TRACE: RELAYING STRUCTURAL STATE BEYOND THE FIXED INPUT WINDOW FOR NON-STATIONARY TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

The reversible normalization family, from RevIN through SAN, FAN, and DDN, mitigates non-stationarity in real-world time series with progressively more precise restoration statistics, but each method estimates its correction quantities from the current input window alone. When the window is short, observation noise dominates the level and phase of the periodic components, and the forecast origin never uses observations before the window boundary. This work proposes TRACE, an adapter that carries the history preceding the window in a small structural state that is relayed from one forecast origin to the next. A fixed linear state-space filter with a per-channel level and K harmonic components updates the state causally and has no learnable parameters; at forecast time the state is rolled out analytically and combined with the backbone forecast through a learned gate. Because the gate starts at zero, TRACE is initialized exactly at the backbone with window-statistic normalization and can learn to close the structural branch. The history reaches the state, but never the backbone, so this state continuity leaves the backbone's input length unchanged and is a design choice separate from window extension. Across four forecasting backbones, seven real-world benchmarks and three seeds, TRACE lowers the overall mean MSE by 10.66% against RevIN and by 6.10% against DDN, the strongest competing adapter, while using only 226–610 learnable parameters, 64–10,129 times fewer than SAN, FAN and DDN. Controlled experiments show that the gain depends on history the backbone never sees and that window-only forecasts combined in the same way do not reproduce it. The code is available at `https://anonymous.4open.science/r/TRACE-8346`.

1 INTRODUCTION

Deep forecasters have raised the accuracy of time series forecasting considerably (Zeng et al., 2023; Wang et al., 2023; Chen et al., 2023; Liu et al., 2024), but real-world series are non-stationary: their mean, variance, and periodic components change over time. The same model therefore meets a different distribution in every input window. Non-stationarity has been identified as a main factor limiting deep forecasters (Kim et al., 2022; Liu et al., 2022).

The reversible normalization family was developed to mitigate this. RevIN normalizes each input window by its own mean and variance and restores those statistics at the output (Kim et al., 2022), and subsequent work made the quantities used for restoration more precise. SAN estimates them per time slice, FAN from the leading Fourier components, and DDN from sliding statistics in both the time and the frequency domain (Liu et al., 2023b; Ye et al., 2024; Dai et al., 2024). This family is practical because it can be attached without changing the backbone, and our TRACE retains that property.

This normalization family shares one constraint: it estimates the quantities used for normalization *from the current input window only*. Two consequences follow: 1) The current level and the phase of the periodic components, which the forecast needs directly, are not identified stably by a single window of length L. 2) Observations before the window boundary are inaccessible at the forecast origin, so the correction is a function of the window alone: two origins whose windows agree receive

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

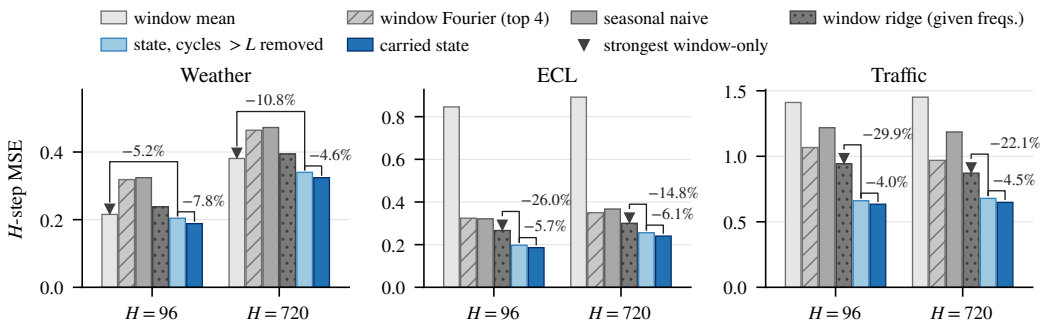

Figure 1: H-step MSE of structural forecasts on three datasets with pronounced cycles. Grey bars use only the window of length L: its mean, its four largest Fourier components, a seasonal naive forecast and a ridge fit given the periods the state uses. Blue bars use the state carried over the entire history, without and with its components of period at least L. The state uses the K most often selected for each dataset in the main table. Brackets give the relative change in MSE, the upper one from the strongest window-only forecast (▼) to the carried state without long components and the lower one between the two carried states. Details and the other datasets are in Appendix E.

the same correction no matter how their earlier histories differ. This work addresses these two consequences for series with persistent periodic structure.

Structural state-space models (Harvey, 1989; Durbin & Koopman, 2012) hold these quantities directly: they summarize the level and the phase of each periodic component, at periods fixed in advance, in a small state vector. They update it recursively as observations arrive, and forecast by rolling it forward through a known transition. Because the update is recursive, the state at a forecast origin can be carried over the entire history rather than rebuilt from the current window, and this work builds on that property.

Figure 1 tests this on three datasets with pronounced cycles by scoring H-step structural forecasts against the observations. Even without its components longer than the window, the carried state beats every window-only forecast by 5–30% against the strongest of them (▼, upper brackets), so the same components are estimated more reliably from the history than from one window. The components longer than the window, whose frequencies lie below the lowest non-zero frequency of a window's discrete Fourier basis, lower the error by a further 4–8% (lower brackets).

Relaying the state from one forecast origin to the next addresses both consequences at once. The level and phase estimates draw on the whole history rather than on one window. Observations before the window boundary remain in the state instead of being discarded. This is state continuity rather than window extension: a longer window still discards everything before its new boundary and feeds the added observations to the backbone, whereas a continuous state processes an arbitrarily long stream recursively in a fixed size.

Recursive state estimation itself is classical (Harvey, 1989; Durbin & Koopman, 2012). What this work adds is carrying such a state in the adapter slot of a window-based forecaster, as a forecast branch rather than a normalization constant, together with evidence of where its gain comes from. In this work, we propose TRACE (s**T**ructural state **R**elayed **A**cross origins with **C**losed-form **E**xtrapolation), an adapter that relays a structural state across forecast origins. It combines the closed-form forecast of that state with the backbone's through a learned gate, rather than estimating another set of restoration statistics. Its *linear state-space model* maintains, for each channel, a structural state made of a level component and K harmonic components at fixed periods, a level-only variant of the basic structural model of Harvey (1989). It updates the state causally via a Kalman update (Kalman, 1960) whenever an observation arrives. Instead of reinitializing the state at each forecast origin, it relays it from one origin to the next. Because the state transition is explicit, the structural forecast over the horizon is obtained in closed form by rolling the state forward (Sec. 3.3), with no learned decoder. All filter quantities are fixed before backbone training, so *the filter has no learnable parameters* and the gate is the only component trained, jointly with the backbone. Because the rolled-out state is already a forecast, TRACE keeps it as an independent branch rather than converting it into normalization

statistics, and combines it with the backbone forecast as in classical forecast combination (Bates & Granger, 1969).

We make three contributions:

1. **State continuity as a design axis.** We identify the fundamental constraint shared by the normalization family as *estimating the correction quantities from the current input window only*, and name the alternative to it *state continuity*, keeping one state across forecast origins. It is an axis distinct from *window extension* in cost and in mechanism, though the two can supply the same information (Sec. 4.3). The size of the state and its per-origin computation grow with neither the window nor the length of the history carried.
2. **A parameter-free branch with a small gate.** We propose TRACE, which relays a fixed structural state across origins and keeps its closed-form forecast as a separate branch rather than a normalization constant. The contribution of the state therefore remains observable in the gate and in the cross moment of the two branch errors; the only learned component is a gate of at most 610 parameters. That count does not grow with the number of channels and is 64–10,129 times smaller than the learnable parameters of SAN, FAN and DDN in the same cell.
3. **Lower error than existing adapters with controlled experiments.** Over four backbones, seven datasets and four forecast horizons, TRACE attains the lowest mean MSE in 85 of 112 cells against the four normalization adapters. It lowers the mean error of every backbone on every dataset relative to the backbone alone. Controlled experiments show that the gain largely disappears when the state is restricted to the window and that it comes from complementarity between the two branches rather than a stronger structural forecast. Under the same combination rule, the carried state with its components longer than the window removed outperforms every window-only forecast, including a ridge fit given the frequencies of the state, in all 12 controlled cells. The full state does so in 8.

2 RELATED WORK

Normalization adapters. The family that handles non-stationarity outside the backbone normalizes the input and restores it at the output, as RevIN does with the window mean and variance and a learnable per-channel affine transform (Kim et al., 2022). Since this assumes that the window and the horizon share a distribution, later work refined what is removed and restored. The Non-stationary Transformer returns the statistics to attention (Liu et al., 2022), Dish-TS separates the input and horizon distributions (Fan et al., 2023), and SAN works per time slice and forecasts future slices (Liu et al., 2023b). FAN uses leading Fourier components instead of the mean and variance (Ye et al., 2024), DDN combines sliding statistics in time and frequency (Dai et al., 2024), and SIN learns a projection (Han et al., 2024). The source of those quantities has not changed: all of them compute it inside the current input window.

Window extension. A longer window also relaxes this constraint (Zeng et al., 2023; Nie et al., 2023), but it still discards the history before its new boundary. TRACE keeps the backbone input fixed and relays the history on a separate path, so the two mechanisms are distinct and can be combined. Their benefits do overlap when a longer window already covers the relevant cycles (Appendix A).

Test-time adaptation. Online and continual learners (Pham et al., 2023; Zhang et al., 2023; Zhao & Shen, 2025) and test-time adapters (Kim et al., 2025; Im & Kwon, 2026) also use observations that arrive after deployment, but as realized targets that update parameters by gradient steps. TRACE feeds the same observations into a fixed-form state and changes no parameter at test time, so it can be stacked under such adapters. Because they follow a different protocol, we evaluate that stacking separately (Appendix B).

Carried statistics. ST-Norm normalizes spatial and temporal components separately (Deng et al., 2021) and AdaRNN aligns the hidden state of a recurrent model across segments with different distributions (Du et al., 2021), but neither carries a correction quantity across forecast origins. GAS-Norm (Urettini et al., 2024) does: it updates a statistical state at each observation by a generalized autoregressive score and forecasts future statistics, making the same choice as TRACE in carrying a

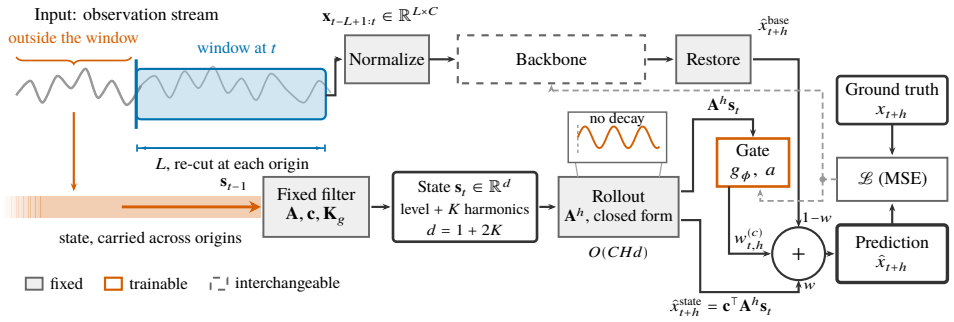

Figure 2: Architecture of TRACE. The backbone reads a window re-cut at every origin, while a fixed filter updates a state carried across origins and the rollout extends it over the horizon in closed form; a gate that reads only the rolled-out state sets the weight w given to the structural forecast $\widehat{x}^{\text{state}}_{t+h}$ against the backbone forecast $\widehat{x}^{\text{base}}_{t+h}$. The filter, the state box it updates from the carried state $\mathbf{s}_{t-1}$, and the rollout that reads it are the subject of Sec. 3.3; Normalize, Restore, the gate, and the interpolation are those of Sec. 3.4. Grey dashed arrows carry the gradient of the single loss.

state across origins. Two constraints remain. First, what is carried is statistics such as the mean and the variance, which track the level but have no place for the phase of a periodic component. Second, that state is consumed as a normalization constant, so its contribution is not separable from the backbone forecast. TRACE instead keeps the structural forecast as an independent branch combined through an explicit gate, so its contribution is observed directly (Sec. 4.3). Unlike GAS-Norm, whose update coefficients are fitted on the training segment, its filter has no learnable parameters.

Structural state-space models. Linear state-space models summarize the observation history recurrently in a fixed size and compute multi-step forecasts analytically (Kalman, 1960; Durbin & Koopman, 2012). Because the state is set at design time, unlike a learned recurrent state, one can ask afterwards which components are transmitted. Deep hybrids generate the state-space parameters with a network (Rangapuram et al., 2018), separate global neural factors from local probabilistic models (Wang et al., 2019), transplant the exponential-smoothing decomposition into attention (Woo et al., 2022), or treat non-stationary dynamics with a Koopman operator (Liu et al., 2023a). This family is explicit about what it holds but designs the forecaster itself. TRACE uses the same structural state as an adapter that does not replace the backbone, so its contribution can be measured in isolation.

3 METHOD

3.1 OVERVIEW

Figure 2 shows the overall structure of TRACE. A fixed filter updates the state along the observation stream. That state is not reinitialized as the forecast origin moves, whereas the backbone's input window is re-cut at every origin. Observations earlier than the window therefore reach the state only and never pass through the backbone. The state is rolled out analytically over the forecast horizon to give a structural forecast, which is interpolated with the backbone forecast by a weight that a gate computes from the rolled-out state. Only the gate that sets the interpolation weight is learned, jointly with the backbone under a single forecasting loss whose gradient is drawn as grey dashed lines in Figure 2. The backbone, drawn dashed, can be any window-based forecaster. Training only the gate on a frozen, independently trained backbone performs on par with joint training, with the difference falling within the spread across seeds (Appendix C), so a trained backbone can also be kept as it is.

3.2 PROBLEM FORMULATION

Let $\mathbf{x}_t \in \mathbb{R}^C$ be the observation at time t over C channels. A forecasting backbone maps the L observations up to t to a forecast $\widehat{\mathbf{Y}}^{\text{base}}_{t+1:t+H}$ over the next H steps. The goal is to preserve, in a fixed size, the level and periodic components of the observations before $t - L + 1$ without changing the backbone's input length L or its architecture, and to measure their contribution separately. Appendix D collects the notation of this section.

3.3 THE STRUCTURAL STATE

The state is what the adapter remembers of the series: a small set of components whose meaning is fixed before training and which every arriving observation updates. The state at an origin therefore summarizes the whole history rather than the current window.

Components. TRACE maintains a state $\mathbf{s}_t \in \mathbb{R}^d$ for each channel, and in this subsection x_t denotes the observation of that channel. The state consists of one level component, a scalar that tracks the local mean, and K two-dimensional harmonic components, so $d = 1 + 2K$. Each harmonic component is a sinusoid at a fixed period whose amplitude is the norm of its pair and whose phase is its angle. This is the basic structural model of Harvey (1989) without a slope term and with the frequencies fixed in advance. The transition matrix $\mathbf{A}$ is block diagonal, made of a term that preserves the level and a rotation block $\left(\begin{smallmatrix} \cos\omega_k & \sin\omega_k \\ -\sin\omega_k & \cos\omega_k \end{smallmatrix}\right)$ for each frequency ω_k. The fixed observation readout $\mathbf{c}$ sums the level and the first coordinate of each harmonic component. Since $\mathbf{A}$ is a direct sum of an identity and rotation blocks, $|\lambda(\mathbf{A})| = 1$: no component is damped or amplified, and each harmonic keeps its amplitude while its phase advances. Every coordinate therefore has its meaning fixed before training and can be read individually afterwards (Appendix E).

Causal update. With the steady-state Kalman gain $\mathbf{K}_g$ fixed, we write the update as a prediction step and a correction step.

$$\mathbf{s}_t^- = \mathbf{A}\mathbf{s}_{t-1}, \qquad \text{(prediction)} \tag{1}$$

$$\mathbf{s}_t = \mathbf{s}_t^- + \mathbf{K}_g\big(x_t - \mathbf{c}^\top \mathbf{s}_t^-\big). \qquad \text{(correction)} \tag{2}$$

With a fixed gain this recursion is an exponential-smoothing filter with trigonometric seasonal components (Harvey, 1989), so a carried seasonal exponential-smoothing forecast with the same periods is the structural branch itself. Starting from a zero state, the two equations are applied once per arriving observation along the full series, without reinitialization at the boundaries between data segments and without restarting as the forecast origin moves. The $\mathbf{s}_t$ used at origin t is therefore a summary of the entire history, including observations before $t - L + 1$. Whether continuing to correct the state helps after an abrupt change depends on the kind of change, which we measure on synthetic changes fixed in advance (Appendix F).

Configuration. All filter quantities except K are configured from the data before backbone training and then frozen, and K is selected together with the learning rate (Sec. 4.1); the filter has no parameter trained by gradient. The frequencies are taken from the K leading peaks of the periodogram summed across channels, so $(\mathbf{A}, \mathbf{c}, \mathbf{K}_g)$ is shared by all channels while each channel keeps its own state. The process-noise ratios are chosen by the H-step error of the structural forecast alone, and the steady-state gain $\mathbf{K}_g$ follows from iterating the Riccati recursion until convergence (Anderson & Moore, 1979). None of these quantities uses the test segment. Appendix G gives the candidate ranges and the exact form of the recursion.

Persistence. The filter that absorbs observations, $(\mathbf{I} - \mathbf{K}_g\mathbf{c}^\top)\mathbf{A}$, forgets old observations at a rate set not by training but by its spectral radius ρ: the slowest mode of the state halves after $\ln 0.5 / \ln \rho$ steps. A fixed-size state therefore processes an arbitrarily long stream while its persistence is fixed at design time. On the factor cells the slowest mode halves after 738–1,791 observations, 7.7–18.7 times the window. This is the persistence of the state rather than an effective history length, since the influence of an observation also passes through the gain and the readout (Appendix H).

Rollout. At forecast origin t, the structural forecast h steps ahead is computed analytically:

$$\widehat{x}^{\,\text{state}}_{t+h} = \mathbf{c}^\top \mathbf{A}^h \mathbf{s}_t, \qquad h = 1, \ldots, H \tag{3}$$

The rollout therefore extends the components held in the state over the whole horizon without any parameter per h. Figure 4 shows it at one test origin of Weather, chosen as the origin at which the carried state's own error is the median over test origins. Its 144-step daily cycle is longer than the window, so the window's leading Fourier components can only repeat the last 96 observations. The carried state instead continues the cycle and lowers the mean absolute error from 0.047 to 0.028. The trajectory can nonetheless swing more widely late in the horizon than near the origin (Appendix E), not because any component grows but because their phases drift into alignment. It therefore cannot be added as it is; its weight has to be bounded, which the gate of the next subsection does.

