, and it is worth saying what. Wherever the released files let us check, the order the benchmark ships is far from arbitrary. In TRAFFIC-CA consecutive columns are 0.8 km apart at the median and 94% of them lie within 5 km of each other, against 272 km for a random pair, because the sensors are stored along the freeways they sit on; the coordinates come from the metadata of LARGEST (Liu et al., 2023a) and are used here and nowhere else, never as an input. In AIR QUALITY 86% of consecutive columns are two pollutants at the same one of 158 cities. In SOLAR the effect is weaker, 177 km between neighbors against 1305 km at random. An operator that reads index differences therefore starts with an index that means something on these datasets, which is the reason to ask what happens when it stops meaning anything. On TRAFFIC-CA the meaning turns out to be worth little; across the three permutations the worst costs TiF-V 3%, against the 28% it leads iTransformer by at the original order, so the permuted model still leads by 26%. On this dataset the fluxer is not living off the spatial ordering it was handed.

What a relabeling costs Read the reference row of Table A15 first. For iTransformer the spread across orderings is no larger than the spread across seeds, 0.41% against 1.13%, which is what equivariance predicts and also checks that the permutation is not leaking into the split or the standardizer, since a relabeling costs that host no more than re-running it. Ours is larger. We move by 1.16% across orderings and by 0.61% across seeds, so a relabeling costs us about two re-runs, and that gap is the order dependence the structure predicts. What it is not is large in absolute terms, 1.16% against the 0.41% of a host that cannot see the order at all. What settles whether it matters is not the spread but the margin it would have to erase. In every cell the swing across orderings is between 3 and 14 times smaller than our lead over iTransformer at the original order, and the lead survives in 12 of the 12 combinations of cell and ordering that we ran. That is as far as the test reaches: it covers the datasets and the single horizon of Table A15, each ordering trained from scratch, and not variables added after training or a trained model whose variables are reordered.

O COST AT SCALE

The mixing step alone Figure 6a times one training step of each mixing operation in its own process, at width 512, eight heads and batch four, after a warm-up at the sizes measured and as the median of repeated runs. The projections that attention and the fluxer both pay lie outside the timed region. Every operator runs in single precision, since the transform library has no `bfloat16` transform and a half-precision one only at power-of-two lengths, which the $C + 4$ of the datasets are not; single precision is therefore the setting both operators share, and fused attention then runs through its memory-efficient kernel. At lengths that factor into small primes the circular fluxer is 2.5

times faster than fused attention at 1024 tokens and 17, 46, 77 and 268 times faster at 7500, 20,250, 32,768 and 131,072 tokens. The padded transform of the time axis takes 2.0 to 2.3 times as long as the circular one.

The lengths the model transforms are $C + 4$, and two of the lengths drawn carry a large prime factor, $7494 = 2 \cdot 3 \cdot 1249$ on TRAFFIC-CA and $20{,}004 = 2^2 \cdot 3 \cdot 1667$ at $C = 20{,}000$, where the transform library falls back to a slower algorithm. The step is 2.1 and 2.4 times slower there than at the neighboring lengths 7500 and 20,250, which leaves the fluxer 8.0 and 18.3 times faster than fused attention there. A ring padded to the neighboring length would run at the faster rate, but it is a different model from the one we trained, and we report the one we trained. With attention in `bfloat16` through FLASHATTENTION, its fastest setting, and the transform left in single precision, fused attention is the faster of the two below 4096 tokens and the fluxer above, 27 times faster at 131,072.

The whole model Figure 6b and c time one training step of each model at the lookback and width of the main grid, in single precision, at batch four or the largest of two and one that fits. Under `bfloat16` autocast, which runs fused attention through FLASHATTENTION while the transform stays in single precision, the whole-model gap all but closes: fused iTransformer takes 0.88 to 1.09 times as long as TiF-V on the seven datasets, 9.1 against 8.4 ms per sample on TRAFFIC-CA. Along time the fluxer is not cheaper than attention; in single precision TiF-T runs 113 ms per sample on TRAFFIC-CA against 89 for PatchTST at the same tokenization. The advantage of $n \log n$ over n^2 is asymptotic in the token count, and the time axis carries eleven patches per variable, padded to twenty-two, whereas the lengths of Figure 6a, from 1024 up, are the ones the variable axis transforms, and nothing there speaks for the time axis. Attention over the twelve tokens of PatchTST is a small fused product, while the transform pays its own cost at any length.

