MAFF: MASK-AWARE FORECASTING FRAMEWORK FOR TIME SERIES WITH MISSING VALUES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployed time series are rarely complete: readings drop out, devices go offline, subsystems fail, and recent observations are lost before they are used. The standard remedy is to impute the gaps and forecast on the completed series, treating the repair as neutral. It is not. Imputed entries enter the instance-normalization statistics that modern forecasters depend on: interpolation deflates the window variance and shifts its mean, the model cannot distinguish fillers from measurements, and the resulting bias is applied back to the forecast at denormalization. We identify this contamination as a mechanistic, architecture-independent source of error, and we eliminate it. MAFF, our mask-aware forecasting framework, keeps observed entries as the only source of statistics, injects a gated summary of the missingness structure at an architecture-adapted position in the backbone, and substitutes a learnable token for fully-missing channels. The repair is parameter-free: it removes the contamination without adding capacity, and parameter-matched ablations isolate its contribution from the wrapper that carries it. Across four missingness protocols (random, block, correlated-group, and tail) at 30% missingness with masks shared across models, MAFF improves MSE on all five benchmarks when coupled with CMoS. We evaluate four backbone families independently: every one of them improves, and the gains are strongest and most consistent on CMoS, the convolution-based backbone whose statistics are most exposed to fillers.

1 INTRODUCTION

Time series collected from real-world systems are rarely complete. Sensors drop readings under communication noise, equipment restarts create temporal gaps, a faulty subsystem takes out a group of correlated channels at once, and data acquisition pipelines may lose the most recent observations before deployment. These failure modes produce missingness with distinct structural shapes: scattered points, temporal blocks, correlated groups, and tail truncation. Each shape carries different information about *why* and *where* data are absent.

The standard recipe in forecasting pipelines is to impute missing entries (linear interpolation, forward-fill) and then feed the completed series to a forecasting model unchanged. This recipe discards the missingness structure itself: the model cannot tell observed values from imputed ones, so it cannot discount unreliable input or exploit the pattern of missingness. The more damaging consequence is statistical. Imputed values enter the statistics that instance normalization layers such as RevIN (Kim et al., 2022) compute. Interpolation smooths the missing regions and deflates the window variance, and the resulting bias survives denormalization to corrupt the final prediction. Figure 1(a) visualizes this distortion, and Figure 1(b) quantifies its consequence on the CMoS backbone: windows whose interpolation deviates most from the ground truth are also those where the mask-aware gain is largest.

The difficulty runs deeper than the accuracy of the fillers. Fillers are consumed as measurements, and modern forecasters process them through components designed for complete data. Two properties make the resulting error systematic. First, the missingness pattern itself is informative: whether a gap is a single dropped reading or a multi-hour outage, whether it affects one sensor or an entire

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.

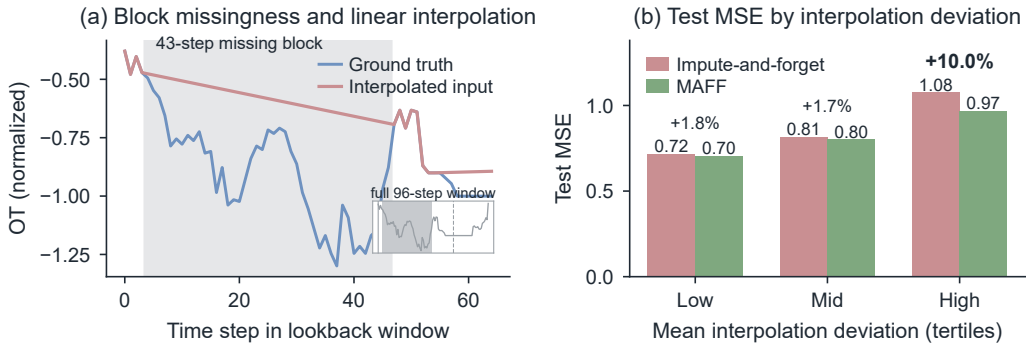

Figure 1: Imputation artifacts under block missingness and their relation to mask-aware gains (ETTh1, block missingness protocol, CMoS backbone, horizon 96; panel (b) uses a single mask seed). (a) First 65 steps of the look-back window: blue denotes observed values, rose the model input after linear interpolation, and the shaded region a contiguous 43-step missing block; the inset marks the block's position within the full 96-step window. (b) All 2,785 test windows are grouped into tertiles by the average deviation of interpolation from ground truth, comparing impute-and-forget with MAFF: the two least-affected groups reduce MSE by 1.8% and 1.7%, and the most affected group by 10.0%.

subsystem, and whether it truncates the most recent observations are all signals about how much the window can be trusted, yet an imputed series erases every one of them. Second, the components that make modern forecasters work are precisely the ones most exposed to fillers, and instance normalization is the clearest case: the distortion of the statistics is applied to the forecast as well as to the input. Scaling the backbone cannot recover either effect, because both are determined by the input representation rather than by the mapping the model learns.

We address this with MAFF, a Mask-Aware Forecasting Framework that treats missingness as first-class information. Three components do the work. *Mask-aware normalization* computes per-window mean and variance over observed entries only, so imputed values never contaminate the normalization statistics; it drops into any backbone with internal instance normalization and replaces that layer outright. *Gate-modulated mask embeddings* summarize missingness (per-channel missing rate, longest missing run, fully-missing indicator) and inject them into the feature stream at a position matched to the backbone: after normalization for convolutional and cyclic backbones, at the embedding level for Transformer-based ones, and at the input level for linear ones. A learned scale starts tiny, so the mask signal never dominates the primary signal. *A learnable MASK token* guards the degenerate case of a channel with no observations at all, so attention and convolution never consume fake zeros. Training and inference follow the *same* missingness protocol, which aligns the input distribution the model learns with the one it faces at deployment.

We evaluate MAFF on four representative architectures, CMoS (convolutional), CycleNet (cyclic), iTransformer (Transformer), and DLinear (linear), across five standard benchmarks (ETTh1, ETTh2, ETTm1, ETTm2, Weather) and four horizons under four missingness protocols at 30% missingness, with mask patterns shared across models for a controlled comparison. Four findings follow. (1) CMoS, whose local convolutional receptive fields amplify imputation artifacts, benefits most, with the improvement holding across all horizons and protocols on ETTh1 and ranging from $+3.7\%$ to $+9.3\%$ in MSE when aggregated by protocol. (2) CycleNet is structurally robust to local missingness, because its cycle queue has already memorized the historical cycle shape of each channel, and its gain is correspondingly small (-0.8% to $+2.5\%$ across datasets and protocols). (3) iTransformer and DLinear gain less, because their compact representation pathways leave little capacity for the injected features to act on. (4) Across all four, the gain tracks the match between the backbone and the missingness structure. We report the negative cases with the same precision as the positive ones, because they delineate that match: a minority of configurations regress, at most -1.2% per dataset–protocol configuration when aggregated over horizons, and down to -5.3% in the least favorable individual horizon cell (ETTh2, DLinear, P4, h=336). Mask injection is not free on every backbone–protocol pair, and identifying which pairs pay for it is part of the contribution.