3.4 THE GATE

The gate decides how much of the structural forecast to use: it weighs the two branches separately at every origin, lead and channel, and its weight is bounded so that the combination always stays between them.

Normalize and restore. The backbone receives an input normalized by the window mean and standard deviation, and its output is restored with the window statistics, which returns it to the training-standardized coordinates in which the filter runs. The channel-c entry of this output at lead h is written as $\widehat{x}^{\mathrm{base}}_{t+h}$, and the two branch forecasts are combined without rescaling.

Interpolation weight.

$$w_{t,h} = \mathrm{clamp}(a, 0, 1) \cdot \sigma\big(g_\phi(\mathbf{A}^h \mathbf{s}_t)\big)\,, \tag{4}$$

$$\widehat{x}_{t+h} = \widehat{x}^{\mathrm{base}}_{t+h} + w_{t,h}\big(\widehat{x}^{\mathrm{state}}_{t+h} - \widehat{x}^{\mathrm{base}}_{t+h}\big). \tag{5}$$

The two equations are for a single channel c, whose state is $\mathbf{s}_t$, and σ denotes the sigmoid. The gate multi-layer perceptron (MLP) g_ϕ is $d \to 32 \to 1$ and shares its weights across origin t, lead h, and channel c. Its input $\mathbf{A}^h\mathbf{s}_t$ differs across all three, so the value of $w_{t,h}$ can vary with all three as well. Its input is not normalized by the window, but every channel is standardized with training statistics before the filter runs, so the shared g_ϕ sees comparable scales across channels. The scalar a is shared globally and caps the maximum weight of the structural branch. That cap is what guarantees $w \in [0, 1]$ and makes Eq. (5) an interpolation rather than an additive correction. Even where the structural branch is wrong, the combined error at any channel, origin and lead can never exceed that of the worse of the two branches, which need not be the backbone (Proposition 1, Appendix I).

What the gate reads. The structural forecast is kept as a separate term on the output side, rather than fed into the backbone input as the normalization family does. Its contribution therefore stays observable in the value of $w_{t,h}$ and in the cross moment of the two branch errors (Sec. 4.3). For the same reason, the gate reads only the rolled-out state $\mathbf{A}^h\mathbf{s}_t$, not the backbone forecast or the input window. A weight that responded to the backbone output would mix the two contributions again, so the weight is a function of the structural branch alone.

Initialization. The weight and bias of the last linear layer of g_ϕ, and a, are initialized at zero (the clamp is straight-through, so a receives gradient at any value). At initialization $w_{t,h} \equiv 0$, so Eq. (5) is bit-for-bit identical to the backbone forecast. If a stays at zero, the adapter is by construction the same function as normalizing and restoring with window statistics, that is, RevIN without its affine transform. The difference from RevIN therefore reflects the state up to that affine transform, whose effect is measured separately (Sec. 4.2, Appendix J). A scalar weight fitted on the validation split does about as well, and is in fact lower in more cells than the learned gate (Sec. 4.3). We keep the learned gate because it is trained with the backbone in one stage and starts exactly at the backbone, not because the gain requires its flexibility.

3.5 SIZE AND COST

Since only g_ϕ and a are learned, the adapter's learnable parameter count is exactly $32d + 66$, determined by K alone and *independent of the number of channels* C. Because the filter is fixed, the states along the whole series are computed once before training, and each forecast origin only reads its state and rolls it out. Because $\mathbf{A}$ is block diagonal, both cost $O(d)$ per step, which is $O(Cd)$ per observation and $O(CHd)$ per origin. Neither cost depends on the input length or the history length, whereas the backbone's computation grows with L. Appendix K gives the measured parameter counts and peak memory, including the comparison against a longer window.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate six configurations per cell – the plain backbone, the four normalization adapters RevIN, SAN, FAN and DDN, and TRACE. The grid is four backbones (DLinear, MICN, TSMixer, iTrans-

Table 1: MSE and MAE of each backbone alone and with TRACE attached, per dataset and forecast horizon H (means over three seeds, $L = 96$; standard deviations in Appendix O); bold marks the lower of each pair.

Methods		DLinear		+TRACE		MICN		+TRACE		TSMixer		+TRACE		iTransformer		+TRACE	
Metrics		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
ECL	96	0.302	0.210	**0.250**	**0.155**	0.270	0.160	**0.249**	**0.146**	0.323	0.225	**0.255**	**0.152**	0.248	0.150	**0.235**	**0.137**
	192	0.305	0.210	**0.259**	**0.166**	0.285	0.173	**0.264**	**0.164**	0.333	0.231	**0.267**	**0.167**	0.263	0.165	**0.255**	**0.158**
	336	0.319	0.223	**0.274**	**0.181**	0.297	0.186	**0.268**	**0.167**	0.350	0.246	**0.284**	**0.182**	0.295	0.194	**0.267**	**0.169**
	720	0.350	0.258	**0.307**	**0.220**	0.321	0.214	**0.287**	**0.189**	0.378	0.280	**0.308**	**0.213**	0.315	0.214	**0.304**	**0.213**
ETTh1	96	0.411	0.396	**0.403**	**0.384**	0.441	0.432	**0.414**	**0.399**	0.492	0.503	**0.408**	**0.388**	0.425	0.403	**0.411**	**0.386**
	192	0.441	0.445	**0.427**	**0.418**	0.477	0.485	**0.441**	**0.429**	0.530	0.569	**0.455**	**0.455**	0.469	0.470	**0.444**	**0.438**
	336	0.465	0.487	**0.450**	**0.458**	0.538	0.571	**0.456**	**0.454**	0.561	0.622	**0.460**	**0.474**	0.505	0.528	**0.471**	**0.487**
	720	0.511	**0.514**	**0.496**	0.534	0.657	0.770	**0.483**	**0.504**	0.604	0.661	**0.527**	**0.585**	0.554	0.575	**0.491**	**0.506**
ETTh2	96	0.398	0.345	**0.334**	**0.282**	0.412	0.369	**0.341**	**0.287**	0.547	0.572	**0.345**	**0.298**	0.516	0.521	**0.350**	**0.299**
	192	0.477	0.478	**0.384**	**0.365**	0.488	0.504	**0.394**	**0.374**	0.639	0.773	**0.399**	**0.384**	0.639	0.738	**0.398**	**0.379**
	336	0.541	0.592	**0.417**	**0.398**	0.542	0.601	**0.427**	**0.409**	0.699	0.909	**0.435**	**0.431**	0.723	0.930	**0.432**	**0.421**
	720	0.656	0.828	**0.431**	**0.398**	0.645	0.802	**0.436**	**0.399**	0.767	1.066	**0.443**	**0.414**	0.827	1.170	**0.445**	**0.425**
ETTm1	96	0.373	0.346	**0.364**	**0.324**	0.371	0.319	**0.361**	**0.317**	0.435	0.410	**0.358**	**0.308**	0.413	0.381	**0.370**	**0.319**
	192	0.391	0.382	**0.377**	**0.341**	0.403	0.366	**0.383**	**0.339**	0.450	0.440	**0.380**	**0.333**	0.434	0.418	**0.389**	**0.355**
	336	0.414	0.414	**0.396**	**0.361**	0.439	0.413	**0.399**	**0.363**	0.480	0.485	**0.398**	**0.364**	0.467	0.464	**0.407**	**0.380**
	720	0.450	0.473	**0.430**	**0.414**	0.503	0.512	**0.426**	**0.412**	0.532	0.572	**0.430**	**0.420**	0.521	0.549	**0.437**	**0.435**
ETTm2	96	0.292	0.193	**0.261**	**0.177**	0.281	0.183	**0.259**	**0.175**	0.380	0.282	**0.259**	**0.175**	0.335	0.232	**0.265**	**0.183**
	192	0.358	0.281	**0.299**	**0.234**	0.355	0.281	**0.299**	**0.232**	0.457	0.410	**0.300**	**0.233**	0.428	0.368	**0.304**	**0.238**
	336	0.426	0.382	**0.338**	**0.291**	0.420	0.379	**0.340**	**0.291**	0.539	0.553	**0.339**	**0.291**	0.567	0.604	**0.346**	**0.301**
	720	0.511	0.533	**0.396**	**0.399**	0.516	0.554	**0.395**	**0.388**	0.713	0.912	**0.404**	**0.409**	0.783	1.077	**0.402**	**0.403**
Traffic	96	0.429	0.696	**0.346**	**0.516**	0.311	0.518	**0.288**	**0.412**	0.484	0.761	**0.326**	**0.452**	0.297	0.530	**0.274**	**0.381**
	192	0.407	0.647	**0.339**	**0.500**	0.316	0.536	**0.294**	**0.432**	0.478	0.731	**0.328**	**0.464**	0.303	0.553	**0.280**	**0.402**
	336	0.410	0.653	**0.340**	**0.509**	0.321	0.547	**0.298**	**0.445**	0.488	0.750	**0.337**	**0.481**	0.315	0.580	**0.285**	**0.413**
	720	0.429	0.694	**0.355**	**0.544**	0.331	0.572	**0.318**	**0.480**	0.508	0.801	**0.351**	**0.511**	0.328	0.608	**0.301**	**0.442**
Weather	96	0.257	0.196	**0.226**	**0.165**	0.263	0.187	**0.214**	**0.161**	0.250	0.181	**0.203**	**0.147**	0.231	0.170	**0.215**	**0.158**
	192	0.295	0.237	**0.262**	**0.212**	0.303	0.232	**0.255**	**0.208**	0.285	0.218	**0.247**	**0.195**	0.274	0.217	**0.256**	**0.208**
	336	0.331	0.282	**0.289**	**0.243**	0.342	0.278	**0.288**	**0.248**	0.319	0.261	**0.283**	**0.236**	0.322	0.276	**0.288**	**0.242**
	720	0.382	0.346	**0.331**	**0.298**	0.377	0.329	**0.330**	**0.301**	0.360	0.318	**0.327**	**0.296**	0.383	0.359	**0.329**	**0.298**

former), seven datasets (ECL, ETTh1, ETTh2, ETTm1, ETTm2, Traffic, Weather (Zhou et al., 2021; Wu et al., 2021)) and four forecast horizons $H \in \{96, 192, 336, 720\}$. The comparison set is the adapters that attach to a fixed backbone under the standard offline protocol; TRACE contains RevIN's normalize-and-restore as its zero-gate special case (Sec. 3.4). SIN (Han et al., 2024) and GAS-Norm (Urettini et al., 2024) belong to the same set but are left out because no public implementation fits this protocol; Appendix G gives the reasons and, for GAS-Norm, what its closest analog reaches here. Exchange (Lai et al., 2018), which has no dominant calendar cycle at its sampling interval, is reported separately because the public scripts of the baselines do not cover it (Appendix L). One combination of backbone, dataset and forecast horizon is called a *cell*. Every cell is filled, and the six methods share the same data split, the same input length $L = 96$, the same three seeds, and the same early-stopping rule. Unlike the window-only baselines, TRACE retains the observations before the window boundary, which is the design under study. The five baselines use the dataset-specific settings of the original authors' public scripts, and a learning-rate sensitivity check for RevIN and DDN is given in Appendix N. For TRACE, the learning rate and K are selected once per backbone–dataset pair by the mean rank across the four horizons and applied unchanged to all of them (Appendix G, Table 21). The metrics are MSE and MAE on the test segment.

The controlled experiments of Sec. 4.3 use 12 *factor cells*, DLinear and TSMixer on ETTh1, ETTm2 and Weather at $H \in \{96, 720\}$. These share one fixed setting ($K = 8$ and one learning rate), so their values are matched pairs within a cell rather than main-table entries (Appendix N). They are a controlled subset that fixes $K = 8$ and excludes ECL and Traffic, so they identify the mechanism rather than decompose every main-table gain.

4.2 MAIN RESULTS

Tables 1 and 2 report means over the three seeds; the standard deviations of individual horizons are in Appendix O.

Table 2: MSE averaged over the four forecast horizons (means over three seeds, $L = 96$); bold and underline mark the lowest and the second lowest at full precision in each backbone–dataset group.

Models	DLinear					MICN					TSMixer					iTransformer				
Methods	TRACE	DDN	SAN	FAN	RevIN	TRACE	DDN	SAN	FAN	RevIN	TRACE	DDN	SAN	FAN	RevIN	TRACE	DDN	SAN	FAN	RevIN
ECL	**0.181**	0.198	0.204	0.210	0.230	0.167	**0.166**	0.175	0.170	0.183	**0.178**	0.200	0.196	0.197	0.240	0.169	0.178	0.172	0.186	**0.169**
ETTh1	0.448	0.449	**0.439**	0.481	0.456	**0.447**	0.530	0.452	0.489	0.482	0.476	**0.457**	0.468	0.483	0.490	**0.454**	0.471	0.508	0.481	0.455
ETTh2	**0.361**	0.379	0.372	0.489	0.378	**0.367**	0.381	0.379	0.513	0.392	**0.382**	0.388	0.427	0.494	0.402	0.381	**0.380**	0.393	0.488	0.383
ETTm1	**0.360**	0.387	0.383	0.408	0.409	**0.358**	0.394	0.392	0.388	0.396	**0.356**	0.392	0.380	0.407	0.402	**0.372**	0.393	0.383	0.407	0.406
ETTm2	**0.275**	0.285	0.280	0.321	0.285	**0.271**	0.282	0.276	0.309	0.279	**0.277**	0.290	0.360	0.329	0.289	**0.281**	0.287	0.282	0.321	0.291
Traffic	**0.517**	0.521	0.615	0.602	0.669	**0.442**	0.497	0.550	0.522	0.529	**0.477**	0.530	0.617	0.576	0.723	**0.409**	0.480	1.378	0.545	0.447
Weather	**0.229**	0.248	0.254	0.251	0.269	**0.230**	0.254	0.267	0.234	0.253	**0.219**	0.247	0.251	0.245	0.263	**0.226**	0.250	0.247	0.253	0.254

Table 3: Sources of the gain on the 12 factor cells, which share one fixed setting. *Gain*: relative MSE reduction of TRACE against the same backbone trained without the state branch. *History*: relative MSE reduction of the carried state over a state restricted to the $M{=}96$ observations inside the window, imposed at inference on the same checkpoint (inf.) or throughout training (retr.). *Combination*: error of the structural forecast alone (R_D) and cross moment of the two branch errors (C_{BD}), relative to the backbone error R_B; $C_{BD}/R_B < 1$ means that a small structural weight lowers the error.

		DLinear					TSMixer				
		Gain	History (%)		Combination		Gain	History (%)		Combination	
Data	H	(%)	inf.	retr.	R_D/R_B	C_{BD}/R_B	(%)	inf.	retr.	R_D/R_B	C_{BD}/R_B
ETTh1	96	0.46	0.52	0.45	1.24	0.89	0.74	0.59	0.65	1.15	0.83
	720	0.69	0.53	0.54	1.17	0.85	0.78	0.45	0.59	1.07	0.79
ETTm2	96	2.50	2.87	2.24	2.54	0.77	3.40	3.20	2.56	2.49	0.75
	720	0.90	8.60	1.49	3.30	0.85	0.62	6.55	3.10	3.30	0.84
Weather	96	12.20	9.19	12.20	1.84	0.49	9.80	6.69	9.21	1.93	0.54
	720	2.26	-2.41	-2.05	3.33	0.78	1.82	-1.97	-1.73	3.41	0.78

Attaching TRACE lowers the mean MSE in all 28 backbone–dataset combinations and in 111 of the 112 cells, and the mean MAE in all 112 cells (Table 1). The gain is largest on ETTh2, ETTm2 and Traffic. The backbone alone has no normalization, so the contribution of the state itself is read against RevIN in Table 2.

TRACE has the lowest mean MSE in 85 of the 112 individual cells, with 105 wins against RevIN, 108 against FAN, 97 against DDN and 101 against SAN. In the ratio of overall means, $100\,(1 - \bar{e}_{\text{TRACE}}/\bar{e}_{\text{ref}})$ with $\bar{e}$ the MSE averaged over the cells, it reduces MSE by 10.66% [6.84, 14.92] against RevIN. Against DDN, the strongest of the four adapters, the reduction is 6.10% [4.00, 8.20]. The 95% intervals come from resampling the 28 backbone–dataset pairs (Appendix O). Against RevIN, the same ratio is 12.1%, 9.3%, 15.8% and 4.6% for DLinear, MICN, TSMixer and iTransformer, respectively. On MAE, the corresponding overall-mean reductions are 4.67% [2.38, 7.56] and 2.15% [0.75, 3.56]. None of these intervals contains zero, and on MSE TRACE has the lower horizon-averaged error in 25–28 of the 28 backbone–dataset pairs against every adapter (Appendix O, Table 24). In a 2×2 design separating RevIN's affine transform from the state, the state lowers the error in 12 of the 12 factor cells with the affine both off and on. The affine main effect ranges from -0.54 to $+1.30$ pp with no consistent sign, and the interaction is at most 0.18 pp. The difference against RevIN is therefore effectively the effect of the state (Appendix J).

Accuracy alone does not settle the comparison, because these adapters differ in size by up to four orders of magnitude and a method can be accurate by being large. Read on both axes at once, TRACE is *first in accuracy and second smallest in size*. It attains the best mean rank on MSE and on MAE, 1.34 on MSE, and is first or second in 105 of the 112 cells. Its median learnable parameter count is 354, against 134.9k for SAN and 299.3k for DDN, which is second on both ranks and about 845 times larger; only RevIN is smaller, at 14 (Table 16 and Figure 6, Appendix K).

4.3 WHERE THE GAIN COMES FROM

On the 12 factor cells, Table 3 asks where the gain comes from. The four paragraphs below ablate the out-of-window history, decompose the errors of the two branches, substitute the structural branch with seven alternatives, and compare the whole design against window extension.

Out-of-window history. The *History* columns of Table 3 keep the backbone window at $L = 96$ and restrict only the history M that the state may absorb before the origin. The restriction is applied at inference on the same checkpoint or throughout training. Retraining is the cleaner control, since restricting at inference also restarts the filter from a finite history. Both favor the carried state in 10 of the 12 cells, and retraining under the restriction then removes most of the gain—on DLinear with Weather at $H = 96$ all 12.20% of it. The whole gain there is therefore attributable to the out-of-window history. The cap a of Eq. (4) ends at zero, leaving the backbone alone, only in that cell and in DLinear on ETTh1 at $H = 96$, both retrained under the restriction, and never when the state carries the out-of-window history (Appendix P). Because every component of the state has a fixed meaning, the two cells where a shorter history is better, Weather at $H = 720$, can be traced to its long-period components. With zero, two or four harmonics instead of eight, the carried state lowers the error there as well (Appendix P).