Table A16: **Peak memory in GiB for one training step at batch one, at the lookback and width the paper reports, as the variable count grows.** Synthetic data; OOM marks an allocation that failed on an 80 GB accelerator, and the last row gives the largest C each method reaches. The scan runs to 100,000 variables, well past the largest dataset here, so that where a method stops is a point on a curve rather than an artifact of where the datasets end. TiF-V keeps the embedding and depth of iTransformer and replaces only its attention with the spatial fluxer layer; that is the pair in which the constants are comparable, and the one that separates the cost of the mixer from the cost of a temporal axis. iTransformer (fused) is the same model with its attention computed by a fused kernel that never stores the $C \times C$ scores; its memory is linear in C and below that of TiF-V at every count.

C	DLinear	TSMixer	TimesNet	PatchTST	iTransformer	iTransformer (fused)	U-Cast	TiF-S	TiF-V
1000	0.07	0.10	3.72	1.07	0.39	0.16	0.12	1.15	0.20
2000	0.07	0.13	3.73	2.02	1.21	0.23	0.15	2.18	0.32
4000	0.07	0.19	3.75	3.96	4.36	0.37	0.20	4.21	0.55
8000	0.08	0.31	3.80	7.84	16.74	0.66	0.34	8.26	1.01
12,000	0.09	0.43	3.84	11.71	37.23	1.06	0.52	12.73	1.50
16,000	0.10	0.56	3.89	15.59	65.81	1.34	0.73	16.93	1.97
20,000	0.11	0.68	3.94	19.43	OOM	1.62	0.98	21.12	2.44
32,000	0.14	1.03	4.21	31.03	OOM	2.43	2.06	32.49	3.70
50,000	0.19	1.57	4.63	48.44	OOM	3.68	4.48	52.63	5.94
100,000	0.31	3.08	5.81	OOM	OOM	7.15	16.17	OOM	11.79
largest C	100,000	100,000	100,000	50,000	16,000	100,000	100,000	50,000	100,000

Memory Table A16 gives the peak memory of one training step at batch one on synthetic data, at the lookback and width of the main grid, with the variable count raised until the allocation fails. Attention written out reaches 16,000 variables at 65.8 GiB, 33 times the memory of TiF-V at the same count, and fails at the next; the fused kernel removes that wall, and its memory stays below that of TiF-V at every count. The two-axis TiF-S stops at 50,000 variables, where it holds 9 times the memory of TiF-V.

P A LONGER LOOKBACK

The main text reports one lookback. We also ran the grid at $L = 720$ with $H \in \{96, 192, 336, 720\}$ under the budget of Section 5 on five of the seven datasets, and Table A17 gives the skill scores; SOLAR and WIND are not part of this grid.

Table A17: **Skill against the better trivial forecast at $L = 720$, four horizons on five datasets.** $\mathcal{S}$ of Equation (8), with the protocol of Section 5 except for the lookback and the horizons; the best entry in each row is **shaded and bold** and the second best shaded and underlined. OOM marks a model that does not fit an 80 GB accelerator at a batch of one per device.