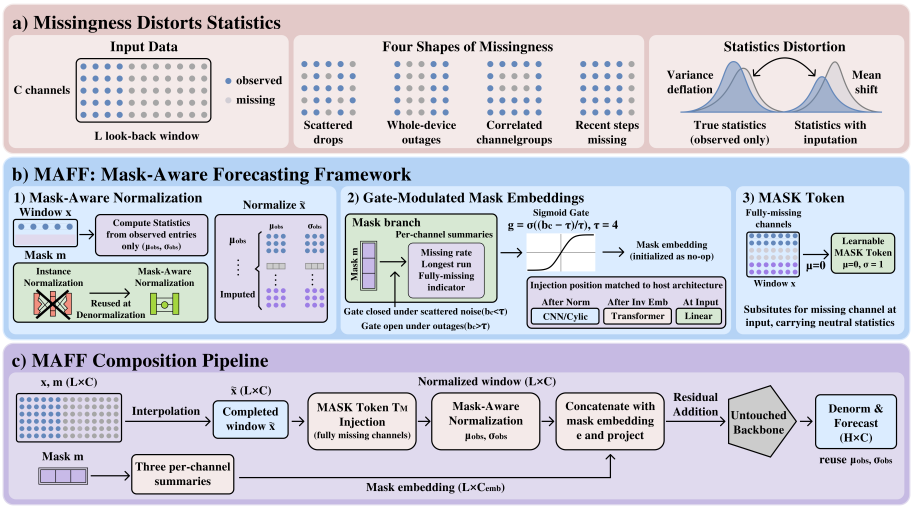

Figure 2: **(a)** Missingness distorts the statistics that instance normalization computes: interpolation deflates the window variance and shifts its mean, and the distortion returns to the forecast at denormalization. **(b)** MAFF modifies an impute-and-forecast backbone in three places. Mask-aware normalization computes the statistics over observed entries only, so fillers never reach μ and σ; gate-modulated mask embeddings encode the missingness pattern (per-channel missing rate, longest run, fully-missing indicator) and inject it at a position matched to the backbone; and a learnable MASK token replaces channels with no observations, carrying neutral statistics so it passes normalization losslessly. **(c)** The three compose into a single pipeline around the backbone: the token substitutes for fully-missing channels before normalization, the normalized window is concatenated with the mask embedding and projected, and denormalization reuses the same statistics. On a fully observed window the injections start from the native path, since the identity initialization contributes nothing until training moves it.

Our contributions are summarized as follows:

- We identify instance-normalization statistics contamination by imputed values as a mechanistic, architecture-independent source of forecast error, and eliminate it with mask-aware normalization.
- We propose MAFF, a three-component mask-aware framework whose injection mechanism adapts to backbone structure and requires no change to the backbone's training recipe.
- We design four missingness protocols mirroring real failure modes, with shared mask patterns across models, and study the interaction between missingness sensitivity and forecast architecture.
- We evaluate MAFF on four backbone families and five benchmarks and characterize where mask-aware treatment pays off: gains are largest on the convolutional CMoS backbone, whose statistics are most exposed to imputation artifacts, and no configuration regresses by more than 1.2% at the dataset–protocol level.

2 Related Work

Missing-value handling in time series. Missing-value imputation is a classical topic. Statistical interpolation (linear, spline, forward-fill) exploits temporal locality and remains the engineering default, matrix-completion methods exploit low-rank structure, and deep approaches such as BRITS (Cao et al., 2018) and SAITS (Du et al., 2023) learn the missingness through recurrent or attention dynamics; see Wang et al. (2025) for a survey. All of these target the intermediate step of "completing the data," and two problems follow. Learned imputation adds training and storage cost and depends on the fit between the missing mechanism and the training data. More fundamentally, however accurate the imputation, the downstream model cannot distinguish observed from imputed

entries: the information carried by the missingness structure is erased, and any imputation error is consumed as if it were real. That failure is independent of filler quality, which is why we remove the contamination at its source rather than add a better filler. A recent line instead forecasts directly from the partially observed series: S4M (Peng et al., 2025), GinAR (Yu et al., 2024) and CRIB (Yang et al., 2025), which reports that unsupervised imputation can corrupt the data distribution and degrade prediction. We keep the impute-and-forecast pipeline that practice already uses and locate the damage in the statistics the fillers enter, which can be repaired without changing the backbone.

Masked and missingness-aware modeling. Masking is a standard self-supervised device, from masked language modeling to MAE (He et al., 2022). Ti-MAE (Li et al., 2023), SimMTM (Dong et al., 2023) and PatchTST (Nie et al., 2023) carry the idea to time series as a pre-training task, in a regime where masks are a training-time device and inputs are complete at inference. MAFF addresses missingness that exists at deployment, with masks present in training and inference alike, and the model learns to forecast *from* missing inputs rather than to reconstruct masked content. Works that feed missingness indicators as features (Che et al., 2018) leave the instance-normalization layer untouched, which is exactly where MAFF's core mechanism operates.

Instance normalization and distribution shift. RevIN (Kim et al., 2022) computes per-window statistics, normalizes the input, and denormalizes the forecast, mitigating distribution shift; follow-ups include Non-stationary Transformers (Liu et al., 2022b) and adaptive refinements of RevIN (Liu et al., 2023). All of these implicitly assume complete windows: statistics are computed over all time steps, so once missing values are filled, fillers enter the statistics. MAFF handles missingness at the instance normalization level directly, and restricting the statistics to observed entries cuts the contamination path in any backbone that contains internal instance normalization.

3 METHOD

Problem setup. Figure 2 gives an overview of the framework: an ordinary impute-and-forecast backbone, modified in exactly three places. The rest of this section walks through the three modifications. Given a look-back window $\mathbf{x} \in \mathbb{R}^{L\times C}$ with L time steps and C channels, the task is to forecast the next H steps $\hat{\mathbf{y}} \in \mathbb{R}^{H\times C}$. Under missingness the input window contains missing entries indicated by a binary mask $\mathbf{m} \in \{0,1\}^{L\times C}$ (0 = missing). We fill missing entries by per-channel linear interpolation over time, obtaining the completed input $\tilde{\mathbf{x}}$, and additionally provide the mask $\mathbf{m}$ to the model. Four protocols mirror real failure modes: **P1** random pointwise missingness (communication noise); **P2** temporal block missingness (intermittent device outage); **P3** correlated channel-group missingness (subsystem failure); **P4** tail missingness of the look-back window (deployment-stage data loss). Each protocol targets a 30% missing rate, and models share identical mask patterns given the same mask seed. P2 is the most destructive for input information: it drops entire blocks across *all* channels with overlaps allowed, so every covered time step is missing in all channels at once.