Complementary combination. In 12/12 cells $C_{BD}/R_B < 1$, exactly the condition under which a small structural weight lowers the error regardless of R_D (Proposition 2, Appendix I). Yet $R_D/R_B > 1$ in all 12 and exceeds 3 at $H = 720$ on ETTm2 and Weather. The gain therefore comes from the complementarity of the two branches, not from a structural forecast that is better on its own. The lowest C_{BD}/R_B, on Weather at $H = 96$, is also the cell with the largest dependence on out-of-window history.

Would any second forecast do? Complementarity could come from any forecast that differs from the backbone, so Table 9 combines seven branches with the same independently trained backbone through one constant weight fitted without the test segment (Appendix C). That weight performs about as well as the learned gate (median difference 0.06%, Appendix C). The window-only forecasts of Figure 1, including a ridge fit given the frequencies of the filter, gain at most 1.2% at the median and stay correlated with the backbone (C_{BD}/R_B of 0.94–0.97). The carried state with every component longer than the window removed gains 8.5%, leads every window-only branch in all 12 cells and reaches $C_{BD}/R_B = 0.78$. The complementarity therefore comes from carrying the state rather than from adding a second forecast. Carrying the level alone is not enough. The longer components that the fixed $K = 8$ of the factor cells includes without selection help only on Weather at $H = 96$, where they contain the daily cycle. Validation in the main table selects $K = 2$ on ETTm2 and Weather and drops most of them (Table 21).

Window extension instead. State continuity and window extension can supply the same information, so we also lengthen the backbone window to $L = 336$ (Appendix A). The carried state then still lowers the error of RevIN on Weather and ECL (7 of 8 cells) but not on ETTh1 and ETTm2 (1 of 8), where the structural forecast alone is weakest (Appendix E). Its benefit is largest when the window is short relative to the cycles of the data. On Weather it is not replaceable by a longer window at all, TRACE staying below RevIN in all four cells by a median of -9.3%. The two axes also compose: lengthening the window still lowers the error of TRACE in 11 of the 16 cells, and the structural branch adds only 0.6–6.3% peak memory from $L = 96$ to $L = 336$.

5 CONCLUSION

This work proposed TRACE, an adapter that carries the history outside a fixed input window in a structural state with no learnable parameters, relayed from one forecast origin to the next, and combines it with the forecast of an arbitrary window-based backbone through a learned gate. Across 112 cells, the adapter attains the lowest mean MSE in 85 and reduces the overall mean MSE by 10.66% against RevIN and 6.10% against DDN. The controlled experiments show that the gain largely disappears when the state is restricted to the window and that it comes from the complementarity of the two branches rather than from a stronger structural forecast. Because the filter basis is fixed before training, the gain is smaller on data without stable cycles or with drifting frequencies (Appendix Q). A basis that adapts to the data is a natural next step.

Ethics Statement

This work proposes a forecasting adapter and evaluates it on public benchmark datasets of electricity consumption, transformer temperature, traffic occupancy and weather measurements that contain no personal data. We foresee no ethical concern specific to the method beyond those common to forecasting research; its failure conditions, in particular series whose periodic structure changes (Appendix Q), are reported so that deployments can account for them.

Reproducibility Statement

The seven main-table datasets and Exchange are public long-term-forecasting benchmarks, used with the splits and channel counts given in Appendix G. Appendix G specifies the filter selection, gate architecture, an exhaustive learning-rate and K search, selection by the mean validation rank across the four horizons, baseline training schedules, and seeds. The selected configurations and common search budget are reported in Table 21; the remaining result and parameter tables are listed in the same appendix. The tables and every number in the body are generated from the recorded result files by the scripts named in Appendix G, which are released with the code at `https://anonymous.4open.science/r/TRACE-8346`.

AI Use Statement

Large language models were used to polish the writing of this manuscript and to assist with implementation code such as helper and plotting scripts. They were not used to develop the method or its theoretical framing, to formulate or prove the propositions, to design the experiments, to generate, clean or reformat data, or to interpret the results. All text and code were checked by the authors.

References

Brian D. O. Anderson and John B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.

J. M. Bates and C. W. J. Granger. The combination of forecasts. *Operational Research Quarterly*, 20 (4):451–468, 1969. doi: 10.1057/jors.1969.103.

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.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Yuerong Xue, Shu-Tao Xia, and Zexuan Zhu. DDN: Dual-domain dynamic normalization for non-stationary time series forecasting. In *Advances in Neural Information Processing Systems*, 2024.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. ST-Norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 269–278, 2021. doi: 10.1145/3447548.3467330.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. AdaRNN: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pp. 402–411, 2021. doi: 10.1145/3459637.3482315.

James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2 edition, 2012.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-TS: A general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7522–7529, 2023. doi: 10.1609/aaai.v37i6.25914.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. SIN: Selective and interpretable normalization for long-term time series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.

Jeonghwan Im and Hyuk-Yoon Kwon. COSA: Context-aware output-space adapter for test-time adaptation in time series forecasting. In *International Conference on Learning Representations*, 2026.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17868–17876, 2025.

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 *International Conference on Learning Representations*, 2022.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, 2022.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Information Processing Systems*, 2023a.

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

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 *Advances in Neural Information Processing Systems*, 2023b.

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

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven C. H. Hoi. Learning fast and slow for online time series forecasting. In *International Conference on Learning Representations*, 2023.

Syama Sundar Rangapuram, Matthias W. Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, pp. 7785–7794, 2018.

Edoardo Urettini, Daniele Atzeni, Reshawn J. Ramjattan, and Antonio Carta. GAS-Norm: Score-driven adaptive normalization for non-stationary time series forecasting in deep learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2282–2291, 2024. doi: 10.1145/3627673.3679822.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations*, 2023.

Yuyang Wang, Alex Smola, Danielle C. Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6607–6617, 2019.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. ETSformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. In *Advances in Neural Information Processing Systems*, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Yifan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. OneNet: Enhancing time series forecasting models under concept drift by online ensembling. In *Advances in Neural Information Processing Systems*, 2023.

Lifan Zhao and Yanyan Shen. Proactive model adaptation against concept drift for online time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 2020–2031, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

A Longer Input Window

This comparison lengthens the backbone window from $L = 96$ to $L = 336$ for DLinear and iTransformer on ETTh1, ETTm2, Weather and ECL at $H \in \{96, 720\}$. The test targets are the same for both window lengths, and each cell's filter remains unchanged. Every method keeps its $L = 96$ learning rate. RevIN has no further hyperparameters, and the one structural hyperparameter of each of the other two adapters is reselected at $L = 336$ by the rule of Appendix G, over $K \in \{2, 4, 8\}$ for TRACE and over the kernel length $\{7, 12, 24\}$ for DDN. Reselection changes K on ETTh1 for both backbones and on ECL for DLinear. Only the ETTh1 changes matter: they lower the error by 2.9–18.8% in all four ETTh1 cells, whereas on ECL the two values differ by at most 0.05%. The reselected kernel changes DDN error by at most 1.4%. Table 4 reports both versions of TRACE. Appendix K compares the cost of the two window lengths.

Table 4: Test MSE with the backbone window lengthened to $L = 336$. The kernel length of DDN and the K of TRACE are reselected at $L = 336$, and K *from* L=96 keeps the value selected at the shorter window. Bold marks the lowest value among RevIN, DDN and the reselected TRACE.

					TRACE	
Backbone	Data	H	RevIN	DDN	K reselected	K from L=96
DLinear	ETTh1	96	0.3767	**0.3705**	0.3708	0.3837
		720	**0.4442**	0.4581	0.4920	0.6058
	ETTm2	96	**0.1632**	0.1638	0.1642	0.1642
		720	0.3666	**0.3641**	0.3693	0.3693
	Weather	96	0.1743	**0.1507**	0.1588	0.1588
		720	0.3311	0.3237	**0.2973**	0.2973
	ECL	96	0.1467	**0.1329**	0.1386	0.1386
		720	0.2161	**0.2008**	0.2085	0.2084
iTransformer	ETTh1	96	0.4062	**0.3759**	0.4264	0.4390
		720	**0.5095**	0.5636	0.6168	0.6498
	ETTm2	96	0.1720	**0.1685**	0.1810	0.1810
		720	**0.3730**	0.3731	0.4002	0.4002
	Weather	96	0.1617	**0.1558**	0.1598	0.1598
		720	0.3274	0.3193	**0.2961**	0.2961
	ECL	96	0.1319	0.1338	**0.1292**	0.1292
		720	0.2079	**0.1957**	0.2083	0.2083

B Combination with Test-time Adaptation

Test-time adapters update parameters from the realized targets of earlier forecasts, whereas TRACE changes no parameters at test time, so the two can be stacked. As a diagnostic of this combination, we apply TAFAS (Kim et al., 2025) to the checkpoints of RevIN, SAN, DDN and TRACE for DLinear, TSMixer and iTransformer on ETTh1, ETTm2 and Weather at $H \in \{96, 720\}$, with one adaptation learning rate for every method and cell. Both variants of a method use the same checkpoint, so the effect of adaptation is a contrast within one checkpoint, and these 18 cells, each from a single training run, do not replace the main comparison. TAFAS lowers the error of every method in 10 to 14 of the 18 cells, but least for TRACE, with a median gain of 0.18% against 0.36–1.28% for the three baselines (Table 5), which is consistent with a partial overlap between what test-time adaptation corrects and what the state supplies. Stacking does not remove the contribution of the state: with TAFAS applied to every method, TRACE remains lower than RevIN, SAN and DDN in 13, 12 and 12 of the 18 cells, against 13, 12 and 13 without adaptation. The costs differ in kind. TAFAS raises the median time per forecast origin from 0.08–0.14 ms to 0.46–0.64 ms and the median peak memory from 14–53 MB to 117–251 MB, because it backpropagates at test time on targets that arrive only after each forecast, whereas the state is updated in closed form from each arriving observation.

C Additional Diagnostics of the Combination

Table 6 compares a configuration in which the backbone is fixed at an independent checkpoint and only the combination is trained against the jointly trained configuration. The fixed configuration is

Table 5: Effect and cost of applying TAFAS to each method on 18 cells. The gain is $100\,(\mathrm{MSE}_{\mathrm{none}} - \mathrm{MSE}_{\mathrm{TAFAS}})/\mathrm{MSE}_{\mathrm{none}}$ in each cell. Time per forecast origin and peak memory at test time are medians over the cells.

	Gain from TAFAS		Time (ms)		Peak memory (MB)	
Method	Cells improved	Median (%)	None	TAFAS	None	TAFAS
RevIN	12/18	0.54	0.08	0.46	14.3	116.6
SAN	14/18	1.28	0.09	0.53	15.6	117.2
DDN	12/18	0.36	0.14	0.64	17.2	124.9
TRACE	10/18	0.18	0.09	0.48	53.3	250.5

Table 6: Test MSE when only the combination is trained on a backbone fixed at an independent checkpoint, against joint training; both mix the same two branches. *Difference* is $100\,(\mathrm{joint} - \mathrm{fixed})/\mathrm{joint}$, so positive values favor the fixed backbone.

Backbone	Data	H	Joint training	Backbone fixed	Difference (%)
DLinear	ETTh1	720	0.4843	0.4824	+0.41
		96	0.3986	0.3980	+0.15
	ETTm2	720	0.4031	0.4039	-0.21
		96	0.1771	0.1778	-0.42
	Weather	720	0.3562	0.3587	-0.68
		96	0.1716	0.1700	+0.94
TSMixer	ETTh1	720	0.5281	0.5234	+0.88
		96	0.4296	0.4268	+0.66
	ETTm2	720	0.4038	0.4029	+0.22
		96	0.1786	0.1802	-0.92
	Weather	720	0.3495	0.3491	+0.11
		96	0.1676	0.1653	+1.38

Table 7: Relative spread (%) of the learned weight along each axis: the standard deviation of the weight averaged over the other two axes, divided by the overall mean weight, as the median over the four horizons of each pair.

Backbone	Data	Origin	Lead	Channel
DLinear	ETTh1	9.17	0.657	18.52
	ETTm2	3.99	0.070	8.26
	Weather	2.91	0.017	9.00
TSMixer	ETTh1	12.66	0.451	23.25
	ETTm2	0.95	0.046	1.19
	Weather	1.14	0.008	3.22

lower in 8/12 cells, with a median difference of $+0.19\%$ and a range from -0.92% to $+1.38\%$. The difference observed when the same setting is rerun with a different random seed overlaps this range, so we read only the direction of this contrast and do not use its magnitude in the conclusions. These differences are smaller than the 2.22–8.52% oracle-mixture gap in the four ETTh1 cells, so joint backbone training does not explain that gap.

The weight of the best constant mixture that creates the gap in Table 13, the $w^\star$ of Proposition 2, is 0.059–0.431 (median 0.183) and is small in cells where the structural forecast is weak. We also decompose the learned weights $w_{t,h}$ of the selected configurations for DLinear and TSMixer on ETTh1, ETTm2, and Weather at the four horizons. For each axis, the relative spread is the standard deviation of the weight averaged over the other two axes, divided by the overall mean weight. The median relative spread is 5.91% across channels, 3.64% across origins and 0.07% across leads; the channel is the largest axis in 22 of the 24 cells and the origin in the other two (Table 7). Since g_ϕ shares its weights across channels, this variation comes entirely from differences in the input $\mathbf{A}^h\mathbf{s}_t$, and along the lead direction the weight is nearly constant. In these configurations, the mean weight is 0.34–0.87 (median 0.53), whereas in the factor experiments the median of the cap a after training is 0.138.

Table 8: Learned gate against a scalar weight fitted on the validation split and held fixed on test. The gain is $100\,(\mathrm{MSE}_{\mathrm{scalar}} - \mathrm{MSE}_{\mathrm{learned}})/\mathrm{MSE}_{\mathrm{scalar}}$, so positive values favor the learned gate.

Backbone	Learned lower	Mean gain (%)	Median gain (%)
DLinear	6/12	$+0.15$	$+0.01$
TSMixer	4/12	-2.92	-0.15
All	10/24	-1.38	-0.06

Table 9: Branches combined with one weight fitted without the test segment, as medians over the 12 factor cells; unlike Table 3, the backbone is trained without the state and the weight is a constant. *Gain* and C_{BD}/R_B are as in Table 3; *Short* is the carried state without components of period at least L, *Full* keeps all $K = 8$ harmonics, and *Short higher* counts the cells in which Short has the larger gain.

	Branch	Gain (%)	C_{BD}/R_B	Short higher
Window only	Mean	0.09	0.97	12/12
	Fourier (top 4)	0.10	0.96	12/12
	Seasonal naive	0.32	0.94	12/12
	Ridge, filter frequencies	1.18	0.96	12/12
Carried	Level only	0.92	0.93	10/12
	Short: periods $< L$	**8.48**	0.78	–
	Full: fixed K=8	2.12	0.78	10/12

Learned versus validation-fitted scalar gate. We compare the learned gate with a single mixture weight fitted on the validation split and held fixed on test, keeping the checkpoint and the structural forecast of each run unchanged, so that no test outcome enters the fit of the scalar weight. Table 8 summarizes the 24 cells formed by DLinear and TSMixer on ETTh1, ETTm2 and Weather at the four horizons, each at its selected setting. The two are within 0.06% at the median, so the contribution of the state does not depend on the flexibility of the gate: a single weight fitted on the validation split recovers essentially the same gain.

Other branches under the same combination. Table 9 summarizes this comparison over the 12 factor cells and Table 10 gives it cell by cell. To test whether any forecast that differs from the backbone would serve, we attach seven branches to the independently trained backbone of Table 13 and combine each with one weight $g = \mathrm{clip}\big(\sum(D - B)(y - B)/\sum(D - B)^2, 0, 1\big)$ fitted on the validation split and applied unchanged to the test segment. A single weight is used because it performs about as well as the learned gate (Table 8) and removes the variation of gate training from the comparison. The window-only branches are the four estimators of Figure 1, and the carried branches are the level-only state ($K = 0$), the factor state with every harmonic of period at least L removed, and the full factor state. The carried state without long components outperforms the best window-only branch in all 12 cells (median $+6.40$ pp), and the full factor state does so in 8 of 12, including every ETTm2 and Weather cell (median $+1.05$ pp). The removed harmonics have periods of at least 96 observations in the fixed $K = 8$ of the factor filter, and the full state is ahead of the shortened one only on Weather at $H = 96$, where the removed set contains the 144-step daily cycle. The level-only state is ahead on ETTm2 and Weather at $H = 720$, the same cells as in the trained comparison of Appendix P, and on ETTh1 its validation-fitted weight transfers poorly to the test segment. The first effect is the robust one: carrying the components a window can hold beats window-only estimation in all 12 cells, while the components longer than the window add a dataset-dependent benefit, as on Weather at $H = 96$ here and on the datasets of Figure 1 at their selected K.

D NOTATION

Every symbol is defined where it is first used; this table is for lookup.

Table 10: Gain (%) over the independently trained backbone when each branch is combined with one weight fitted on the validation split. *Window* is the best of the four window-only branches, marked M (mean), F (Fourier), S (seasonal naive) or R (ridge); *Short* is the carried state without harmonics of period at least L.