Dataset	H	DLinear (2023)	TSMixer (2023)	TimesNet (2023)	PatchTST (2023)	iTransformer (2024a)	U-Cast (2025)	TiF-T (Ours)	TiF-V (Ours)	TiF-S (Ours)	TiF-P (Ours)
Air Quality	96	0.391	0.411	0.398	0.383	0.413	0.394	0.399	0.413	**0.427**	0.410
	192	0.339	0.386	0.394	0.336	0.399	0.366	0.361	**0.413**	0.411	0.387
	336	0.289	0.374	0.393	0.309	0.396	0.330	0.326	**0.419**	0.404	0.397
	720	0.270	0.380	0.425	0.335	0.382	0.341	0.351	**0.429**	0.399	0.408
Atec	96	0.829	0.775	0.766	0.828	0.847	0.831	0.851	**0.852**	0.841	0.848
	192	0.804	0.719	0.746	0.783	0.800	0.812	0.816	0.825	**0.829**	0.827
	336	0.778	0.712	0.697	0.784	0.792	0.787	0.791	0.795	**0.805**	0.789
	720	0.715	0.656	0.690	0.719	0.747	0.732	0.752	0.751	0.757	**0.763**
Meter	96	0.341	0.305	0.265	0.349	0.356	0.340	0.363	0.358	0.364	**0.365**
	192	0.329	0.258	0.248	0.327	**0.348**	0.324	0.346	0.344	0.346	0.346
	336	0.320	0.253	0.272	0.319	0.319	0.314	0.332	0.330	0.333	**0.334**
	720	0.299	0.239	0.261	0.295	0.287	0.289	**0.309**	0.299	0.304	0.305
Temp	96	0.585	0.585	0.509	0.582	0.585	0.577	0.578	0.572	0.585	**0.594**
	192	0.533	0.537	0.539	0.575	0.563	**0.579**	0.528	0.577	0.567	0.577
	336	0.522	0.538	0.576	0.517	0.550	0.556	0.532	**0.577**	0.559	0.567
	720	0.480	**0.636**	0.593	0.438	0.537	0.405	0.495	0.558	0.569	0.572
Traffic-CA	96	0.942	0.901	0.912	0.949	0.940	0.938	OOM	0.953	**0.954**	0.945
	192	0.938	0.906	0.907	0.936	0.937	0.936	OOM	**0.945**	0.937	0.941
	336	0.934	0.906	0.900	0.931	0.932	0.929	OOM	**0.939**	0.935	0.937
	720	0.923	0.883	0.893	0.921	0.918	0.920	OOM	**0.930**	0.927	0.928

Our variants lead 17 of the 20 cells. The three cells we do not take are METER at $H = 192$, TEMP at $H = 192$ and TEMP at $H = 720$, where iTransformer, U-Cast and TSMixer lead by 0.002, 0.001 and 0.051 MSE respectively; the run-to-run spread at this lookback is 0.0293 (Section L), which covers two of the three. TiF-T has no entry on TRAFFIC-CA, where 7490 variables at eighty-nine patches each exceed the accelerator before the first training step completes.

Q REPRODUCING THE NUMBERS

Every entry in Table A9 is produced by one run of the pipeline and is stored as a JSON record under `sources/results/`, containing the model, dataset, lookback, horizon, seed, parameter count, the number of accelerators and the batch size on each (the pipeline scales neither the learning rate nor the loss with the number of accelerators, so their product is the batch the optimizer sees), the number of epochs run, the reported MSE and MAE, and the two trivial baselines computed on the same test windows. The table in the main text is generated from those records, so no number in the paper is transcribed by hand.

R LIMITATIONS AND FUTURE WORK

The evidence is a one-seed grid on seven datasets of one benchmark at one lookback, read against the run-to-run spread rather than tested; Section P adds a second lookback on five of them, and repetition over seeds would sharpen every count. The saving is in time and on the variable axis alone. Along time the fluxer costs about what attention does, the whole model uses more memory than fused attention, and with attention in `bfloat16` the fluxer is the faster mixer only above 4096 tokens, a gap that a half-precision transform at arbitrary lengths would narrow. The circulant reads the labeling of an axis that has none, at a cost of about 1.16% under relabeling (Section N); an equivariant closure at the same cost remains an open question. The datasets that reward the variable axis most are those whose variance is most concentrated (Section D), a coincidence that seven datasets cannot make a rule.