Mask-aware normalization. Instance normalization is a standard component of modern forecasters: it computes window mean and variance and standardizes the input. When the input contains fillers, interpolation smooths the missing regions and deflates the variance entering the statistics, and the distortion propagates to the forecast through denormalization. Mask-aware normalization removes it at the source by restricting the statistics to observed entries (Equation 1):

$$\mu_c = \frac{1}{\sum_l m_{l,c}} \sum_{l:\, m_{l,c}=1} x_{l,c}, \qquad \sigma_c^2 = \frac{1}{\sum_l m_{l,c}} \sum_{l:\, m_{l,c}=1} (x_{l,c} - \mu_c)^2, \tag{1}$$

so fillers never participate in any statistic. Denormalization uses the same statistics for consistency. All divisions carry a numerical lower bound to avoid division by zero for fully-missing channels. When the mask is all ones, mask-aware normalization reduces to standard instance normalization, so behavior on clean input matches the native backbone up to the normalization epsilon (10^{-5} in the mask-aware path; individual backbones may use a different constant, e.g. CMoS uses 10^{-10}).

Gate-modulated mask embeddings. Beyond repairing statistics, MAFF encodes the missingness structure as features. For each channel we extract three summaries: the per-channel missing rate

$r_c = 1 - \frac{1}{L}\sum_l m_{l,c}$, the longest contiguous missing run b_c (in time steps), and the fully-missing indicator $\mathbb{1}[r_c = 1]$. The summaries pass a structural gate and are mapped to a fixed-dimensional mask feature: the gate applies a sigmoid compression of the longest run, $g = \sigma((b_c - \tau)/\tau)$ with $\tau = 4$ steps, weighting features by run length. Under random pointwise missingness (P1) the median longest run is about 3 steps, $g \approx 0.44$, so the features are attenuated; under block, group, and tail missingness (runs of tens of steps) g approaches 1 and the gate opens fully. The gate is an initialization-time prior rather than a tuned mechanism: varying τ over $\{1, 4, 16\}$ leaves the results unchanged (Section 6). The mask features then pass a GELU projection to produce the mask embedding.

Architecture-adapted injection. Backbones differ sharply in internal structure, and mask features should enter the feature stream without disturbing the primary signal. For backbones with internal instance normalization and convolutional or recurrent structure (CMoS, CycleNet), mask features are injected after normalization: each channel's mask feature is concatenated to its normalized sequence and mapped back to the original length by a linear projection over time whose weights are initialized to identity on the sequence block and zero on the mask block, scaled by a learnable scalar initialized to 0.01 and added residually. Identity initialization guarantees that a zero mask feature contributes nothing beyond the backbone's own path, and the tiny initial scale keeps the injected signal from swamping the normalization statistics on small-variance datasets (e.g., ETTh2). For the Transformer backbone (iTransformer), mask features are injected after the inverted embedding, aligned per variable token with the representation consumed by attention, with the same identity-plus-zero initialization. For the linear backbone (DLinear), which has no normalization layer, mask features are concatenated to the input and passed through the same identity projection residually. All three injections share one principle: at initialization, with zero mask features, the output is bit-identical to the native backbone (degenerate invariance). The identity blocks are trained like any other parameter, so the property describes the starting point rather than the trained model. For fully missing channels, MAFF replaces the input with a learnable MASK token so normalization and subsequent operations never act on invalid fillers. To let the token pass normalization losslessly, the statistics of a fully-missing channel are fixed to neutral values (mean 0, standard deviation 1). The token guards this degenerate case; under the protocols used here it never activates, as we report in Section 5.

Training setup and baseline. Training and evaluation use consistent protocols: training windows receive masks drawn independently under the same protocol (same distribution as test masks, different instances), the model trains on completed inputs, and forecast targets remain complete. The baseline is *impute-and-forget*: it receives the identical completed input but is mask-blind and follows the native normalization and inference path, so the two arms differ only in whether missingness is used. MAFF is an external wrapper that requires no change to the backbone's loss or training recipe; all components activate only in the mask-aware mode (*full* against the mask-blind *off*), and the mask-aware path reduces to the native forward path up to normalization epsilon when no missingness is present.

4 Experimental setup

Datasets. We evaluate on five standard long-term forecasting benchmarks that span three sampling frequencies and two channel regimes: ETTh1 and ETTh2 (hourly, 7 channels, 17,420 time steps), ETTm1 and ETTm2 (15-minute, 7 channels, 69,680 steps), and Weather (10-minute, 21 channels, 52,696 steps). The ETT datasets follow the standard 12/4/4-month train/validation/test split and Weather the standard 70%/10%/20% split; with a look-back of 96 and a horizon of 96 this yields 2,785 test windows on the ETT hourly datasets, 11,425 on the ETT minute datasets, and 10,444 on Weather. We use the standard per-channel standardization fitted on the training split.

Backbones. Four backbones span the dominant architectural families: CMoS (Si et al., 2025) (chunk-wise spatial correlations), CycleNet (Lin et al., 2024) (learnable recurrent cycles), iTransformer (Liu et al., 2024) (inverted attention over variate tokens), and DLinear (Zeng et al., 2023) (per-channel linear decomposition). All four use their public implementations with a look-back of 96 and horizons 96/192/336/720; the remaining hyperparameters are listed in Appendix G.

Table 1: MSE improvement (%) of MAFF over the impute-and-forget baseline, per dataset, missingness protocol (including the complete-data control *none*), backbone, and forecast horizon (3 mask seeds; gains computed from summed MSE as (off−full)/off, in percent). Blue marks gains beyond 0.5% (bold beyond 5%), red marks regressions beyond 0.5%, and gray marks differences within the $\pm 0.5\%$ error band. Under *none* the mask is all ones, so no missingness information is available and the entries reflect training-path differences (wrapper parameters and normalization epsilon) rather than a mechanism effect; they are shown in gray and never in bold regardless of magnitude, and should be read as an offset rather than a gain.