		DLinear				TSMixer			
Data	H	Window	Level	Short	Full	Window	Level	Short	Full
ETTh1	96	2.42^{S}	-0.58	3.83	0.18	7.15^{R}	-0.04	8.29	4.45
	720	1.52^{R}	-8.53	11.61	1.35	8.64^{R}	-3.74	18.17	7.19
ETTm2	96	0.29^{R}	0.45	3.09	2.44	1.17^{R}	1.29	3.95	2.89
	720	0.12^{F}	1.90	8.67	0.81	0.52^{R}	2.67	9.23	0.90
Weather	96	1.24^{R}	1.65	5.50	13.61	1.31^{R}	0.55	4.30	10.29
	720	0.27^{M}	13.62	9.91	1.79	0.26^{R}	12.44	9.22	1.66

Symbol	Meaning	Defined in
$\mathbf{x}_t \in \mathbb{R}^C$, x_t	observation at time t over C channels; x_t is the entry of one channel	Sec. 3
L, H, h	backbone input window; forecast horizon; lead index $h = 1, \dots, H$	Sec. 3
$\widehat{\mathbf{Y}}^{\mathrm{base}}$, $\widehat{x}^{\mathrm{base}}_{t+h}$	backbone forecast, and its entry for one channel and lead	Sec. 3, Eq. (5)
$\mathbf{s}_t \in \mathbb{R}^d$, $\mathbf{s}_t^-$	structural state of one channel, and its prediction before the correction step	Sec. 3.3
$d = 1 + 2K$, K	state dimension; number of harmonic components	Sec. 3.3
ω_k	angular frequency of harmonic k, from the top-K periodogram peaks	Sec. 3.3
$\mathbf{A}$, $\mathbf{c}$, $\mathbf{K}_g$	transition, readout and steady-state Kalman gain of the fixed filter	Sec. 3.3
ρ	spectral radius of the closed-loop transition $(\mathbf{I} - \mathbf{K}_g\mathbf{c}^\top)\mathbf{A}$	Sec. 3.3
$\widehat{x}^{\mathrm{state}}_{t+h}$	structural forecast $\mathbf{c}^\top \mathbf{A}^h \mathbf{s}_t$	Eq. (3)
g_ϕ, a, $w_{t,h}$	gate network, its global cap, and the combination weight it produces	Eq. (4)
$\widehat{x}_{t+h}$	combined forecast	Eq. (5)
M	history the state may absorb before the origin; *continuous* when unrestricted	Sec. 4.3
R_B, R_D, C_{BD}	backbone MSE, structural-branch MSE, and the mean product of the two branch errors	Sec. 4.3
cell	one combination of backbone, dataset and forecast horizon	Sec. 4.1

E THE COMPOSITION AND ROLLOUT OF THE STATE

Figure 3 shows what the structural state of Sec. 3 holds in one cell and how it is rolled out. The two panels come from the same run with the same backbone, dataset and forecast horizon. The summed components follow the slow periodic part of the series, and the range of the undamped rollout grows from 0.28 in the first quarter of the horizon to 0.49 in the last, which is why its weight is bounded (Sec. 3.4).

Figure 1 and Table 11 score H-step structural forecasts against the observations in the training-standardized coordinates, averaged over leads, channels and test origins taken every eight observations. The carried state is the fixed filter of Sec. 3.3 run along the full series, and its variant without long components zeroes, before the rollout, every harmonic block whose period is at least L. Window-only forecasts use only the L observations preceding the origin: the window mean held over the horizon; the four largest non-constant discrete Fourier components of the mean-removed window, extrapolated periodically; a seasonal naive forecast that repeats the last cycle of the longest selected period shorter than L; and a ridge regression of the window on a level and the filter's own frequencies. The ridge penalty applies only to the components whose period is at least L and is selected on validation origins whose targets lie inside the validation segment; as the penalty grows, the fit reduces to least squares on the level and the components shorter than the window, so the selected penalty cannot do worse on validation than ignoring the longer frequencies. Because the ridge fit is given frequencies that

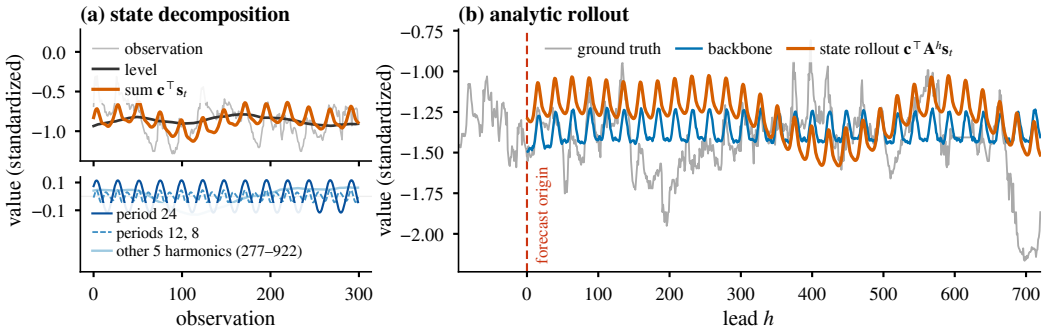

Figure 3: Composition and rollout of the structural state for one channel of ETTh1 at $H{=}720$, from one run. **(a)** Component decomposition over 300 observations; the harmonic components are zero-mean and drawn on a separate axis, with periods 12 and 8 summed and the five components with periods of 277–922, which do not complete a cycle within the segment, drawn as one line. **(b)** Backbone forecast and state rollout at one forecast origin.

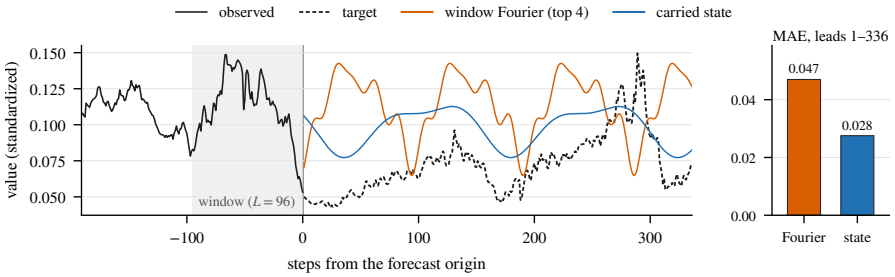

Figure 4: Structural forecasts at one test origin of Weather ($L{=}96$; the first 336 of the $H{=}720$ leads are shown for legibility). The shaded band is the input window, the solid line the observations before the origin and the dashed line the target. The orange line is a window-only forecast that extrapolates the four largest Fourier components of the mean-removed window, and the blue line is the state carried over the whole history; values are in training-standardized units and the bars give the mean absolute error of each forecast over the leads shown.

a window-only method does not have, it serves as an upper reference for window estimators rather than as a practical baseline. Each dataset uses the K selected in the main table; where the selected K differs across backbones, every such K is reported. Figure 1 uses $K{=}8$ on ECL, the value selected for three of its four backbones. On the ETT datasets the ridge fit is lowest at $H{=}96$ for ETTh1, for ETTm1 with $K = 4$, and for ETTm2 at $H = 720$, whereas the carried state without long components is lowest in most of the remaining rows. The example in Figure 4 uses the target channel at the test origin at which the carried state's own error is the median over test origins, a rule fixed before inspection.

F Recovery After Structural Changes

Synthetic protocol. We generated seven-channel series of length 46,811 from eight harmonic components with periods $\{24, 48, 96, 168, 240, 336, 512, 672\}$, channel-specific amplitudes $\mathcal{U}(0.2, 1.0)$ and phases $\mathcal{U}(0, 2\pi)$, and AR(1) residuals $r_t = 0.6r_{t-1} + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, 0.2^2)$. The train, validation, and test lengths are 32,767, 4,682, and 9,362. For each horizon, the change occurs two horizons after the first test observation. We evaluate eight matched residual realizations; conditions within a realization share the same noise and are identical before the change. The five changes are a permanent $+1$-standard-deviation level shift, a ramp reaching $+1$ standard deviation over H observations, a $1.5\times$ amplitude change, a $\pi/2$ phase shift, and a $1.1\times$ frequency change with phase continuity.

Table 11: H-step MSE of window-only and carried-state structural forecasts on all seven datasets, with L=96. K is the number of harmonic components selected in the main table, and the letters give the backbones that use it (D: DLinear, T: TSMixer, M: MICN, I: iTransformer). Bold marks the lowest value in each row; "Cycles $\geq L$ removed" zeroes the harmonic components whose period is at least L, and "–" marks a cell with no selected period shorter than L.

			Window only				Carried state	
Data	K	H	Mean	Fourier	S. naive	Ridge	Cycles $\geq L$ removed	Full
Weather	2 (DTMI)	96	0.215	0.318	0.324	0.238	0.204	**0.188**
		720	0.381	0.464	0.472	0.394	0.340	**0.324**
ECL	2 (T)	96	0.846	0.324	0.321	0.267	**0.242**	**0.242**
		720	0.892	0.350	0.367	0.316	**0.294**	**0.294**
	8 (DMI)	96	0.846	0.324	0.321	0.266	0.197	**0.186**
		720	0.892	0.350	0.367	0.300	0.256	**0.240**
Traffic	8 (DTMI)	96	1.411	1.067	1.218	0.942	0.661	**0.634**
		720	1.452	0.969	1.186	0.872	0.679	**0.648**
ETTh1	4 (DTM)	96	0.701	0.519	0.512	**0.419**	0.449	0.484
		720	0.712	0.584	0.655	0.503	**0.446**	0.502
	8 (I)	96	0.701	0.519	0.512	**0.411**	0.464	0.518
		720	0.712	0.584	0.655	0.492	**0.439**	0.569
ETTh2	2 (DTM)	96	0.352	0.424	0.390	0.324	**0.312**	0.321
		720	0.451	0.505	0.526	0.430	0.401	**0.397**
	4 (I)	96	0.352	0.424	0.390	0.320	**0.311**	0.327
		720	0.451	0.505	0.526	0.433	**0.394**	0.461
ETTm1	4 (DTM)	96	0.693	0.397	0.423	**0.387**	0.423	0.452
		720	0.746	0.550	0.574	0.543	**0.411**	0.415
	8 (I)	96	0.693	0.397	0.423	0.392	**0.337**	0.360
		720	0.746	0.550	0.574	0.546	**0.409**	0.432
ETTm2	2 (DTMI)	96	0.229	0.242	–	0.207	0.208	**0.186**
		720	0.433	0.450	–	**0.427**	0.442	0.439

Table 12: MSE after a synthetic change whose type, size, and time point were fixed in advance. The three policies use the same checkpoint and differ only in how the state is handled on the test segment. Values are means over eight matched residual realizations; Appendix F gives the generator and change protocol.

		DLinear						TSMixer					
H	State policy	No change	Level	Slope	Amplitude	Phase	Frequency	No change	Level	Slope	Amplitude	Phase	Frequency
96	Correction on	0.3148	0.3495	0.4072	0.7675	0.9756	0.9189	0.2700	0.3486	0.3786	0.6127	1.1103	0.9164
	Correction off	0.3151	0.3655	0.4396	0.7806	0.9885	0.9260	0.2703	0.3552	0.4120	0.6246	1.1183	0.9238
	Rebuild from window	0.5211	0.5273	0.5595	1.1220	0.7737	1.0089	0.4209	0.4625	0.4797	0.8882	0.9118	0.9577
720	Correction on	0.4289	0.5953	0.7442	0.9988	0.6770	1.5314	0.4187	0.5835	0.7302	0.9597	0.6893	1.5219
	Correction off	0.4306	0.5545	0.7415	1.0688	0.8332	1.4651	0.4203	0.5421	0.7271	1.0282	0.8500	1.4552
	Rebuild from window	0.6708	0.8025	0.9830	1.4423	0.7172	1.5135	0.6567	0.7861	0.9647	1.3955	0.7312	1.5010

Recovery. In the experiment of Table 12, the recovery time is the number of elapsed origins after which the error of a policy enters and stays within the band of the mean plus two standard deviations of the same policy on the paired no-change stream. On a level change all three policies recover within 91 origins, and at $H = 96$ immediately. On a phase change the policy that keeps correcting takes 149–395 origins, while the uncorrected policy takes 163–8,498 and two of the eight TSMixer realizations at $H = 720$ do not recover. On amplitude and frequency changes the excess error of every policy stays positive (Figure 5), so we compare these conditions by the size of the excess rather than by recovery time.

The policy that rebuilds the state from the window recovers fastest after a phase change because it does not carry an outdated phase. Its absolute post-change MSE is the lowest of the three policies at H=96 and lies between correction on and correction off at H=720; recovery time and absolute error therefore need not induce the same ordering.

Figure 5 plots the excess error of the same experiment against the time elapsed since the change. A vertical value of zero means the change left no trace at all, that is a complete recovery. Over the one horizon immediately following the change, the policy that keeps correcting is already 18.8–18.9% lower than the uncorrected one on the phase change, while it is 0.4–7.6% higher on the level and

slope changes. The larger differences arise afterwards. On a level shift and a slope change, only the policy that keeps correcting returns to zero while the other two remain positive (late phase: -0.004 versus $+0.084$ for both changes). The two changes share their late-phase values because the slope ramp ends after H observations, after which the slope condition is a level shift of the same size. On an amplitude change none of the three policies returns to zero and correction only narrows the gap (26.8–28.0% lower than the uncorrected policy in the middle and late phases). On a frequency change the policy that keeps correcting is the worst of the three over the first horizon and in the late phase, while rebuilding from the window is the best throughout, so absorbing observations into a state with the wrong basis does not pay off.

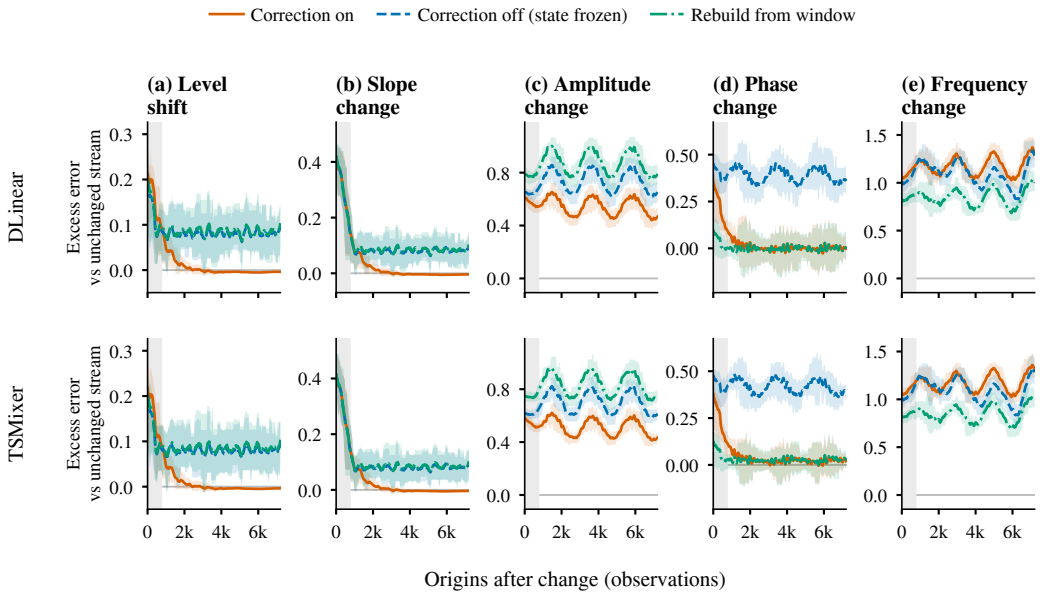

Figure 5: Excess error after each synthetic change at $H{=}720$, against a paired stream that shares the prefix but has no change, so zero means a complete recovery. Curves are moving averages over ± 360 observations (one horizon) and bands the 25–75 percentiles over the same window, showing local variation along time rather than uncertainty across realizations. The shading marks the first horizon after the change; the vertical range differs by column, and the two rows of a column share an axis.

G Implementation and Reproduction

Methods left out of the comparison. GAS-Norm (Urettini et al., 2024) is also left out because its public implementation targets univariate probabilistic forecasters in GluonTS and would need to be reimplemented for the multivariate point-forecasting protocol used here. Under a Gaussian score its mean path is an exponentially weighted level with mean reversion; its closest analog here, the carried level-only state, gains 0.92% at the median, against 2.12% for the full carried state and 8.48% for its components shorter than the window (Table 9).

Selection protocol. The protocol has three stages: for each candidate K the frequencies and noise ratios of the filter are configured on the training and validation segments, the backbone and the gate are then trained with the filter frozen while the learning rate and K are selected, and the selected model forecasts the test segment origin by origin, with $\mathbf{s}_t$ depending only on $x_{\leq t}$ and no test value entering any stage of configuration or selection.

Data and splits. The seven datasets of the main benchmark and Exchange use the public versions from the long-term forecasting literature; the channel counts are 7 for the ETT family, 8 for Exchange, 21 for Weather, 321 for ECL and 862 for Traffic. We use training, validation and evaluation splits that preserve temporal order, and normalization statistics are computed on the training segment only. ETTh1 and ETTh2 are split into 12, 4 and 4 months (8,640, 2,880 and 2,880 observations), ETTm1

and ETTm2 into the same months at four times the resolution (34,560, 11,520 and 11,520), and ECL, Traffic, Weather and Exchange into 70%, 10% and 20% in time order. Each validation and test window may take its input from the last L observations of the preceding segment, but every target lies inside its own segment. The input length is 96 in every cell except in the comparison of Appendix A, which uses 336.

The structural state. The state consists of one level per channel and K harmonic components, so its dimension is $1 + 2K$; K is chosen per (backbone, dataset) pair and its values are in Table 21. The ranked frequency candidates are obtained once per dataset by detrending the combined training and validation segments, computing their periodograms, and summing across channels; the selected K determines how many candidates are retained. Period candidates run from 4 observations to $\min(1024, \text{training segment length}/4)$ observations. The selected periods are in Table 14. The filter operates on the series standardized with training statistics, not on window-normalized inputs, and at combination time the backbone output has been denormalized by the window statistics, so the two branch forecasts live in the same space. For each forecast horizon, the level and harmonic process-noise ratios are selected from the 16 combinations of $\{10^{-6}, 10^{-4}, 10^{-2}, 1\}$ by the H-step error on the combined training and validation segments, while the observation-noise variance is fixed at 1. With $\mathbf{Q} = \mathrm{diag}(q_\ell, q_h, \ldots, q_h)$ built from the two ratios and $r = 1$, the prior covariance is iterated from $\mathbf{P}_0 = 10^4\mathbf{I}$ under

$$\mathbf{P} \leftarrow \mathbf{A}\big(\mathbf{P} - \mathbf{P}\mathbf{c}(\mathbf{c}^\top\mathbf{P}\mathbf{c} + r)^{-1}\mathbf{c}^\top\mathbf{P}\big)\mathbf{A}^\top + \mathbf{Q}$$

until convergence, and the gain is $\mathbf{K}_g = \mathbf{P}\mathbf{c}/(\mathbf{c}^\top\mathbf{P}\mathbf{c}+r)$. The resulting filter is fixed before backbone training. The selected pair is the same at all four horizons on ECL and Traffic and varies with the horizon on the other five datasets. The pair $(\mathbf{A}, \mathbf{c}^\top)$ is observable whenever the K frequencies are distinct and lie in $(0, \pi)$, and the process noise is positive on every component; under these conditions the discrete Riccati recursion started from a diffuse prior converges to the unique stabilizing solution and $\mathbf{K}_g$ is its steady-state gain (Anderson & Moore, 1979). Both conditions hold for every selected filter: the selected periods are distinct and every noise ratio is at least 10^{-6} (the periods of the factor cells are listed in Table 14). The initial state is a zero prior, and the state is computed along a single time axis on which the three segments are concatenated, without reinitialization at the segment boundaries. The state update at forecast time uses only values already observed up to that origin; values in the forecast horizon do not enter the update.