		CMoS				CycleNet				iTransformer				DLinear			
Dataset	Prot.	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
ETTh1	none	+7.7	+5.3	+3.7	+3.0	+1.8	+0.4	+0.2	+0.1	-1.0	-0.8	+0.9	+0.4	+0.1	-0.0	-0.8	+0.3
	P1	**+7.3**	+4.8	+3.3	+2.5	+0.9	-0.2	-0.4	-0.5	-1.5	-1.0	+0.5	-0.3	+0.1	-0.0	-0.8	+0.3
	P2	+5.0	+4.2	+3.1	+2.5	+2.8	+1.7	+1.5	+1.4	+1.5	+1.1	+1.8	+2.8	+0.3	+0.2	+0.4	+0.1
	P3	**+11.4**	**+8.2**	**+6.6**	**+5.9**	+4.1	+2.4	+2.0	+1.8	+2.0	+0.1	+3.9	-1.4	+0.3	+0.1	-0.8	+0.5
	P4	**+12.6**	**+9.0**	**+8.3**	**+7.5**	+3.3	+1.9	+1.4	+1.3	-0.5	+2.1	+3.8	-2.7	-0.4	+0.1	-0.9	+0.3
ETTh2	none	+1.1	+0.4	+0.3	+0.4	+0.6	-0.8	+0.1	+0.1	-0.6	+2.0	+6.0	-4.7	-2.5	+1.8	-1.0	+0.2
	P1	+0.9	+0.3	+0.2	+0.3	+0.2	-1.1	-0.1	-0.1	-0.6	+1.4	+2.9	-2.2	-2.7	+1.7	-1.1	+0.2
	P2	+1.6	+1.3	+0.8	+0.5	+0.6	+0.4	+0.6	+0.6	-0.3	+0.8	+0.9	+1.5	-1.4	+2.6	-1.3	-0.2
	P3	+1.0	+0.3	+0.3	+0.2	-0.3	-1.4	-0.2	-0.3	-1.5	+1.8	+0.4	+4.3	-2.4	+1.6	-0.8	+0.3
	P4	+1.2	+0.9	+1.0	+1.2	+1.5	-0.2	+0.8	+0.7	-0.3	+2.9	+1.9	+2.2	+0.3	**+5.0**	-5.3	+1.7
ETTm1	none	+6.5	+4.0	+4.1	+2.1	+0.7	+0.3	+0.1	+0.1	+0.5	-3.3	-0.1	-0.3	+0.2	+0.1	+0.5	-0.3
	P1	**+5.8**	+3.2	+3.4	+1.4	+0.2	-0.1	-0.2	-0.3	+1.9	-1.6	+2.7	-0.5	+0.2	+0.1	+0.5	-0.3
	P2	+1.8	-0.2	+0.1	-0.8	-0.1	-0.6	-0.9	-1.3	+1.4	+2.2	+1.9	+1.2	-0.0	-0.3	+0.2	-0.2
	P3	**+5.2**	+2.9	+2.4	-0.2	+0.3	-0.1	-0.2	-0.5	-0.1	-1.1	-1.7	-1.5	+0.2	+0.1	+0.5	-0.3
	P4	**+5.1**	+4.6	+3.2	+2.3	+0.8	+0.7	+0.5	+0.3	-0.2	-0.1	-0.4	-0.9	+0.1	+0.0	+0.6	-0.1
ETTm2	none	+3.9	+1.6	+0.7	+0.6	+0.2	+0.1	+0.1	+0.1	+0.6	-0.7	-1.4	+0.5	-0.8	+0.5	+0.3	+2.0
	P1	+3.5	+1.3	+0.4	+0.4	-0.1	-0.1	-0.1	-0.0	+0.8	-1.4	+2.4	+1.3	-0.9	+0.4	+0.2	+2.0
	P2	+2.5	+0.9	+0.3	-0.0	+0.3	+0.1	+0.1	+0.0	+0.2	+0.4	-0.2	-0.8	-0.3	+0.3	-2.0	+1.3
	P3	+3.7	+1.3	+0.2	+0.2	+0.3	+0.2	+0.2	+0.1	+0.5	-2.3	+0.8	+0.3	-0.9	+0.6	+0.1	+2.1
	P4	+0.9	+0.7	+0.5	+0.4	+0.8	+0.5	+0.3	+0.2	+2.6	-0.3	+1.5	+0.4	-0.4	+0.2	-0.2	+1.3
Weather	none	+2.0	+2.1	+0.7	+0.5	+0.2	+0.2	+0.1	+0.1	-2.4	-1.1	+0.1	-0.6	+0.3	+0.9	-0.1	+0.1
	P1	+1.8	+1.7	+0.5	+0.4	-0.0	-0.1	-0.1	-0.1	-0.2	-0.9	-0.9	-0.4	+0.3	+1.0	-0.1	+0.1
	P2	+1.0	+1.8	+0.5	+0.6	+0.3	+0.1	-0.0	-0.1	+0.6	+0.5	+0.4	-0.1	+0.4	+2.9	+1.6	+0.1
	P3	+1.4	+1.5	+0.4	+0.3	-0.0	-0.2	-0.3	-0.4	+1.1	+0.0	+0.3	+0.2	+0.3	+0.6	-0.3	+0.1
	P4	-0.1	-0.1	-0.4	-0.0	-0.0	+0.3	+0.0	-0.0	+1.3	+2.2	+0.4	+0.8	+1.1	+0.1	-0.9	+0.2

Missingness protocols. Four protocols mirror the failure modes described in Section 1: **P1** random pointwise missingness (missing completely at random, communication noise); **P2** temporal block missingness covering all channels simultaneously (intermittent device outage); **P3** correlated channel-group missingness (subsystem failure), where channels are partitioned into contiguous groups and each group receives a temporal block plus scattered points; **P4** tail missingness of the look-back window (deployment-stage data loss). Each protocol targets a 30% missing rate, and the same mask seed produces identical masks for every model, so all comparisons are made on bit-identical inputs.

Training. Every configuration is repeated over 3 mask seeds whose runs are averaged, and the training seed is fixed, so the reported variance reflects the missingness realization rather than weight initialization. All experiments run on a single NVIDIA GPU. MAFF leaves the training objective untouched: the wrapper shares the backbone's loss and optimizer, and its parameters are trained jointly from the identity initialization. Appendix G lists the full recipe.

Baseline and metric. The baseline is *impute-and-forget*: the native backbone consuming the same linearly interpolated input with its own instance normalization and no access to the mask. It is the recipe deployed pipelines actually use, so it is the comparison that matters. MAFF and the baseline consume bit-identical inputs, so the off/full contrast is attributable to the wrapper alone, and the clean-data control fixes how much of it comes from the wrapper's own parameters. We report MSE and MAE (mean squared and mean absolute error) as relative gains from summed statistics, $(s_{\text{off}} - s_{\text{full}})/s_{\text{off}}$, where the sums run over all test windows, channels, and seeds of a cell; summed statistics are robust to per-window normalization effects and to the heavy-tailed error distributions typical of Weather.

5 RESULTS

Main results. Table 1 reports the MSE change of MAFF relative to the baseline over the 80 dataset–backbone–protocol configurations (5 datasets $\times$ 4 backbones $\times$ 4 protocols). Aggregated gains in this section are computed over the four missingness protocols; the *none* row of Table 1 is a clean-data control and is excluded from them. CMoS benefits most, and it benefits on every dataset: its aggregate gain is positive on all five ($+0.6\%$ to $+6.2\%$), $+2.7\%$ overall, and on ETTh1 the improvement holds across all horizons and protocols ($+3.7\%$ to $+9.3\%$). The remaining backbones gain less but still positively on average: $+0.7\%$ for iTransformer, $+0.4\%$ for CycleNet, and $+0.2\%$ for DLinear. At the dataset–protocol level the sign varies, with 57 of 80 configurations positive; the 23 that regress stay within -1.2% at that granularity and reach -5.3% in the least favorable individual horizon cell, which is the architecture-dependence anticipated in Section 1. The clean-data control (protocol *none*) bounds how much of this comes from the wrapper rather than from missingness: 16 of the 20 dataset–backbone combinations differ within $\pm 0.9\%$ once horizons are pooled, and the exceptions are all CMoS (ETTh1 $+4.8\%$, ETTm1 $+4.0\%$, ETTm2 $+1.4\%$, Weather $+1.2\%$); individual horizon cells deviate further, up to $+7.7\%$ (ETTh1, h=96). Under *none* the mask is all ones, so no missingness information is available, and the residual difference comes from the wrapper's additional parameters together with the normalization epsilon, which is 10^{-5} in the mask-aware path and 10^{-10} in CMoS's native branch. We therefore read the protocol rows against this control: gains under structural missingness exceed it (ETTh1 P3/P4 $+7.9\%$ and $+9.3\%$ against a $+4.8\%$ offset), while the P2 gain on CMoS ($+1.6\%$ aggregate) does not, consistent with P2 being destructive enough to remove most of the exploitable structure. The component ablation isolates the mechanism itself on ETTh1: with the baseline's parameter count, mask-aware normalization alone recovers $+2.8\%$, so the statistics repair contributes independently of the wrapper's added capacity, and that is the component the paper is built on. Appendix Figures 8–10 show the same comparison at the level of individual forecasts, forecast steps, and test windows.