Design checks of the filter. Two alternatives were evaluated at the selected learning rate of each pair, for DLinear and TSMixer on ETTh1, ETTm2 and Weather at the four horizons. First, damping every harmonic component with a factor $\rho < 1$ whose half-life is 96, 336 or 720 observations was offered alongside the undamped filter and selected by the rule of the setting search; the undamped filter was chosen in all six backbone–dataset pairs. Second, we replaced the two periodogram periods at $K = 2$ by the daily and half-daily periods. On ETTh1 and Weather the periodogram already selects these two, so the filters coincide; on ETTm2, where it selects a period of 963 observations instead of the half-daily one, the calendar pair is lower by 1.1–4.7% at $H \le 336$ but higher by 1.2–7.6% at $H = 720$, so neither choice dominates.

Setting search. Only the learning rate and the number of harmonic components K are tuned for the proposed method. For every backbone–dataset pair, we exhaustively evaluate

$$\{10^{-4},\, 5\times10^{-4},\, 10^{-3},\, 5\times10^{-3}\} \times \{2, 4, 8\},$$

for 12 configurations in total. Each configuration θ is trained separately at $H \in \{96, 192, 336, 720\}$. Let $V(\theta, H)$ denote the validation MSE at the checkpoint selected by early stopping, and $r(\theta, H)$ its rank among the 12 configurations at horizon H. We select

$$\theta^\star = \arg\min_\theta \frac{1}{4} \sum_{H\in\{96,192,336,720\}} r(\theta, H),$$

break ties by the smaller worst-horizon validation MSE $\max_H V(\theta, H)$, and use $\theta^\star$ unchanged at all four horizons. Ranking gives each horizon the same weight even though their validation errors differ in scale. The selection uses the runs of the first seed, and the selected configuration is kept for every reported seed. The validation segment is therefore used both for checkpoint selection

Table 13: Combination of the two forecasts. R_B is the error of the backbone alone and R_D that of the structural forecast alone; R_D is independent of the backbone and is therefore given a single column. C_{BD} is the cross moment of the two errors, *Comb.* is the MSE of the learned combination, and *Gap* is the relative error against the best constant mixture fitted to the evaluation segment, so a negative value means the learned combination is worse than that oracle.

		Struct.	DLinear				TSMixer			
Data	H	R_D	R_B	C_{BD}	Comb.	Gap (%)	R_B	C_{BD}	Comb.	Gap (%)
ETTh1	96	0.4960	0.4004	0.3560	0.3986	-2.22	0.4328	0.3581	0.4296	-5.36
	720	0.5684	0.4877	0.4134	0.4843	-4.27	0.5322	0.4183	0.5281	-8.52
ETTm2	96	0.4612	0.1816	0.1404	0.1771	-0.09	0.1849	0.1387	0.1786	+0.28
	720	1.3426	0.4068	0.3444	0.4031	-0.00	0.4063	0.3398	0.4038	-0.40
Weather	96	0.3595	0.1954	0.0962	0.1716	-1.94	0.1858	0.0998	0.1676	-1.90
	720	1.2134	0.3645	0.2829	0.3562	+0.47	0.3559	0.2783	0.3495	+0.15

and configuration selection; the baselines also use it for early stopping. No test value enters either selection. The input length $L = 96$, the maximum of 10 epochs, patience 3, the MSE loss, and the structural form of the adapter are fixed across configurations. The selected configurations are reported in Table 21.

Gate initialization. Because the last layer of g_ϕ and a start at zero, at initialization the gradient reaches a but not g_ϕ, since $\partial w/\partial a = \sigma(0) = 1/2$ while $\partial w/\partial g_\phi \propto a = 0$; training therefore first opens the structural branch through a and only then lets g_ϕ shape it. The clamp is implemented straight-through, so the gradient reaches a at any value and a closed gate can reopen.

Training. All methods are trained for at most 10 epochs with patience 3. The main benchmark trains every method under three seeds, with method-specific initializations rather than paired initial states. For every cell, the reported mean and sample standard deviation use the intersection of seeds available for all six methods; this intersection contains all three seeds in every one of the 112 cells. SAN and DDN reproduce the original authors' multi-stage schedules exactly. SAN trains the statistics forecasting module alone for 5 epochs, reverts to the best checkpoint by the validation statistics loss and then freezes it; DDN trains alone for 5 epochs and then, after a one-epoch delay, enters joint fine-tuning at the backbone learning rate. FAN uses the learning rate 3×10^{-4} of the original authors' script and the per-dataset `freq_topk`. iTransformer is used with its own normalization layer turned off.

The released code reproduces the main benchmark. It contains the implementation of TRACE and of the five baselines, the configurations of Table 21 together with the learning rates selected for the baselines on Exchange, and a script that runs every cell of the main tables under the three seeds with exactly these settings.

H CLOSED-LOOP DYNAMICS OF THE FILTER

Table 14 lists, for the cells of the factor analyses, the selected periods and the spectral radius of the closed-loop matrix $F = (I - \mathbf{K}_g \mathbf{c}^\top)\mathbf{A}$. Each set contains the daily cycle and its sub-daily harmonics at the sampling interval of the dataset, and the remaining periods are long-period components of several hundred observations. Since the radius is 0.99906–0.99961, the half-life of the slowest mode is 738–1,791 observations, that is 7.7–18.7 times the input window. This value states how long a component of the state persists; it is not the same as the decay of the influence of past observations on the forecast. The latter passes through $F^j \mathbf{K}_g$ and the readout together, so a slow mode may be driven only weakly by the observations.

I TWO ELEMENTARY PROPERTIES OF THE COMBINATION

The two statements below are called from Sec. 3.4 and Sec. 4.3; both follow from the fact that Eq. (5) is a convex combination of the two branch forecasts.

Table 14: Closed-loop dynamics of the fixed filter and the periods held by the state, for the cells of the factor analyses. $\rho(F)$ is the spectral radius of $F = (\mathbf{I} - \mathbf{K}_g \mathbf{c}^\top)\mathbf{A}$, and the half-life is $\ln 2/(-\ln \rho(F))$ observations, stating how long the slowest mode persists. The last column lists the K selected periods. † marks the synthetic series of Appendix F rather than a benchmark dataset.

Dataset	H	$\rho(F)$	Half-life (obs.)	vs. L=96	Selected periods (obs.)
ETTh1	720	0.99927	955	9.9×	24, 12, 650, 524, 396, 922, 8, 277
	96	0.99961	1,791	18.7×	24, 12, 650, 524, 396, 922, 8, 277
ETTm2	720	0.99917	832	8.7×	96, 963, 702, 826, 496, 48, 408, 594
	96	0.99917	832	8.7×	96, 963, 702, 826, 496, 48, 408, 594
Synth†	720	0.99923	903	9.4×	24, 96, 512, 168, 336, 672, 48, 240
	96	0.99923	903	9.4×	24, 96, 512, 168, 336, 672, 48, 240
Weather	720	0.99906	738	7.7×	144, 72, 960, 675, 513, 820, 445, 371
	96	0.99922	887	9.2×	144, 72, 960, 675, 513, 820, 445, 371

Proposition 1 (Interpolation bound)**.** *For any channel, origin and lead, if $w \in [0,1]$ then $\hat{x} = \hat{x}^{\text{base}} + w(\hat{x}^{\text{state}} - \hat{x}^{\text{base}})$ satisfies $|\hat{x} - y| \le \max\{|\hat{x}^{\text{base}} - y|,\ |\hat{x}^{\text{state}} - y|\}$ for every target y; consequently the squared error of the combined forecast never exceeds the larger of the two branch errors, and equals the backbone's when $w = 0$.*

Proof. $\hat{x} - y = (1-w)(\hat{x}^{\text{base}} - y) + w(\hat{x}^{\text{state}} - y)$ is a convex combination of the two errors, and $|\cdot|$ is convex. □

Proposition 2 (A small structural weight helps iff $C_{BD} < R_B$)**.** *Let $e_B = \hat{x}^{\text{base}} - y$ and $e_D = \hat{x}^{\text{state}} - y$ over an evaluation segment, with $R_B = \mathbb{E}\, e_B^2$, $R_D = \mathbb{E}\, e_D^2$ and $C_{BD} = \mathbb{E}\, e_B e_D$, and assume the two branches are not identical, $\mathbb{E}(e_B - e_D)^2 > 0$. The MSE of the constant mixture with weight w is $J(w) = (1-w)^2 R_B + 2w(1-w)C_{BD} + w^2 R_D$, so $J'(0) = 2(C_{BD} - R_B)$: some $w > 0$ lowers the error if and only if $C_{BD} < R_B$, whatever R_D is. The best constant weight is $w^\star = (R_B - C_{BD})/(R_B - 2C_{BD} + R_D)$, with $J(w^\star) = R_B - (R_B - C_{BD})^2/(R_B - 2C_{BD} + R_D)$ (Bates & Granger, 1969).*

Proof. Expand $\mathbb{E}\big((1-w)e_B + we_D\big)^2$; J is a convex quadratic because its leading coefficient is $R_B - 2C_{BD} + R_D = \mathbb{E}(e_B - e_D)^2 > 0$, and the stated $w^\star$ is its stationary point. □

Because J is convex, the best weight within $[0,1]$ is $w^\star$ clipped to that interval.

J Separating the Affine from the Structural State

TRACE places the structural state on a backbone normalized by window statistics without an affine, while the RevIN baseline uses a learnable per-channel affine transform, so the contrast between the two contains both factors. To separate them, we treated the two as independent factors on the same 12 cells as the factor experiments of Sec. 4.3 and trained the four configurations with the same seed and the same initial backbone weights. In Table 15, the structural state lowers the error in 12/12 cells both with the affine off and with it on, and the interaction is at most 0.18 pp in absolute value, whereas the affine main effect ranges from -0.54 pp to $+1.30$ pp and is not consistent in sign. What the RevIN contrast of Sec. 4.2 reads out is therefore effectively the effect of the structural state.

K The Cost of the Adapter

Table 16 lists the learnable parameter count of each adapter. The per-dataset range reflects the spread of the K selected on that dataset and is unrelated to the number of channels. At the same K, all datasets take the same value: 226, 354 and 610 for $K = 2$, 4 and 8.

Table 17 lists the learnable parameters, including the backbone, and the peak training memory at $H = 96$. Lengthening the window adds 30,720–122,880 backbone parameters, while the adapter adds 226–610 at either length. The peak memory of TRACE grows by only 0.6–6.3% from $L = 96$ to

Table 15: A 2×2 design that treats the affine transform and the structural state as independent factors. The four MSE values come from four configurations trained from the same initial backbone weights, and *interaction* is the state effect with the affine on minus the state effect with it off.

Cell			State off		State on		Effect (%)		
Backbone	Data	H	aff off	aff on	aff off	aff on	affine	$\text{state}_{\text{aff off}}$	interaction
DLinear	ETTh1	720	0.4877	0.4904	0.4843	0.4870	-0.54	+0.69	-0.01
		96	0.4004	0.4003	0.3986	0.3985	+0.03	+0.46	-0.00
	ETTm2	720	0.4068	0.4068	0.4031	0.4031	-0.00	+0.90	+0.01
		96	0.1816	0.1817	0.1771	0.1771	-0.03	+2.50	+0.01
	Weather	720	0.3645	0.3597	0.3562	0.3523	+1.30	+2.26	-0.18
		96	0.1954	0.1932	0.1716	0.1699	+1.14	+12.20	-0.12
TSMixer	ETTh1	720	0.5322	0.5303	0.5281	0.5263	+0.36	+0.78	-0.03
		96	0.4328	0.4322	0.4296	0.4290	+0.14	+0.74	-0.00
	ETTm2	720	0.4063	0.4064	0.4038	0.4037	-0.02	+0.62	+0.04
		96	0.1849	0.1851	0.1786	0.1787	-0.09	+3.40	+0.01
	Weather	720	0.3559	0.3557	0.3495	0.3493	+0.06	+1.82	-0.01
		96	0.1858	0.1855	0.1676	0.1673	+0.15	+9.80	+0.01

Table 16: Learnable parameter count of each adapter. The backbone is identical across methods and is excluded. Ranges span the four forecast horizons and the settings selected for the four backbones of each dataset.

Dataset	C	TRACE	RevIN	FAN	SAN	DDN
ECL	321	226–610	642	39,842–120,338	113,290–166,590	1,459,009–1,778,497
ETTh1	7	354–610	14	39,214–119,710	112,662–165,962	139,527–299,271
ETTh2	7	226–354	14	39,214–119,710	112,662–165,962	139,527–299,271
ETTm1	7	354–610	14	39,214–119,710	124,958–231,558	270,599–430,343
ETTm2	7	226	14	39,214–119,710	124,958–231,558	30,791–70,727
Traffic	862	610	1,724	40,924–121,420	114,372–167,672	5,539,678–6,178,654
Weather	21	226	42	39,242–119,738	124,986–231,586	270,613–430,357

Table 17: Learnable parameters and peak training memory at $H = 96$. The three configurations of each backbone–dataset pair use the same batch size. The proposed method at $L = 336$ uses the K selected at $L = 96$.

		Parameters (backbone + adapter)			Peak memory (MB)		
		RevIN	TRACE		RevIN	TRACE	
Backbone	Data	L=336	L=96	L=336	L=336	L=96	L=336
DLinear	ETTh1	64,718	18,978	65,058	19.9	26.3	27.1
	ETTm2	64,718	18,850	64,930	20.0	26.0	26.7
	Weather	64,746	18,850	64,930	25.6	44.0	46.0
	ECL	65,346	19,234	65,314	150.4	475.0	505.0
iTransformer	ETTh1	903,022	842,178	903,618	45.4	46.3	47.0
	ETTm2	254,958	224,450	255,170	28.1	33.3	34.0
	Weather	4,956,810	4,834,114	4,956,994	172.5	170.6	171.6
	ECL	4,957,410	4,834,498	4,957,378	827.8	1,002.9	1,013.3

$L = 336$, because its largest intermediate, the gate evaluated for every lead and channel, does not depend on the window. The same term makes TRACE heavier than RevIN at the same window, by up to $3.4\times$ for DLinear on ECL, except for iTransformer on Weather, where the two are equal.

Figure 6 reads accuracy and adapter size together, as discussed in Sec. 4.2. The radius of a circle there is $r = 2 + 1.15 \log_{10} \max(p, 1)$ pt for the median learnable parameter count p, so its area is not proportional to the parameter count.

Figure 7 varies one axis at a time with the other two fixed, measuring the adapter path of a DLinear model with $L = H = 96$ and batch 32. The learnable parameter count is $32d + 66$, so it responds to K alone and is independent of both the number of channels and the history length. Peak memory grows linearly with the number of channels, by 1.39 MB per channel, because the gate is evaluated for every channel and lead, and changes by 3.1 MB along the K axis. Time per batch stays between 3.96

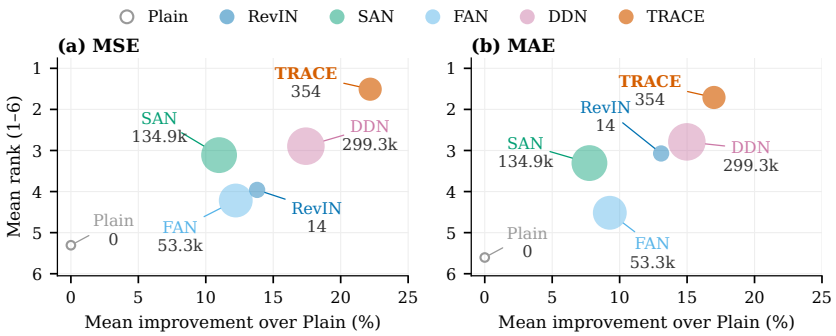

Figure 6: Accuracy and adapter size of each method over the 112 cells, for (a) MSE and (b) MAE. The horizontal axis is the mean improvement per cell over the plain backbone and the vertical axis the mean rank among the six configurations, best at the top; the radius of a circle grows with the median learnable parameter count, printed beside it.

Table 18: Adapter cost on ETTh1 ($C = 7$, $L = H = 96$, batch 32, adapter path only); time is the median over five independent processes.

Method	Parameters	Peak memory (MB)	Time (ms)
TRACE ($K = 8$)	610	26.07	4.08
RevIN	14	17.16	2.10
SAN	112,662	20.74	6.13
FAN	39,214	18.11	–
DDN	139,527	21.61	12.72

Table 19: History length M given to the state, with the backbone window at $L = 96$ in every column. *Restricted M at inference* intervenes on the same checkpoint, *Retrained M=96* is constrained to window-length history from training onwards, and *advantage of continuity* is the relative error reduction of the continuous state against each restricted condition. The full sweep including $M \in \{192, 384\}$ is in Figure 8.

		DLinear					TSMixer				
		Restricted M at inf.		Cont.	Advantage of cont. (%)		Restricted M at inf.		Cont.	Advantage of cont. (%)	
Data	H	96	768	state	inf.	retrain	96	768	state	inf.	retrain
ETTh1	96	0.4007	0.3998	0.3986	+0.52	+0.45	0.4321	0.4310	0.4296	+0.59	+0.65
	720	0.4869	0.4852	0.4843	+0.53	+0.54	0.5305	0.5289	0.5281	+0.45	+0.59
ETTm2	96	0.1823	0.1792	0.1771	+2.87	+2.24	0.1845	0.1810	0.1786	+3.20	+2.56
	720	0.4410	0.4380	0.4031	+8.60	+1.49	0.4321	0.4357	0.4038	+6.55	+3.10
Weather	96	0.1889	0.1758	0.1716	+9.19	+12.20	0.1796	0.1705	0.1676	+6.69	+9.21
	720	0.3478	0.3563	0.3562	**-2.41**	**-2.05**	0.3427	0.3496	0.3495	**-1.97**	**-1.73**

and 4.41 ms on every axis; on this GPU a fixed per-iteration overhead dominates the small adapter path, so we do not read differences below about 5%. The history-length column is flat because the state is built once before training, so the history length does not appear in the training or inference path; the cost of a longer history therefore lies in the preprocessing stage only, not in training or inference. Building the state bank is linear in the series length and takes at most 0.81 s, for Traffic on a CPU, once before training. Against the other adapters on ETTh1 (Table 18), TRACE takes 4.08 ms per batch, against 2.10 ms for RevIN, 6.13 ms for SAN and 12.72 ms for DDN; the time of FAN varied by more than 10% across processes and is not reported.