Effect of missingness structure. Aggregating CMoS by protocol across the five datasets gives a stable ordering of gains: tail $+3.8\%$, correlated-group $+3.2\%$, random scatter $+2.3\%$, block $+1.6\%$. The ordering inverts the intuition that more destruction means more to repair: block missingness is the most destructive protocol and yet yields the smallest gain. The reason is an asymmetry between two kinds of information. The *position* of tail and group missingness is itself a strong prior ("the recent window is unreliable," "the whole group is out"), and the mask embedding needs only to convey position to collect most of the gain, whereas the artifacts of block missingness are concentrated inside the block, where both methods are equally inaccurate. The gain tracks how much exploitable information the missingness structure carries, not how severe it is.

Effect of backbone. The four families respond differently, and the pattern is systematic rather than incidental. CMoS and CycleNet are both channel-wise models with internal instance normalization and after-normalization injection, which leaves the primary pathway untouched; they achieve $+2.7\%$ (CMoS, five-dataset mean) and $+0.4\%$ (CycleNet). DLinear stays between -0.3% and $+1.2\%$ when aggregated per dataset and protocol, and iTransformer averages $+0.7\%$, between -1.2% and $+1.8\%$. Absolute MSE values for CMoS are in Table 2, and the per-protocol gains of all four backbones in Table 7.

Efficiency. MAFF adds no learned imputer, no auxiliary objective, and no separate imputation pipeline at deployment, so its cost is the backbone's cost. Figure 3 gives the per-epoch training-time ratio of each full run against its off counterpart, over 240 pairs per backbone. The medians are 0.99 and 1.00 for CMoS and CycleNet and 0.77 for DLinear: the third, below one, shows the ratios carry scheduling noise from the shared GPU rather than a wrapper cost. iTransformer is the one systematic case, at 1.34, because its projection acts on the variate tokens after the inverted embedding. Twenty runs shared one GPU, so we read these ratios as an upper bound on the wrapper's cost.

What the mask pathway actually does. Figure 4 shows what the gate receives. The gate value it implies sits at 0.47 under P1, where the missing runs straddle the threshold, and above 0.95 under the three structural protocols, where it opens fully, which is the discrimination the design intends. The MASK token is inactive in these experiments: no channel is fully missing in any test window of any dataset under any protocol, so the token never fires. We report it as a robustness guarantee for the degenerate case, not as a source of the measured gains. The same reading applies to the clean-data control, whose residual off/full differences reflect the wrapper's training path, not the mechanism.

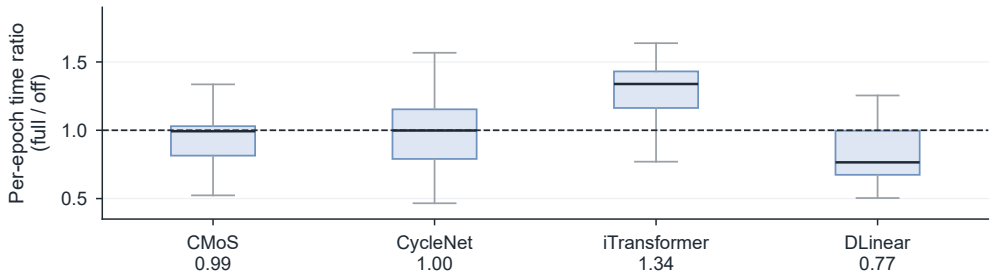

Figure 3: Per-epoch training-time ratio between the full and off runs of the same configuration, over 240 pairs per backbone. Boxes span the interquartile range, the bar and the number under each label give the median, and the dashed line marks equal cost.

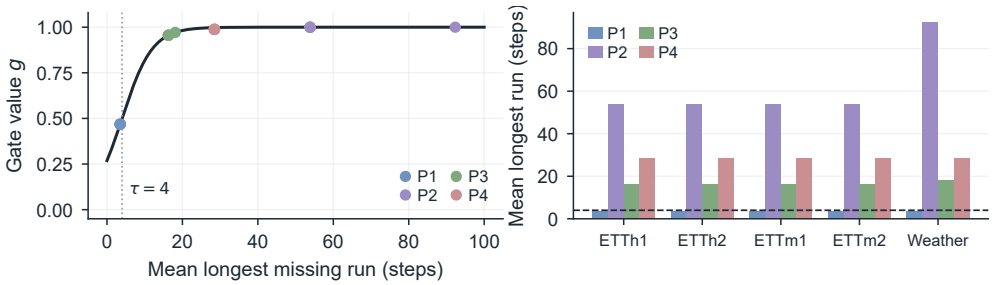

Figure 4: Gate transfer curve and protocol operating points (left), and the mean longest missing run per dataset and protocol (right), both from masks regenerated with the mask generator and seeds used in the experiments (ratio 0.3, look-back 96, three seeds). The dashed line marks the gate threshold $\tau = 4$.

Where mask awareness pays off. The differences among the four backbones follow one rule, and it is the rule the introduction anticipated. DLinear's per-channel linear map has little representation capacity to exploit an injected feature, and it carries no internal instance normalization for the statistics repair to act on, so both mechanisms are idle. CycleNet's cycle queue already encodes the dominant periodic structure of each channel, which makes it robust to localized missingness and leaves little for mask features to add. iTransformer occupies the informative middle: injecting per-variate features into attention helps most on structured missingness (P2: $+1.1\%$, P4: $+0.8\%$) and least when the missingness is scattered (P1: $+0.2\%$) or concentrated in correlated groups (P3: $+0.4\%$). The rule holds across the four: mask injection pays off when the backbone has both statistics that fillers can contaminate and spare capacity in the pathway where the features are injected.