L EXCHANGE DATASET

Exchange (Lai et al., 2018) records the daily exchange rates of eight countries and has no dominant calendar cycle at its sampling interval. It is reported separately because the public scripts of SAN, FAN, and DDN do not cover it, so each of the five baselines receives a learning-rate search over $\{10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$, selected by the mean validation error across horizons,

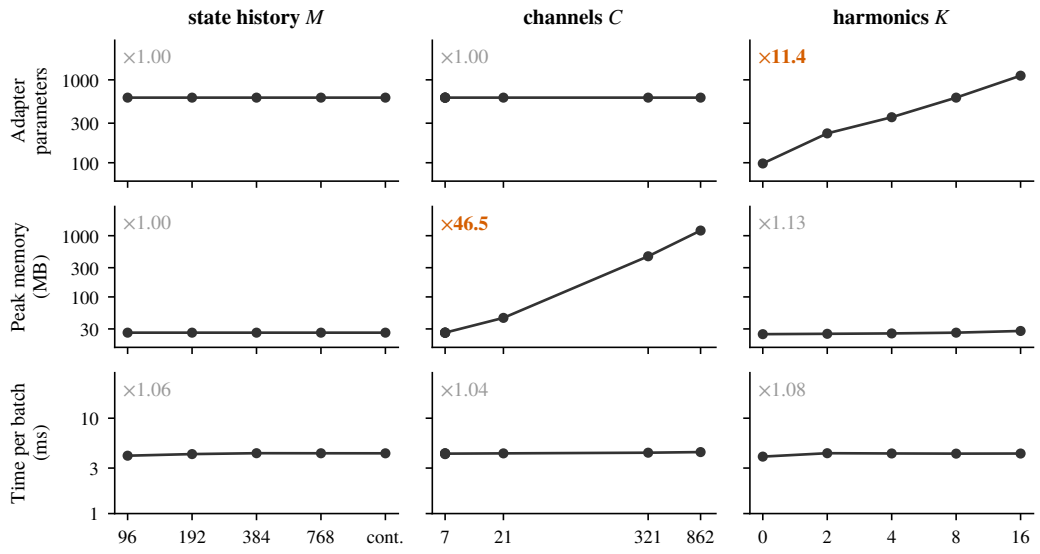

Figure 7: Cost of the adapter along each axis with the other two fixed (DLinear, $L = H = 96$, batch 32, adapter path only). The number in each panel is the ratio between the two ends of the axis, and the vertical axis is logarithmic and shared within a row. Time is the median over five independent processes, each averaging 100 iterations after 20 warm-up iterations on one NVIDIA RTX PRO 6000; its interquartile bars are smaller than the markers.

Table 20: Test MSE and MAE on Exchange, averaged as in the main table. Bold marks the lowest value in each row for each metric.

		MSE						MAE					
Backbone	H	Plain	RevIN	SAN	FAN	DDN	TRACE	Plain	RevIN	SAN	FAN	DDN	TRACE
DLinear	96	0.0885	0.0865	**0.0797**	0.0892	0.0867	0.0807	0.2163	0.2058	0.1987	0.2169	0.2043	**0.1966**
	192	0.1745	0.1782	**0.1610**	0.1940	0.1850	0.1710	0.3119	0.3003	**0.2895**	0.3273	0.3064	0.2924
	336	0.3089	0.3250	**0.3006**	0.4275	0.3314	0.3163	0.4223	0.4119	**0.4005**	0.4954	0.4172	0.4088
MICN	96	0.0886	0.0913	0.0942	0.0923	0.0878	**0.0860**	0.2196	0.2108	0.2181	0.2230	0.2062	**0.2045**
	192	0.1778	0.1929	0.1976	0.1839	**0.1775**	0.1803	0.3222	0.3109	0.3219	0.3204	0.3042	**0.3016**
	336	**0.3097**	0.3484	0.3953	0.4027	0.3379	0.3479	0.4316	0.4256	0.4544	0.4868	0.4266	**0.4255**
TSMixer	96	0.1391	0.0891	**0.0811**	0.0903	0.0848	0.0874	0.2861	0.2105	**0.2010**	0.2193	0.2031	0.2069
	192	0.2340	0.1822	**0.1638**	0.2228	0.1796	0.1863	0.3745	0.3058	**0.2910**	0.3519	0.3015	0.3087
	336	0.3592	0.3350	**0.3061**	0.3574	0.3460	0.3538	0.4670	0.4202	**0.4045**	0.4531	0.4264	0.4325
iTransformer	96	0.3270	0.0877	**0.0805**	0.0937	0.0840	0.0884	0.3999	0.2080	**0.2005**	0.2233	0.2022	0.2089
	192	0.5670	0.1807	**0.1621**	0.1869	0.1753	0.1827	0.5567	0.3044	**0.2917**	0.3233	0.2988	0.3064
	336	0.8882	0.3300	**0.2945**	0.4022	0.3354	0.3356	0.7268	0.4180	**0.3965**	0.4813	0.4205	0.4210

while TRACE follows the protocol of Appendix G. $H = 720$ is omitted because the validation segment does not contain a complete window of $L + H$ observations. In Table 20, TRACE is lower than RevIN in 7 of the 12 cells and lower than the backbone alone in 9, but SAN has the lowest MSE in 9 cells and TRACE in one; on MAE, TRACE is lowest in 4 cells and SAN in 8. The periods selected on this dataset are several hundred observations long and do not correspond to a calendar cycle, so the structural forecast has little stable periodic content to add.

M PER-CELL SETTINGS AND SEARCH BUDGET OF TRACE

Table 21 reports the configuration selected for every backbone–dataset pair by the search of Appendix G. Against these validation-tuned baselines, TRACE at its main-table setting has the lower two-horizon mean MSE in all six pairs against DDN and in four of six against RevIN, the exceptions being ETTh1 with both backbones.

Table 21: Learning rate (LR) and K selected per backbone–dataset pair for the proposed method. Every pair exhaustively evaluates the same 12 configurations, each trained at all four horizons, and the configuration with the lowest mean validation rank across the four horizons is applied unchanged to all four. Appendix G defines the grid and the selection criterion.

	DLinear		MICN		TSMixer		iTransformer	
Dataset	LR	K	LR	K	LR	K	LR	K
ECL	0.005	8	0.001	8	0.005	2	0.001	8
ETTh1	0.005	4	0.005	4	0.005	4	0.001	8
ETTh2	0.001	2	0.005	2	0.005	2	0.0001	4
ETTm1	0.0005	4	0.001	4	0.001	4	0.0005	8
ETTm2	0.0005	2	0.001	2	0.0005	2	0.001	2
Traffic	0.005	8	0.005	8	0.005	8	0.001	8
Weather	0.005	2	0.0005	2	0.001	2	0.0005	2

Table 22: Values of the proposed method in the main table (means over three seeds, settings chosen per pair) and in the factor experiments (one fixed setting shared by the twelve cells). *Difference* is $100\,(\text{factor} - \text{main})/\text{main}$.

Backbone	Data	H	Main table	Factor exp.	Difference (%)
DLinear	ETTh1	720	0.5339	0.4843	-9.28
		96	0.3836	0.3986	$+3.91$
	ETTm2	720	0.3985	0.4031	$+1.15$
		96	0.1771	0.1771	-0.02
	Weather	720	0.2980	0.3562	$+19.53$
		96	0.1649	0.1716	$+4.05$
TSMixer	ETTh1	720	0.5846	0.5281	-9.66
		96	0.3877	0.4296	$+10.82$
	ETTm2	720	0.4090	0.4038	-1.26
		96	0.1755	0.1786	$+1.79$
	Weather	720	0.2965	0.3495	$+17.89$
		96	0.1470	0.1676	$+13.98$

N Setting Contrast Between the Main Table and the Factor Experiments

In the factor experiments of Sec. 4.3, the twelve cells share a single fixed setting. Table 22 places the main-table value and the factor-experiment value of the same cell side by side. The differences range from -9.7% to $+19.5\%$, and since every comparison in the factor experiments is a matched pair within the same cell, these differences do not enter the conclusions of either table.

On the 12 factor cells, we sweep the backbone learning rate over $\{10^{-4}, 3\times10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}\}$ for RevIN and DDN while retaining every method-specific setting from the public implementation. For each backbone–dataset–arm combination, one learning rate is selected by the mean validation MSE over $H \in \{96, 720\}$ and is then applied unchanged to both test horizons. The 120 training runs comprise 2 backbones $\times$ 3 datasets $\times$ 2 baselines $\times$ 2 horizons $\times$ 5 learning rates. Table 23 reports the selected learning rate, resulting two-horizon test MSE, and change from the published learning rate for every combination.

Relative to the published learning rate, validation selection changes the mean test MSE by a median improvement of 0.09% and a mean improvement of 0.22%, with a maximum improvement of 7.37%; it is worse than the published setting in 4 of the 12 combinations. The effect is heterogeneous across methods. For RevIN, the median and mean improvements are 1.44% and 2.32%, and validation selection improves 5/6 combinations. For DDN, they are -0.03% and -1.88%; validation selection improves 2/6, is unchanged in 1/6, and is worse in 3/6. The largest improvement is 7.37% for DDN with TSMixer on Weather, whereas the largest deterioration is -18.94% for DDN with TSMixer on ETTh1. Thus the public-setting comparison is only mildly affected on average in this diagnostic, but tuning does not move every baseline or dataset in the same direction.

Table 23: Validation-selected learning-rate sensitivity on the factor cells. MSE is averaged over $H \in \{96, 720\}$ after selecting one learning rate by the corresponding mean validation MSE. Δ is the test-MSE improvement relative to the published learning rate; positive values are better.

		DDN			RevIN		
Backbone	Data	$\mathrm{lr}^*_{\mathrm{val}}$	MSE	$\Delta(\%)$	$\mathrm{lr}^*_{\mathrm{val}}$	MSE	$\Delta(\%)$
DLinear	ETTh1	10^{-3}	0.459104	-3.56	5×10^{-3}	0.429886	$+2.73$
DLinear	ETTm2	5×10^{-4}	0.291740	-0.06	3×10^{-4}	0.294177	$+0.03$
DLinear	Weather	5×10^{-3}	0.246567	$+3.89$	5×10^{-4}	0.275837	$+0.14$
TSMixer	ETTh1	5×10^{-3}	0.532475	-18.94	10^{-3}	0.444644	$+6.25$
TSMixer	ETTm2	10^{-4}	0.297954	0.00	5×10^{-4}	0.299588	-1.29
TSMixer	Weather	10^{-3}	0.237769	$+7.37$	5×10^{-3}	0.253744	$+6.09$

Table 24: Comparison of TRACE with each adapter over the 112 cells. Intervals are 95% percentile intervals from 10,000 resamples of the 28 backbone–dataset pairs.

Metric	vs.	Cells won	Pairs won	Ratio red. (%)	Mean red. (%)
MSE	RevIN	105/112	27/28	10.66 [6.84, 14.92]	9.69 [6.80, 12.88]
	SAN	101/112	26/28	16.11 [5.97, 28.97]	9.73 [5.63, 15.12]
	FAN	108/112	28/28	13.78 [10.86, 16.77]	11.63 [9.47, 13.91]
	DDN	97/112	25/28	6.10 [4.00, 8.20]	5.82 [4.09, 7.56]
MAE	RevIN	97/112	26/28	4.67 [2.38, 7.56]	4.57 [2.49, 7.07]
	SAN	93/112	24/28	9.30 [3.49, 17.91]	7.22 [3.73, 12.21]
	FAN	110/112	28/28	9.08 [7.18, 11.07]	8.44 [6.86, 10.12]
	DDN	79/112	21/28	2.15 [0.75, 3.56]	2.29 [0.76, 3.77]

O FULL RESULTS

Each backbone alone and with TRACE attached is reported in Table 1 (Sec. 4.2). Tables 25–28 present the MSE and MAE of all 112 cells used to compute the means and win counts in the body. The only cell in which TRACE has a higher mean MSE than the backbone alone is DLinear on ETTh1 at $H = 720$, and the cell in which the MSE of TRACE varies most across seeds is TSMixer on ETTh1 at $H = 720$, with a coefficient of variation of 6.5% against a median of 0.67% over the 112 cells. Table 24 summarizes the comparison over cells. Intervals resample the 28 backbone–dataset pairs together with their four horizons, so cells that share a dataset and a backbone move together. On MSE, TRACE has the lower horizon-averaged error in 25–28 of the 28 pairs against every adapter, and no interval of the reductions contains zero.

We separately reran the 12 conspicuously large baseline cells with the public setting and the first seed. The reruns reproduce the reported scale and structured pattern: SAN with iTransformer on Traffic increases smoothly from 1.356 at $H = 96$ to 1.405 at $H = 720$; FAN on ETTh2 at $H = 720$ remains high across DLinear, TSMixer, MICN and iTransformer (0.728, 0.732, 0.881 and 0.682); and the remaining checks reproduce SAN with TSMixer on ETTm2 at $H \in \{192, 336\}$, DDN with MICN on ETTh1 at $H = 96$, and RevIN with MICN on ETTh1 at $H = 192$. We therefore retain these cells rather than treating them as isolated divergence. These single-seed reruns are diagnostic checks, not uncertainty estimates.

P PER-CELL CURVES FOR THE HISTORY LENGTH

Figure 8 plots the values of Table 19 from Sec. 4.3 as per-cell curves. The inference-time intervention starts from a diffuse prior at every origin and processes only the most recent M observations, so it contains both the removal of old history and the initial convergence of the filter, whereas the retrained series is subject to the same restriction from training onwards. The two runs in which the cap a remained exactly zero after training are both retraining conditions with M=96: DLinear/ETTh1 H=96 and DLinear/Weather H=96.

History length and the number of harmonics. The reversal on Weather at $H = 720$ is tied to the long-period components of the factor filter. We retrain the factor configuration on Weather with

Table 25: Full test MSE and MAE results for DLinear. Values are mean ± sample standard deviation over three seeds. The lowest full-precision mean in each row is bold and the second lowest is underlined.