6 Ablation Studies

Components. MAFF has two active components: mask-aware normalization (which repairs the statistics) and mask-feature injection (which tells the model where the gaps are). We isolate them with two intermediate arms: *norm only* (`fill+mask`, mask forwarded for statistics but no features injected) and *injection only* (`mask`, features injected while the backbone keeps its filler-contaminated normalization). Figure 5 and Table 8 report all four arms against the mask-blind baseline on ETTh1. On CMoS, normalization alone recovers $+2.8\%$ and injection alone $+4.2\%$, while together they reach $+6.9\%$, close to the sum: the two components address largely independent error sources and compose almost additively. The *norm only* arm carries the baseline's parameter count, so its share of the gain cannot come from added capacity, which is what separates the mechanism from the wrapper. The ordering of the two depends on the protocol: injection dominates under scattered missingness (P1: $+5.6\%$ against -0.5%; with runs of only 3.5 steps the contaminated statistics hurt less than the loss of positional information), whereas normalization dominates under block missingness (P2: $+3.0\%$ against $+1.1\%$; with most of the window gone the statistics are the dominant defect). CycleNet reproduces the same pattern at roughly a third of the magnitude.

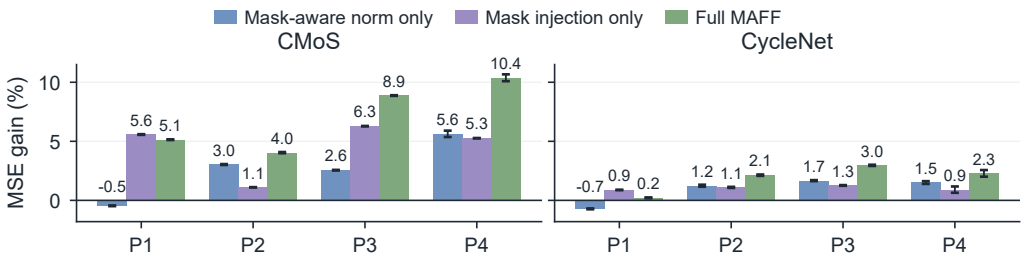

Figure 5: Component ablation on ETTh1 (horizons 96 and 336, 3 mask seeds, error bars show the spread across seeds). *Norm only* forwards the mask for statistics without injecting features; *injection only* injects features while leaving the backbone's filler-contaminated normalization untouched. The two components address largely independent error sources and combine almost additively on CMoS.

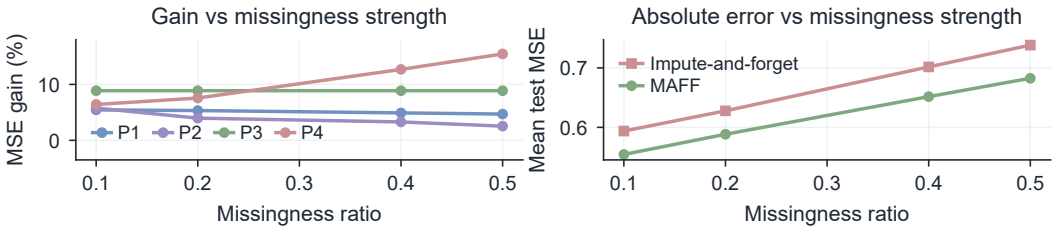

Figure 6: Effect of missingness strength (CMoS, ETTh1). Left: gain per protocol; tail missingness gains monotonically, block missingness saturates and then declines, and correlated-group missingness is flat. Right: mean test MSE of both arms.

Missingness strength. Figure 6 and Table 9 sweep the ratio from 0.1 to 0.5; the main experiments use 0.3. The protocols respond in opposite directions. Under tail missingness (P4) the gain grows monotonically with strength, from $+6.4\%$ to $+15.4\%$ at ratios 0.1–0.5, exactly what the design predicts when the mask carries information proportional to the gap size. Under block missingness (P2) the gain instead *shrinks* ($+5.8\%$, $+4.0\%$, $+3.3\%$, $+2.5\%$). Random missingness is comparatively flat ($+5.4\% \rightarrow +4.7\%$), while correlated-group missingness is flat across the sweep ($+8.9\%$ at every ratio). The two directions are consistent with one reading: the advantage of MAFF scales with the *recoverable* structure in the window, not with raw missingness. Stronger tail missingness adds recoverable structure; stronger block missingness removes it.

Gate threshold and fill strategy. Varying the gate threshold over $\tau \in \{1, 4, 16\}$ (Figure 7, left) changes MSE by less than $2 \times 10^{-3}\%$: the projection downstream rescales whatever the gate attenuates. The fill strategy matters more: zero-filling and repeat-last beat the baseline under structural missingness (P3: $+11.7\%$, $+8.7\%$; P4: $+12.2\%$, $+10.8\%$); zero-filling matches the linear-interpolation reference on P3 ($+11.7\%$ against $+11.4\%$) while repeat-last trails it, and under P1 both fall below the baseline (-5.0%, -5.6%): MAFF repairs statistics and injects structure, it does not rescue an input that is actively misleading.

7 Conclusion

We presented MAFF, a mask-aware forecasting framework that treats missingness as first-class information. Mask-aware normalization removes the imputation contamination of instance-normalization statistics at the source, gated mask embeddings inject missingness features at positions matched to the backbone, and a learnable MASK token guards the fully missing channel. Across five benchmarks, four backbone families and four missingness protocols, it gains most on the backbone whose statistics are most exposed to imputation artifacts, and the ordering of the gains follows the match between the mechanism and the backbone rather than the severity of the missingness. The repair is parameter-free, acts at the normalization layer, and leaves the backbone's training recipe untouched, so it drops into existing impute-and-forecast pipelines without retuning them.

AI USE STATEMENT

In this work, generative AI tools were used only to assist with language polishing and improve the clarity and readability of the manuscript.

REPRODUCIBILITY STATEMENT

All experiments use shared mask patterns generated deterministically from mask seeds, with the training seed fixed. Appendix G lists the training protocol and hyperparameters, Appendix A gives the absolute per-cell MSE of both arms for all four backbones, and Appendix B the per-backbone and per-protocol breakdown. The sweep harness, mask generator, and evaluation scripts accompany the submission as anonymous supplementary material. Each configuration is repeated over 3 mask seeds; gains are computed from summed MSE as (off−full)/off.

REFERENCES

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. BRITS: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, 2018.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. SimMTM: A simple pre-training framework for masked time-series modeling. In *Advances in Neural Information Processing Systems*, 2023.

Wenjun Du, David Côté, and Yan Liu. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023.

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 *AAAI Conference on Artificial Intelligence*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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.

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

Zhe Li, Zhongwen Rao, Lujia Pan, Pengyun Wang, and Zenglin Xu. Ti-MAE: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. CycleNet: Enhancing time series forecasting through modeling periodic patterns. In *Advances in Neural Information Processing Systems*, 2024.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. In *Advances in Neural Information Processing Systems*, 2022a.

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*, 2022b.

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*, 2023.

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *International Conference on Learning Representations*, 2024.

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.

Jing Peng, Meiqi Yang, Qiong Zhang, and Xiaoxiao Li. S4M: S4 for multivariate time series forecasting with missing values. In *International Conference on Learning Representations*, 2025.