	Dataset	H	Plain	RevIN	SAN	FAN	DDN	TRACE
MSE	ECL	96	0.2104±0.0001	0.2124±0.0000	0.1873±0.0002	0.1907±0.0012	0.1766±0.0003	**0.1554±0.0001**
		192	0.2102±0.0000	0.2128±0.0000	0.1889±0.0003	0.1943±0.0005	0.1820±0.0000	**0.1659±0.0001**
		336	0.2231±0.0001	0.2273±0.0000	0.2022±0.0007	0.2094±0.0005	0.1965±0.0001	**0.1812±0.0001**
		720	0.2578±0.0000	0.2683±0.0000	0.2377±0.0001	0.2459±0.0003	0.2350±0.0004	**0.2196±0.0005**
	ETTh1	96	0.3964±0.0011	0.3988±0.0007	0.3844±0.0025	0.3994±0.0030	**0.3821±0.0005**	0.3836±0.0029
		192	0.4452±0.0003	0.4490±0.0007	0.4316±0.0026	0.4552±0.0031	0.4327±0.0003	**0.4178±0.0016**
		336	0.4872±0.0003	0.4891±0.0004	0.4714±0.0073	0.5046±0.0017	0.4776±0.0007	**0.4581±0.0034**
		720	0.5137±0.0023	0.4852±0.0003	**0.4704±0.0022**	0.5647±0.0079	0.5036±0.0005	0.5339±0.0083
	ETTh2	96	0.3452±0.0052	0.2942±0.0002	0.2892±0.0016	0.3234±0.0081	0.2934±0.0002	**0.2824±0.0011**
		192	0.4784±0.0050	0.3793±0.0002	0.3709±0.0003	0.4265±0.0062	0.3799±0.0002	**0.3648±0.0004**
		336	0.5921±0.0108	0.4175±0.0000	0.4116±0.0016	0.4876±0.0080	0.4192±0.0003	**0.3976±0.0031**
		720	0.8279±0.0145	0.4207±0.0001	0.4161±0.0009	0.7195±0.0230	0.4240±0.0003	**0.3983±0.0008**
	ETTm1	96	0.3455±0.0004	0.3505±0.0001	0.3279±0.0027	0.3379±0.0011	0.3285±0.0040	**0.3242±0.0003**
		192	0.3816±0.0003	0.3871±0.0002	0.3634±0.0010	0.3843±0.0024	0.3651±0.0009	**0.3411±0.0030**
		336	0.4142±0.0010	0.4193±0.0001	0.3918±0.0013	0.4212±0.0039	0.3969±0.0015	**0.3608±0.0003**
		720	0.4727±0.0011	0.4794±0.0003	0.4506±0.0004	0.4873±0.0008	0.4582±0.0007	**0.4145±0.0019**
	ETTm2	96	0.1932±0.0021	0.1817±0.0000	0.1792±0.0014	0.1862±0.0039	0.1798±0.0011	**0.1771±0.0006**
		192	0.2809±0.0047	0.2460±0.0001	0.2435±0.0023	0.2611±0.0090	0.2463±0.0009	**0.2336±0.0001**
		336	0.3815±0.0055	0.3066±0.0002	0.3014±0.0009	0.3290±0.0097	0.3091±0.0028	**0.2908±0.0001**
		720	0.5334±0.0206	0.4068±0.0001	**0.3961±0.0044**	0.5096±0.0319	0.4049±0.0013	0.3985±0.0003
	Traffic	96	0.6965±0.0001	0.6905±0.0002	0.6207±0.0019	0.5936±0.0021	**0.4996±0.0022**	0.5160±0.0035
		192	0.6465±0.0001	0.6442±0.0000	0.5959±0.0011	0.5795±0.0016	0.5050±0.0021	**0.5002±0.0025**
		336	0.6532±0.0001	0.6515±0.0002	0.6027±0.0053	0.5925±0.0002	0.5190±0.0003	**0.5087±0.0017**
		720	0.6941±0.0002	0.6896±0.0002	0.6396±0.0020	0.6407±0.0013	0.5622±0.0018	**0.5437±0.0112**
	Weather	96	0.1961±0.0004	0.1931±0.0000	0.1802±0.0001	0.1776±0.0005	0.1704±0.0007	**0.1649±0.0012**
		192	0.2369±0.0018	0.2369±0.0002	0.2208±0.0004	0.2190±0.0007	0.2136±0.0005	**0.2119±0.0030**
		336	0.2816±0.0005	0.2879±0.0002	0.2721±0.0002	0.2684±0.0011	0.2655±0.0007	**0.2425±0.0005**
		720	0.3458±0.0007	0.3596±0.0003	0.3411±0.0005	0.3408±0.0029	0.3438±0.0011	**0.2980±0.0003**
MAE	ECL	96	0.3015±0.0001	0.2974±0.0001	0.2721±0.0004	0.2784±0.0015	0.2662±0.0004	**0.2499±0.0001**
		192	0.3048±0.0000	0.3005±0.0000	0.2762±0.0003	0.2843±0.0004	0.2732±0.0001	**0.2587±0.0000**
		336	0.3192±0.0001	0.3145±0.0000	0.2906±0.0002	0.3013±0.0003	0.2899±0.0000	**0.2744±0.0000**
		720	0.3497±0.0000	0.3450±0.0001	0.3223±0.0001	0.3343±0.0001	0.3251±0.0002	**0.3071±0.0007**
	ETTh1	96	0.4111±0.0010	0.4092±0.0005	**0.3951±0.0018**	0.4126±0.0026	0.3959±0.0004	0.4033±0.0017
		192	0.4406±0.0002	0.4369±0.0007	**0.4246±0.0020**	0.4469±0.0039	0.4257±0.0001	0.4268±0.0013
		336	0.4653±0.0003	0.4563±0.0004	**0.4456±0.0041**	0.4774±0.0035	0.4502±0.0004	0.4504±0.0020
		720	0.5113±0.0019	0.4758±0.0002	**0.4696±0.0013**	0.5428±0.0063	0.4916±0.0003	0.4956±0.0010
	ETTh2	96	0.3983±0.0041	0.3419±0.0002	0.3422±0.0023	0.3770±0.0058	0.3421±0.0003	**0.3336±0.0007**
		192	0.4771±0.0031	0.3932±0.0001	0.3909±0.0008	0.4397±0.0043	0.3951±0.0008	**0.3840±0.0001**
		336	0.5412±0.0050	0.4268±0.0000	0.4273±0.0003	0.4788±0.0052	0.4290±0.0001	**0.4167±0.0019**
		720	0.6561±0.0058	0.4390±0.0000	0.4416±0.0009	0.6104±0.0062	0.4446±0.0002	**0.4308±0.0001**
	ETTm1	96	0.3725±0.0010	0.3684±0.0002	**0.3595±0.0010**	0.3722±0.0005	0.3693±0.0040	0.3642±0.0001
		192	0.3905±0.0005	0.3858±0.0002	0.3810±0.0002	0.3992±0.0034	0.3886±0.0002	**0.3773±0.0014**
		336	0.4140±0.0010	0.4065±0.0002	0.4001±0.0006	0.4256±0.0051	0.4117±0.0019	**0.3961±0.0000**
		720	0.4503±0.0013	0.4401±0.0003	0.4392±0.0004	0.4667±0.0007	0.4480±0.0014	**0.4298±0.0014**
	ETTm2	96	0.2923±0.0028	0.2648±0.0001	0.2697±0.0046	0.2766±0.0049	0.2683±0.0022	**0.2610±0.0004**
		192	0.3577±0.0047	0.3044±0.0001	0.3127±0.0056	0.3314±0.0061	0.3105±0.0021	**0.2995±0.0001**
		336	0.4264±0.0039	0.3423±0.0002	0.3503±0.0051	0.3727±0.0061	0.3502±0.0025	**0.3378±0.0002**
		720	0.5111±0.0112	0.3979±0.0000	0.4077±0.0101	0.4899±0.0137	0.4031±0.0001	**0.3958±0.0004**
	Traffic	96	0.4287±0.0001	0.4161±0.0001	0.3869±0.0004	0.3772±0.0004	**0.3138±0.0025**	0.3463±0.0016
		192	0.4073±0.0001	0.3979±0.0000	0.3771±0.0004	0.3729±0.0005	**0.3144±0.0010**	0.3389±0.0019
		336	0.4098±0.0001	0.4005±0.0002	0.3762±0.0089	0.3775±0.0002	**0.3225±0.0002**	0.3402±0.0005
		720	0.4287±0.0001	0.4172±0.0002	0.3871±0.0004	0.3993±0.0003	**0.3415±0.0001**	0.3552±0.0005
	Weather	96	0.2569±0.0014	0.2332±0.0000	0.2405±0.0001	0.2346±0.0009	0.2321±0.0011	**0.2255±0.0017**
		192	0.2955±0.0033	0.2685±0.0001	0.2793±0.0003	0.2728±0.0007	0.2740±0.0011	**0.2624±0.0026**
		336	0.3313±0.0006	0.3040±0.0001	0.3211±0.0005	0.3131±0.0021	0.3125±0.0007	**0.2889±0.0003**
		720	0.3823±0.0012	0.3499±0.0002	0.3714±0.0012	0.3664±0.0052	0.3653±0.0017	**0.3307±0.0003**

Table 26: Full test MSE and MAE results for MICN. Values are mean ± sample standard deviation over three seeds. The lowest full-precision mean in each row is bold and the second lowest is underlined.

	Dataset	H	Plain	RevIN	SAN	FAN	DDN	TRACE
MSE	ECL	96	0.1600±0.0007	0.1665±0.0015	0.1572±0.0004	0.1522±0.0005	**0.1454±0.0010**	0.1463±0.0011
		192	0.1735±0.0007	0.1775±0.0017	0.1653±0.0027	0.1616±0.0004	**0.1607±0.0009**	0.1637±0.0064
		336	0.1861±0.0003	0.1835±0.0014	0.1746±0.0007	0.1732±0.0006	0.1700±0.0006	**0.1670±0.0022**
		720	0.2143±0.0057	0.2059±0.0034	0.2012±0.0016	0.1944±0.0008	**0.1894±0.0012**	0.1894±0.0003
	ETTh1	96	0.4325±0.0193	0.3994±0.0015	**0.3932±0.0104**	0.4047±0.0059	0.5251±0.0276	0.3990±0.0126
		192	0.4853±0.0361	0.4918±0.0675	0.4586±0.0188	0.4580±0.0064	0.5187±0.0144	**0.4295±0.0077**
		336	0.5708±0.0442	0.5050±0.0100	0.4747±0.0082	0.5121±0.0048	0.5861±0.0205	**0.4541±0.0092**
		720	0.7699±0.0575	0.5337±0.0344	**0.4823±0.0045**	0.5813±0.0373	0.4904±0.0123	0.5043±0.0098
	ETTh2	96	0.3690±0.0313	0.3071±0.0072	0.2905±0.0026	0.3134±0.0101	0.2881±0.0007	**0.2872±0.0008**
		192	0.5039±0.0228	0.3929±0.0136	**0.3710±0.0015**	0.4106±0.0148	0.3746±0.0021	0.3735±0.0153
		336	0.6012±0.0252	0.4292±0.0047	0.4149±0.0009	0.5142±0.0442	0.4172±0.0006	**0.4085±0.0080**
		720	0.8025±0.0173	0.4370±0.0091	0.4378±0.0086	0.8145±0.0577	0.4427±0.0068	**0.3987±0.0084**
	ETTm1	96	0.3191±0.0042	0.3367±0.0122	**0.3159±0.0053**	0.3185±0.0033	0.3280±0.0011	0.3166±0.0040
		192	0.3657±0.0004	0.3913±0.0044	0.3631±0.0060	0.3638±0.0032	0.3720±0.0089	**0.3391±0.0037**
		336	0.4133±0.0063	0.3997±0.0080	0.4029±0.0066	0.4019±0.0010	0.4036±0.0008	**0.3634±0.0063**
		720	0.5115±0.0110	0.4550±0.0033	0.4844±0.0082	0.4683±0.0032	0.4734±0.0079	**0.4117±0.0030**
	ETTm2	96	0.1834±0.0112	0.1797±0.0017	**0.1727±0.0008**	0.1777±0.0038	0.1802±0.0025	0.1752±0.0005
		192	0.2812±0.0114	0.2448±0.0019	0.2378±0.0020	0.2506±0.0105	0.2407±0.0029	**0.2316±0.0001**
		336	0.3795±0.0130	0.3019±0.0029	0.2999±0.0011	0.3229±0.0129	0.3007±0.0022	**0.2912±0.0022**
		720	0.5545±0.0220	0.3899±0.0059	0.3944±0.0018	0.4842±0.0290	0.4064±0.0034	**0.3876±0.0105**
	Traffic	96	0.5181±0.0038	0.4972±0.0025	0.5390±0.0186	0.5168±0.0015	0.4678±0.0046	**0.4119±0.0018**
		192	0.5355±0.0024	0.5147±0.0025	0.5273±0.0071	0.5071±0.0009	0.4865±0.0015	**0.4320±0.0038**
		336	0.5472±0.0015	0.5360±0.0005	0.5500±0.0514	0.5173±0.0012	0.5007±0.0011	**0.4446±0.0013**
		720	0.5716±0.0032	0.5697±0.0043	0.5828±0.0150	0.5471±0.0007	0.5337±0.0046	**0.4797±0.0008**
	Weather	96	0.1865±0.0015	0.1677±0.0005	0.1757±0.0055	**0.1573±0.0002**	0.1662±0.0071	0.1609±0.0007
		192	0.2322±0.0016	0.2194±0.0011	0.2289±0.0013	**0.2045±0.0016**	0.2177±0.0034	0.2085±0.0013
		336	0.2780±0.0088	0.2747±0.0021	0.2946±0.0060	0.2538±0.0020	0.2756±0.0104	**0.2481±0.0039**
		720	0.3293±0.0164	0.3485±0.0016	0.3686±0.0003	0.3193±0.0043	0.3581±0.0046	**0.3012±0.0006**
MAE	ECL	96	0.2703±0.0010	0.2717±0.0014	0.2606±0.0006	0.2532±0.0005	0.2502±0.0010	**0.2491±0.0013**
		192	0.2850±0.0010	0.2808±0.0017	0.2681±0.0030	**0.2635±0.0004**	0.2642±0.0008	0.2642±0.0061
		336	0.2972±0.0003	0.2873±0.0016	0.2796±0.0011	0.2776±0.0008	0.2757±0.0005	**0.2678±0.0022**
		720	0.3212±0.0038	0.3048±0.0025	0.3031±0.0014	0.2983±0.0005	0.2960±0.0007	**0.2866±0.0008**
	ETTh1	96	0.4414±0.0082	0.4138±0.0027	**0.4070±0.0049**	0.4212±0.0035	0.4384±0.0079	0.4140±0.0030
		192	0.4769±0.0195	0.4573±0.0240	0.4451±0.0091	0.4522±0.0050	0.4473±0.0038	**0.4407±0.0073**
		336	0.5385±0.0246	0.4676±0.0023	**0.4537±0.0052**	0.4836±0.0020	0.4804±0.0059	0.4560±0.0023
		720	0.6574±0.0240	0.4932±0.0110	0.4799±0.0071	0.5539±0.0155	**0.4628±0.0049**	0.4828±0.0026
	ETTh2	96	0.4117±0.0222	0.3506±0.0043	0.3430±0.0013	0.3686±0.0072	**0.3382±0.0012**	0.3410±0.0017
		192	0.4885±0.0126	0.4018±0.0052	0.3936±0.0010	0.4301±0.0100	**0.3919±0.0015**	0.3944±0.0072
		336	0.5421±0.0117	0.4373±0.0032	0.4291±0.0010	0.4924±0.0252	0.4294±0.0009	**0.4266±0.0032**
		720	0.6451±0.0062	0.4494±0.0057	0.4558±0.0065	0.6531±0.0248	0.4593±0.0053	**0.4359±0.0038**
	ETTm1	96	0.3707±0.0030	0.3700±0.0062	**0.3590±0.0030**	0.3627±0.0027	0.3674±0.0015	0.3611±0.0021
		192	0.4029±0.0011	0.3977±0.0025	0.3898±0.0035	0.3882±0.0022	0.3912±0.0060	**0.3827±0.0021**
		336	0.4386±0.0046	0.4084±0.0057	0.4140±0.0042	0.4160±0.0008	0.4142±0.0016	**0.3989±0.0025**
		720	0.5026±0.0080	0.4367±0.0027	0.4584±0.0047	0.4570±0.0011	0.4508±0.0041	**0.4258±0.0020**
	ETTm2	96	0.2810±0.0123	0.2624±0.0011	0.2626±0.0035	0.2695±0.0038	0.2634±0.0030	**0.2591±0.0006**
		192	0.3548±0.0107	0.3040±0.0014	0.3067±0.0053	0.3212±0.0106	0.3012±0.0028	**0.2989±0.0001**
		336	0.4196±0.0112	0.3409±0.0004	0.3485±0.0047	0.3694±0.0107	0.3400±0.0028	**0.3398±0.0005**
		720	0.5164±0.0126	0.3956±0.0050	0.4008±0.0037	0.4707±0.0177	0.4038±0.0025	**0.3950±0.0029**
	Traffic	96	0.3108±0.0021	0.3077±0.0014	0.3409±0.0170	0.3309±0.0021	**0.2718±0.0010**	0.2877±0.0023
		192	0.3163±0.0019	0.3140±0.0004	0.3515±0.0072	0.3220±0.0002	**0.2797±0.0008**	0.2944±0.0024
		336	0.3206±0.0016	0.3236±0.0003	0.3567±0.0220	0.3262±0.0017	**0.2843±0.0003**	0.2979±0.0024
		720	0.3306±0.0027	0.3379±0.0021	0.3599±0.0136	0.3380±0.0002	**0.2987±0.0013**	0.3179±0.0022
	Weather	96	0.2629±0.0016	**0.2137±0.0004**	0.2354±0.0041	0.2172±0.0005	0.2307±0.0096	0.2145±0.0001
		192	0.3028±0.0026	0.2576±0.0004	0.2845±0.0008	0.2647±0.0009	0.2786±0.0018	**0.2549±0.0005**
		336	0.3419±0.0087	0.2972±0.0012	0.3354±0.0053	0.3076±0.0025	0.3216±0.0102	**0.2885±0.0044**
		720	0.3767±0.0159	0.3460±0.0011	0.3818±0.0001	0.3558±0.0065	0.3818±0.0018	**0.3300±0.0006**

Table 27: Full test MSE and MAE results for TSMixer. Values are mean ± sample standard deviation over three seeds. The lowest full-precision mean in each row is bold and the second lowest is underlined.