Haotian Si, Changhua Pei, Jianhui Li, Dan Pei, and Gaogang Xie. CMoS: Rethinking time series prediction through the lens of chunk-wise spatial correlations. In *International Conference on Machine Learning*, 2025.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, 2021.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011.

Jun Wang, Wenjie Du, Yiyuan Yang, Linglong Qian, Wei Cao, Keli Zhang, Wenjia Wang, Yuxuan Liang, and Qingsong Wen. Deep learning for multivariate time series imputation: A survey. In *International Joint Conference on Artificial Intelligence*, pp. 10696–10704, 2025.

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

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.

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

Jie Yang, Yifan Hu, Kexin Zhang, Luyang Niu, Philip S. Yu, and Kaize Ding. Revisiting multivariate time series forecasting with missing values. *arXiv preprint arXiv:2509.23494*, 2025.

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.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, 2018.

Jinsung Yoon, William R. Zame, and Mihaela van der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2019.

Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, and Yongjun Xu. GinAR: An end-to-end multivariate time series forecasting model suitable for variable missing. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

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

A FULL RESULTS

Tables 2–5 give the absolute test MSE of both arms for all four backbones on all five datasets, four protocols and four horizons, and Table 6 gives the spread of the gain across the three mask seeds. Read as gains, ETTh1 is the strongest column at every protocol ($+4.4\%$, $+3.7\%$, $+7.9\%$ and $+9.3\%$ for P1–P4), and tail missingness is where the two arms separate most clearly. ETTh2 and Weather carry the smallest CMoS gains, between $+0.3\%$ and $+1.2\%$. Weather under P4 is the only CMoS dataset–protocol cell that turns negative (-0.16%), and it is negative at all four horizons (-0.05% to -0.35%), so that one is systematic rather than a single sampling artifact; Weather's dataset-level aggregate of $+0.62\%$ covers it up. Four further CMoS cells are slightly negative, all under P2 or P3 on ETTm1 and ETTm2, the worst at -0.77% (ETTm1, P2, h=720). The gain also falls with the forecast horizon on every dataset except Weather. On ETTh1 it moves from $+8.7\%$ at h=96 to $+4.5\%$ at h=720; on ETTm2 from $+2.6\%$ to $+0.2\%$. A repaired input statistic is worth most over the steps where the forecast is still close to the look-back window, and least at the far end of a long horizon, which is the pattern the mechanism predicts.

Table 2: Absolute test MSE on the CMoS backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.509 / 0.472	0.558 / 0.531	0.582 / 0.563	0.590 / 0.575
	P2	0.871 / 0.828	0.875 / 0.839	0.876 / 0.849	0.890 / 0.868
	P3	0.567 / 0.503	0.606 / 0.557	0.625 / 0.584	0.635 / 0.598
	P4	0.622 / 0.544	0.659 / 0.599	0.674 / 0.617	0.673 / 0.622
ETTh2	P1	0.346 / 0.343	0.428 / 0.427	0.454 / 0.454	0.454 / 0.453
	P2	0.372 / 0.366	0.451 / 0.445	0.481 / 0.478	0.483 / 0.480
	P3	0.349 / 0.345	0.430 / 0.428	0.456 / 0.455	0.457 / 0.456
	P4	0.396 / 0.391	0.470 / 0.466	0.485 / 0.480	0.480 / 0.475
ETTm1	P1	0.412 / 0.388	0.440 / 0.426	0.470 / 0.454	0.524 / 0.517
	P2	0.706 / 0.693	0.725 / 0.726	0.748 / 0.747	0.791 / 0.797
	P3	0.424 / 0.402	0.453 / 0.439	0.483 / 0.471	0.536 / 0.537
	P4	0.518 / 0.491	0.540 / 0.515	0.556 / 0.538	0.605 / 0.591
ETTm2	P1	0.200 / 0.193	0.260 / 0.256	0.318 / 0.317	0.417 / 0.415
	P2	0.222 / 0.217	0.284 / 0.281	0.343 / 0.342	0.443 / 0.443
	P3	0.201 / 0.194	0.261 / 0.257	0.319 / 0.319	0.418 / 0.417
	P4	0.242 / 0.240	0.295 / 0.293	0.346 / 0.345	0.440 / 0.438
Weather	P1	0.202 / 0.199	0.248 / 0.244	0.299 / 0.297	0.369 / 0.368
	P2	0.233 / 0.231	0.284 / 0.279	0.345 / 0.343	0.426 / 0.424
	P3	0.202 / 0.199	0.249 / 0.245	0.299 / 0.298	0.370 / 0.369
	P4	0.236 / 0.237	0.281 / 0.281	0.328 / 0.329	0.390 / 0.390

Table 3 gives the same breakdown for CycleNet. Its absolute error sits at the same level as CMoS's, but the two arms differ far less: the per-dataset gain is $+1.6\%$ on ETTh1 and within $\pm 0.2\%$ everywhere else, the best cell is ETTh1 under correlated-group missingness ($+2.5\%$) and the worst is ETTm1 under block missingness (-0.8%).

Table 4 covers iTransformer, whose off runs have the lowest absolute error of the four (0.405 on average) and which carries the largest dataset–protocol regression in the study, -1.2% on ETTm1 under correlated-group missingness. Its gains concentrate on block and tail missingness, and on ETTh2 it is the only backbone other than CMoS to gain more than 1%.

Table 5 covers DLinear. Its off and full columns are close to identical in every ETT cell; the two datasets where it moves at all are ETTm2 and Weather, both under block missingness, reaching $+1.2\%$ on Weather. Its per-dataset gain is $+0.5\%$ on those two and 0.0% on the other three.

Table 6 gives the spread of the per-configuration gain across the three mask seeds. The median is 0.10 percentage points, so the comparisons the paper draws rest on differences far larger than seed variation. The spread is not uniform: it reaches 3.1 points on Weather under block missingness for

Table 3: Absolute test MSE on the CycleNet backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.438 / 0.434	0.481 / 0.483	0.520 / 0.522	0.518 / 0.520
	P2	0.779 / 0.757	0.803 / 0.789	0.812 / 0.799	0.816 / 0.805
	P3	0.466 / 0.447	0.508 / 0.495	0.543 / 0.532	0.542 / 0.533
	P4	0.502 / 0.485	0.534 / 0.523	0.559 / 0.551	0.552 / 0.545
ETTh2	P1	0.317 / 0.316	0.402 / 0.406	0.439 / 0.439	0.441 / 0.442
	P2	0.350 / 0.347	0.434 / 0.433	0.473 / 0.471	0.475 / 0.472
	P3	0.320 / 0.321	0.405 / 0.411	0.442 / 0.443	0.445 / 0.446
	P4	0.388 / 0.382	0.457 / 0.458	0.474 / 0.470	0.471 / 0.467
ETTm1	P1	0.364 / 0.364	0.399 / 0.399	0.430 / 0.430	0.489 / 0.490
	P2	0.664 / 0.665	0.694 / 0.698	0.719 / 0.725	0.758 / 0.768
	P3	0.378 / 0.377	0.412 / 0.413	0.443 / 0.444	0.501 / 0.504
	P4	0.465 / 0.462	0.486 / 0.483	0.502 / 0.499	0.555 / 0.553
ETTm2	P1	0.188 / 0.188	0.252 / 0.252	0.313 / 0.313	0.412 / 0.413
	P2	0.213 / 0.213	0.277 / 0.277	0.338 / 0.338	0.439 / 0.439
	P3	0.189 / 0.189	0.253 / 0.253	0.314 / 0.314	0.414 / 0.414
	P4	0.240 / 0.238	0.293 / 0.291	0.344 / 0.343	0.437 / 0.436
Weather	P1	0.199 / 0.199	0.243 / 0.243	0.293 / 0.293	0.364 / 0.365
	P2	0.233 / 0.232	0.280 / 0.279	0.340 / 0.340	0.421 / 0.422
	P3	0.199 / 0.199	0.243 / 0.243	0.293 / 0.294	0.365 / 0.366
	P4	0.239 / 0.239	0.283 / 0.282	0.326 / 0.326	0.389 / 0.389