	Dataset	H	Plain	RevIN	SAN	FAN	DDN	TRACE
MSE	ECL	96	0.2247±0.0006	0.2162±0.0019	0.1702±0.0002	0.1767±0.0002	0.1766±0.0003	**0.1519±0.0002**
		192	0.2310±0.0002	0.2233±0.0002	0.1799±0.0014	0.1832±0.0004	0.1859±0.0005	**0.1673±0.0014**
		336	0.2457±0.0015	0.2400±0.0003	0.1980±0.0022	0.1966±0.0002	0.2004±0.0003	**0.1818±0.0046**
		720	0.2801±0.0038	0.2817±0.0007	0.2357±0.0042	0.2296±0.0005	0.2381±0.0006	**0.2126±0.0085**
	ETTh1	96	0.5034±0.0139	0.4301±0.0022	0.3881±0.0002	0.4037±0.0043	0.3917±0.0007	**0.3877±0.0041**
		192	0.5685±0.0165	0.4821±0.0020	0.4601±0.0447	0.4606±0.0043	**0.4418±0.0002**	0.4555±0.0293
		336	0.6221±0.0209	0.5213±0.0042	0.4977±0.0169	0.5123±0.0049	0.4860±0.0009	**0.4744±0.0090**
		720	0.6605±0.0229	0.5254±0.0047	0.5250±0.0338	0.5559±0.0056	**0.5071±0.0040**	0.5846±0.0379
	ETTh2	96	0.5718±0.0198	0.3153±0.0037	0.3151±0.0136	0.3289±0.0088	0.2999±0.0059	**0.2978±0.0028**
		192	0.7726±0.0054	0.4000±0.0037	0.4132±0.0147	0.4216±0.0102	0.3867±0.0024	**0.3841±0.0172**
		336	0.9087±0.0267	0.4408±0.0023	0.4352±0.0099	0.4980±0.0117	**0.4270±0.0038**	0.4307±0.0220
		720	1.0663±0.0633	0.4507±0.0074	0.5455±0.0585	0.7278±0.0114	0.4373±0.0014	**0.4141±0.0046**
	ETTm1	96	0.4100±0.0314	0.3398±0.0012	0.3234±0.0011	0.3378±0.0012	0.3411±0.0007	**0.3078±0.0008**
		192	0.4404±0.0162	0.3789±0.0006	0.3605±0.0010	0.3802±0.0004	0.3722±0.0008	**0.3334±0.0014**
		336	0.4850±0.0200	0.4108±0.0013	0.3891±0.0011	0.4185±0.0019	0.4015±0.0017	**0.3637±0.0034**
		720	0.5719±0.0205	0.4775±0.0008	0.4473±0.0008	0.4896±0.0083	0.4544±0.0054	**0.4205±0.0120**
	ETTm2	96	0.2824±0.0186	0.1858±0.0008	0.1776±0.0017	0.1890±0.0028	0.1815±0.0003	**0.1755±0.0002**
		192	0.4097±0.0563	0.2507±0.0006	0.3595±0.0959	0.2562±0.0092	0.2510±0.0038	**0.2335±0.0002**
		336	0.5526±0.0375	0.3109±0.0004	0.4457±0.1513	0.3579±0.0112	0.3132±0.0008	**0.2907±0.0015**
		720	0.9119±0.0273	**0.4071±0.0007**	0.4567±0.0160	0.5132±0.0431	0.4154±0.0033	0.4090±0.0070
	Traffic	96	0.7611±0.0052	0.7220±0.0055	0.6144±0.0239	0.5676±0.0032	0.5047±0.0016	**0.4525±0.0004**
		192	0.7307±0.0029	0.7000±0.0049	0.5831±0.0220	0.5545±0.0045	0.5097±0.0007	**0.4637±0.0026**
		336	0.7500±0.0078	0.7137±0.0033	0.6145±0.0248	0.5689±0.0035	0.5306±0.0007	**0.4814±0.0070**
		720	0.8009±0.0127	0.7566±0.0030	0.6541±0.0077	0.6145±0.0012	0.5735±0.0028	**0.5113±0.0024**
	Weather	96	0.1805±0.0014	0.1841±0.0008	0.1764±0.0025	0.1710±0.0029	0.1684±0.0004	**0.1470±0.0004**
		192	0.2176±0.0014	0.2295±0.0007	0.2180±0.0020	0.2176±0.0004	0.2120±0.0011	**0.1946±0.0009**
		336	0.2613±0.0007	0.2815±0.0020	0.2688±0.0012	0.2619±0.0012	0.2657±0.0008	**0.2364±0.0018**
		720	0.3175±0.0007	0.3550±0.0004	0.3407±0.0011	0.3279±0.0034	0.3439±0.0030	**0.2965±0.0007**
MAE	ECL	96	0.3228±0.0007	0.3097±0.0016	0.2705±0.0011	0.2703±0.0001	0.2725±0.0004	**0.2548±0.0011**
		192	0.3328±0.0006	0.3159±0.0006	0.2775±0.0007	0.2779±0.0004	0.2809±0.0005	**0.2668±0.0011**
		336	0.3495±0.0021	0.3310±0.0004	0.2998±0.0035	0.2936±0.0002	0.2975±0.0001	**0.2841±0.0044**
		720	0.3776±0.0044	0.3616±0.0000	0.3335±0.0053	0.3248±0.0001	0.3317±0.0003	**0.3084±0.0083**
	ETTh1	96	0.4925±0.0109	0.4340±0.0014	**0.4017±0.0009**	0.4162±0.0034	0.4032±0.0006	0.4080±0.0034
		192	0.5298±0.0124	0.4614±0.0011	0.4433±0.0229	0.4516±0.0036	**0.4323±0.0002**	0.4546±0.0192
		336	0.5610±0.0148	0.4792±0.0027	0.4614±0.0082	0.4827±0.0051	**0.4563±0.0010**	0.4603±0.0062
		720	0.6040±0.0136	0.4989±0.0028	0.5037±0.0270	0.5348±0.0046	**0.4951±0.0024**	0.5270±0.0193
	ETTh2	96	0.5467±0.0138	0.3612±0.0028	0.3659±0.0073	0.3819±0.0073	0.3480±0.0045	**0.3449±0.0019**
		192	0.6386±0.0228	0.4111±0.0028	0.4266±0.0058	0.4378±0.0066	0.4007±0.0007	**0.3987±0.0097**
		336	0.6993±0.0344	0.4458±0.0020	0.4472±0.0038	0.4853±0.0065	0.4348±0.0022	**0.4347±0.0105**
		720	0.7668±0.0493	0.4594±0.0043	0.5400±0.0381	0.6115±0.0040	0.4526±0.0007	**0.4432±0.0011**
	ETTm1	96	0.4349±0.0206	0.3733±0.0017	**0.3580±0.0015**	0.3737±0.0019	0.3802±0.0008	0.3583±0.0022
		192	0.4496±0.0113	0.3911±0.0011	**0.3795±0.0019**	0.3952±0.0005	0.3922±0.0009	0.3797±0.0013
		336	0.4804±0.0147	0.4110±0.0022	0.4003±0.0013	0.4239±0.0026	0.4142±0.0022	**0.3983±0.0012**
		720	0.5324±0.0156	0.4458±0.0030	0.4365±0.0006	0.4700±0.0102	0.4476±0.0006	**0.4298±0.0064**
	ETTm2	96	0.3804±0.0020	0.2694±0.0009	0.2645±0.0018	0.2806±0.0035	0.2685±0.0009	**0.2586±0.0005**
		192	0.4570±0.0166	0.3103±0.0006	0.3904±0.0612	0.3260±0.0081	0.3151±0.0057	**0.2999±0.0004**
		336	0.5394±0.0046	0.3480±0.0003	0.4374±0.0789	0.3937±0.0071	0.3510±0.0010	**0.3391±0.0010**
		720	0.7127±0.0224	**0.4022±0.0006**	0.4528±0.0117	0.4956±0.0252	0.4181±0.0017	0.4040±0.0027
	Traffic	96	0.4839±0.0024	0.4680±0.0013	0.3885±0.0124	0.3801±0.0044	**0.3117±0.0009**	0.3257±0.0029
		192	0.4777±0.0038	0.4563±0.0021	0.3824±0.0125	0.3755±0.0052	**0.3180±0.0004**	0.3283±0.0023
		336	0.4876±0.0068	0.4606±0.0014	0.3937±0.0111	0.3797±0.0035	**0.3246±0.0002**	0.3371±0.0073
		720	0.5082±0.0086	0.4778±0.0016	0.4030±0.0064	0.4015±0.0006	**0.3433±0.0011**	0.3514±0.0015
	Weather	96	0.2500±0.0016	0.2290±0.0003	0.2369±0.0021	0.2310±0.0029	0.2327±0.0010	**0.2027±0.0003**
		192	0.2850±0.0021	0.2658±0.0004	0.2761±0.0003	0.2762±0.0010	0.2741±0.0014	**0.2473±0.0005**
		336	0.3187±0.0017	0.3018±0.0014	0.3151±0.0009	0.3113±0.0016	0.3142±0.0013	**0.2833±0.0001**
		720	0.3598±0.0016	0.3490±0.0005	0.3672±0.0026	0.3609±0.0037	0.3692±0.0034	**0.3274±0.0008**

Table 28: Full test MSE and MAE results for iTransformer. Values are mean ± sample standard deviation over three seeds. The lowest full-precision mean in each row is bold and the second lowest is underlined.

	Dataset	H	Plain	RevIN	SAN	FAN	DDN	TRACE
MSE	ECL	96	0.1498±0.0006	0.1395±0.0001	0.1485±0.0001	0.1655±0.0003	0.1440±0.0002	**0.1370±0.0006**
		192	0.1648±0.0015	0.1580±0.0002	0.1631±0.0016	0.1742±0.0003	0.1648±0.0013	**0.1577±0.0011**
		336	0.1937±0.0026	0.1740±0.0017	0.1744±0.0011	0.1886±0.0007	0.1829±0.0012	**0.1687±0.0030**
		720	0.2137±0.0012	0.2032±0.0017	**0.2020±0.0016**	0.2164±0.0001	0.2211±0.0015	0.2129±0.0058
	ETTh1	96	0.4032±0.0035	0.3864±0.0011	0.4643±0.0333	0.3983±0.0009	**0.3843±0.0004**	0.3864±0.0069
		192	0.4698±0.0029	0.4438±0.0027	0.5132±0.0186	0.4556±0.0020	0.4391±0.0004	**0.4377±0.0061**
		336	0.5276±0.0118	**0.4851±0.0033**	0.5172±0.0059	0.5043±0.0028	0.4941±0.0039	0.4866±0.0137
		720	0.5748±0.0243	**0.5048±0.0108**	0.5386±0.0121	0.5648±0.0082	0.5678±0.0051	0.5062±0.0225
	ETTh2	96	0.5210±0.0715	0.2994±0.0004	0.3104±0.0057	0.3212±0.0039	**0.2968±0.0006**	0.2989±0.0014
		192	0.7383±0.0801	0.3797±0.0007	0.3920±0.0060	0.4252±0.0105	0.3817±0.0009	**0.3788±0.0013**
		336	0.9299±0.0396	0.4230±0.0006	0.4307±0.0018	0.5075±0.0107	**0.4189±0.0062**	0.4214±0.0017
		720	1.1701±0.0566	0.4294±0.0018	0.4390±0.0030	0.6997±0.0219	**0.4227±0.0060**	0.4253±0.0027
	ETTm1	96	0.3809±0.0075	0.3391±0.0015	0.3263±0.0021	0.3362±0.0019	0.3327±0.0020	**0.3185±0.0011**
		192	0.4180±0.0077	0.3805±0.0013	0.3614±0.0013	0.3811±0.0032	0.3703±0.0025	**0.3551±0.0055**
		336	0.4640±0.0077	0.4179±0.0020	0.3924±0.0011	0.4194±0.0037	0.4043±0.0010	**0.3804±0.0047**
		720	0.5490±0.0054	0.4878±0.0024	0.4524±0.0010	0.4901±0.0131	0.4643±0.0033	**0.4351±0.0028**
	ETTm2	96	0.2322±0.0157	0.1837±0.0010	0.1815±0.0013	0.1864±0.0062	**0.1790±0.0010**	0.1825±0.0032
		192	0.3678±0.0756	0.2525±0.0008	0.2455±0.0019	0.2571±0.0012	0.2503±0.0021	**0.2381±0.0016**
		336	0.6035±0.0433	0.3148±0.0025	0.3042±0.0015	0.3392±0.0161	0.3103±0.0039	**0.3008±0.0010**
		720	1.0768±0.0611	0.4134±0.0019	**0.3982±0.0009**	0.5033±0.0034	0.4083±0.0025	0.4032±0.0052
	Traffic	96	0.5299±0.0027	0.4120±0.0014	1.3556±0.0003	0.5192±0.0026	0.4832±0.0035	**0.3809±0.0006**
		192	0.5534±0.0074	0.4355±0.0021	1.3677±0.0002	0.5204±0.0027	0.4451±0.0009	**0.4018±0.0010**
		336	0.5799±0.0113	0.4506±0.0009	1.3839±0.0006	0.5411±0.0025	0.4634±0.0010	**0.4127±0.0019**
		720	0.6076±0.0060	0.4888±0.0020	1.4051±0.0004	0.5985±0.0028	0.5281±0.0220	**0.4422±0.0034**
	Weather	96	0.1696±0.0019	0.1697±0.0010	0.1724±0.0024	0.1756±0.0019	0.1697±0.0013	**0.1582±0.0010**
		192	0.2168±0.0023	0.2164±0.0016	0.2136±0.0025	0.2208±0.0005	0.2120±0.0034	**0.2082±0.0078**
		336	0.2757±0.0020	0.2743±0.0012	0.2664±0.0012	0.2739±0.0029	0.2680±0.0013	**0.2416±0.0012**
		720	0.3590±0.0154	0.3541±0.0015	0.3346±0.0010	0.3432±0.0066	0.3485±0.0047	**0.2978±0.0004**
MAE	ECL	96	0.2478±0.0011	**0.2350±0.0002**	0.2461±0.0002	0.2592±0.0001	0.2434±0.0004	0.2351±0.0010
		192	0.2629±0.0023	**0.2520±0.0001**	0.2621±0.0022	0.2688±0.0006	0.2645±0.0012	0.2546±0.0013
		336	0.2948±0.0031	0.2690±0.0016	0.2766±0.0008	0.2857±0.0008	0.2855±0.0013	**0.2670±0.0032**
		720	0.3155±0.0008	**0.2954±0.0018**	0.3030±0.0024	0.3124±0.0001	0.3195±0.0005	0.3041±0.0032
	ETTh1	96	0.4247±0.0039	0.4042±0.0006	0.4530±0.0158	0.4118±0.0010	**0.3957±0.0001**	0.4112±0.0043
		192	0.4691±0.0021	0.4369±0.0012	0.4777±0.0086	0.4475±0.0010	**0.4292±0.0005**	0.4440±0.0035
		336	0.5054±0.0071	**0.4601±0.0027**	0.4785±0.0032	0.4759±0.0027	0.4612±0.0027	0.4709±0.0066
		720	0.5542±0.0158	0.4916±0.0062	0.5161±0.0075	0.5415±0.0054	0.5222±0.0019	**0.4908±0.0107**
	ETTh2	96	0.5165±0.0456	0.3494±0.0006	0.3635±0.0056	0.3741±0.0032	**0.3449±0.0008**	0.3498±0.0005
		192	0.6391±0.0511	0.3986±0.0007	0.4128±0.0042	0.4387±0.0069	**0.3964±0.0005**	0.3979±0.0001
		336	0.7231±0.0182	0.4322±0.0005	0.4436±0.0021	0.4907±0.0067	**0.4304±0.0028**	0.4316±0.0014
		720	0.8273±0.0339	0.4468±0.0010	0.4587±0.0022	0.5983±0.0096	**0.4443±0.0039**	0.4449±0.0018
	ETTm1	96	0.4130±0.0041	0.3749±0.0008	**0.3618±0.0005**	0.3722±0.0014	0.3718±0.0023	0.3703±0.0009
		192	0.4336±0.0069	0.3942±0.0007	**0.3845±0.0004**	0.3966±0.0015	0.3946±0.0021	0.3895±0.0047
		336	0.4675±0.0075	0.4177±0.0012	**0.4067±0.0011**	0.4258±0.0033	0.4202±0.0026	0.4069±0.0037
		720	0.5207±0.0050	0.4556±0.0011	0.4427±0.0010	0.4702±0.0102	0.4528±0.0025	**0.4372±0.0039**
	ETTm2	96	0.3355±0.0176	0.2679±0.0030	0.2732±0.0039	0.2796±0.0087	0.2678±0.0039	**0.2650±0.0029**
		192	0.4281±0.0547	0.3124±0.0004	0.3137±0.0054	0.3282±0.0043	0.3179±0.0031	**0.3035±0.0016**
		336	0.5675±0.0110	0.3513±0.0018	0.3529±0.0050	0.3821±0.0120	0.3517±0.0051	**0.3461±0.0012**
		720	0.7826±0.0163	0.4071±0.0010	0.4047±0.0046	0.4849±0.0044	0.4102±0.0090	**0.4025±0.0025**
	Traffic	96	0.2968±0.0037	**0.2715±0.0007**	0.7828±0.0003	0.3359±0.0017	0.3279±0.0037	0.2745±0.0004
		192	0.3027±0.0053	0.2804±0.0001	0.7871±0.0002	0.3336±0.0021	**0.2735±0.0009**	0.2802±0.0004
		336	0.3150±0.0065	0.2865±0.0005	0.7903±0.0002	0.3425±0.0016	**0.2821±0.0004**	0.2850±0.0006
		720	0.3277±0.0022	0.3033±0.0011	0.7935±0.0002	0.3723±0.0009	0.3066±0.0098	**0.3011±0.0015**
	Weather	96	0.2308±0.0029	**0.2122±0.0013**	0.2330±0.0029	0.2348±0.0040	0.2300±0.0031	0.2145±0.0016
		192	0.2739±0.0046	**0.2557±0.0008**	0.2715±0.0020	0.2787±0.0007	0.2694±0.0063	0.2559±0.0035
		336	0.3225±0.0031	0.2969±0.0005	0.3146±0.0026	0.3236±0.0071	0.3148±0.0008	**0.2881±0.0009**
		720	0.3831±0.0152	0.3504±0.0011	0.3624±0.0011	0.3713±0.0108	0.3739±0.0094	**0.3289±0.0006**

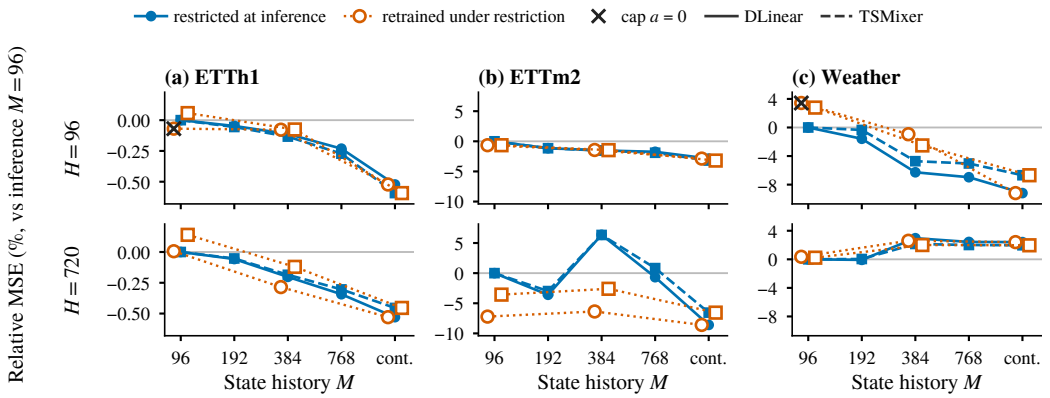

Figure 8: Relative MSE change against the history restricted to M=96 at inference in the same cell, as the history M given to the state grows; *cont.* is the continuous state. Filled markers restrict the history of one checkpoint at inference, open markers are models retrained under the restriction (shifted slightly for visibility), and crosses mark the two retrained runs whose cap a is zero after training. The vertical range differs by column.

Table 29: Continuous-state MSE on Weather for four numbers of harmonic components, and the change when the history is restricted to $M = 96$ at inference, $100\,(\mathrm{MSE}_{M=96} - \mathrm{MSE}_{\mathrm{cont}})/\mathrm{MSE}_{\mathrm{cont}}$, so positive values mean the out-of-window history helps.

Backbone	H	$K=0$ Cont.	$K=0$ M=96 (%)	$K=2$ Cont.	$K=2$ M=96 (%)	$K=4$ Cont.	$K=4$ M=96 (%)	$K=8$ Cont.	$K=8$ M=96 (%)
DLinear	96	0.1954	−0.68	0.1827	+2.42	0.1821	+2.66	0.1716	+10.12
	720	0.3287	+3.51	0.3174	+6.59	0.3227	+4.85	0.3562	−2.35
TSMixer	96	0.1852	−1.30	0.1781	+1.21	0.1774	+1.33	0.1676	+7.16
	720	0.3229	+3.11	0.3169	+4.75	0.3210	+3.38	0.3495	−1.93

$K \in \{0, 2, 4, 8\}$ and restrict the history to $M = 96$ at inference. At $K = 2$ and $K = 4$ the restriction raises the error in all four cells, by 3.4–6.6% at $H = 720$ and 1.2–2.7% at $H = 96$, whereas at $K = 8$ it raises the error at $H = 96$ by 7.2–10.1% but lowers it at $H = 720$ by 1.9–2.4% (Table 29). The level-only state ($K = 0$) uses the long history only at the long horizon: the restriction raises its error at $H = 720$ by 3.1–3.5% and lowers it at $H = 96$ by 0.7–1.3%. With $K = 2$, the value validation selects for Weather in the main table, the continuous harmonic state is lower than the level-only state in all four cells. At $H = 720$ the continuous state with $K = 8$ is also 10–12% worse than with $K = 2$, so the long-period components help at the short horizon but hurt when rolled out over the long one.

A level-only state on all factor cells. Trained in the same configuration on all 12 factor cells, the level-only state (98 parameters) gains 0.41% on average over the backbone at $H = 96$, against 4.85% for $K = 8$, which is ahead in all six cells. At $H = 720$ it is ahead on ETTm2 and Weather, where the fixed $K = 8$ carries the long-period components discussed above, and behind on ETTh1.

Q Scope and Limitations

The filter basis is fixed before training, so continued correction does not recover from a drifting frequency (Appendix F). When a longer window already covers the relevant cycles, its benefit overlaps with that of the carried state (Appendix A). On data without a stable cycle the gain is small (Appendix L).