Table 4: Absolute test MSE on the iTransformer backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.392 / 0.398	0.447 / 0.451	0.496 / 0.493	0.516 / 0.518
	P2	0.620 / 0.610	0.665 / 0.657	0.703 / 0.690	0.735 / 0.715
	P3	0.407 / 0.399	0.455 / 0.454	0.511 / 0.491	0.522 / 0.530
	P4	0.451 / 0.453	0.500 / 0.490	0.531 / 0.511	0.519 / 0.533
ETTh2	P1	0.297 / 0.299	0.384 / 0.378	0.443 / 0.430	0.436 / 0.446
	P2	0.331 / 0.332	0.424 / 0.421	0.464 / 0.460	0.471 / 0.464
	P3	0.296 / 0.300	0.386 / 0.379	0.435 / 0.434	0.461 / 0.441
	P4	0.372 / 0.374	0.450 / 0.437	0.466 / 0.457	0.480 / 0.469
ETTm1	P1	0.347 / 0.340	0.393 / 0.399	0.448 / 0.436	0.497 / 0.500
	P2	0.577 / 0.568	0.618 / 0.604	0.651 / 0.639	0.700 / 0.692
	P3	0.350 / 0.351	0.401 / 0.405	0.447 / 0.455	0.508 / 0.516
	P4	0.425 / 0.426	0.458 / 0.458	0.488 / 0.490	0.543 / 0.548
ETTm2	P1	0.185 / 0.183	0.251 / 0.255	0.319 / 0.312	0.419 / 0.413
	P2	0.205 / 0.204	0.273 / 0.272	0.334 / 0.335	0.433 / 0.437
	P3	0.185 / 0.184	0.252 / 0.258	0.315 / 0.313	0.417 / 0.416
	P4	0.243 / 0.237	0.293 / 0.294	0.349 / 0.344	0.443 / 0.441
Weather	P1	0.175 / 0.175	0.223 / 0.225	0.280 / 0.283	0.357 / 0.358
	P2	0.233 / 0.231	0.282 / 0.280	0.341 / 0.340	0.424 / 0.424
	P3	0.175 / 0.173	0.223 / 0.223	0.280 / 0.279	0.357 / 0.357
	P4	0.224 / 0.221	0.271 / 0.265	0.317 / 0.316	0.385 / 0.382

Table 5: Absolute test MSE on the DLinear backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.402 / 0.402	0.450 / 0.451	0.492 / 0.496	0.517 / 0.516
	P2	0.672 / 0.670	0.697 / 0.696	0.715 / 0.712	0.733 / 0.732
	P3	0.419 / 0.418	0.466 / 0.465	0.506 / 0.510	0.531 / 0.529
	P4	0.461 / 0.463	0.497 / 0.496	0.527 / 0.532	0.547 / 0.545
ETTh2	P1	0.343 / 0.353	0.484 / 0.476	0.594 / 0.601	0.843 / 0.841
	P2	0.403 / 0.409	0.547 / 0.532	0.659 / 0.667	0.913 / 0.915
	P3	0.345 / 0.354	0.486 / 0.479	0.598 / 0.603	0.846 / 0.844
	P4	0.453 / 0.452	0.591 / 0.561	0.654 / 0.689	0.938 / 0.922
ETTm1	P1	0.350 / 0.349	0.385 / 0.384	0.418 / 0.416	0.475 / 0.476
	P2	0.612 / 0.612	0.638 / 0.640	0.661 / 0.660	0.694 / 0.695
	P3	0.361 / 0.360	0.396 / 0.396	0.429 / 0.427	0.485 / 0.487
	P4	0.445 / 0.444	0.465 / 0.465	0.484 / 0.481	0.533 / 0.534
ETTm2	P1	0.195 / 0.197	0.286 / 0.285	0.386 / 0.385	0.557 / 0.546
	P2	0.225 / 0.226	0.317 / 0.316	0.412 / 0.420	0.592 / 0.585
	P3	0.195 / 0.197	0.287 / 0.285	0.386 / 0.386	0.558 / 0.547
	P4	0.260 / 0.261	0.339 / 0.339	0.434 / 0.435	0.599 / 0.591
Weather	P1	0.198 / 0.197	0.240 / 0.238	0.282 / 0.282	0.346 / 0.346
	P2	0.259 / 0.258	0.302 / 0.293	0.348 / 0.342	0.405 / 0.405
	P3	0.197 / 0.196	0.239 / 0.237	0.282 / 0.283	0.346 / 0.346
	P4	0.245 / 0.242	0.284 / 0.284	0.320 / 0.323	0.372 / 0.371

Table 6: Standard deviation of the per-configuration gain across the three mask seeds, in percentage points, aggregated over horizons (summed-MSE ratio computed per seed).

Dataset	Prot.	CMoS	CycleNet	iTransformer	DLinear
ETTh1	P1	0.05	0.06	0.21	0.00
	P2	0.04	0.06	0.61	0.16
	P3	0.02	0.04	0.35	0.02
	P4	0.37	0.24	0.58	0.03
ETTh2	P1	0.04	0.04	1.68	0.01
	P2	0.65	0.30	0.68	1.06
	P3	0.13	0.08	0.18	0.06
	P4	0.23	0.13	0.46	0.27
ETTm1	P1	0.05	0.03	1.84	0.00
	P2	0.01	0.02	0.50	0.06
	P3	0.03	0.01	0.76	0.01
	P4	0.09	0.07	0.80	0.00
ETTm2	P1	0.04	0.04	0.20	0.03
	P2	0.05	0.01	0.34	0.23
	P3	0.07	0.03	0.38	0.06
	P4	0.11	0.13	0.79	0.09
Weather	P1	0.02	0.03	0.59	0.08
	P2	0.52	0.64	0.32	3.12
	P3	0.11	0.07	0.19	0.19
	P4	0.35	0.27	0.31	0.10

DLinear, where the two arms are close enough that the ratio turns on which blocks the seed drew. That is a property of the near-tie, not of the mechanism, and the same pattern does not appear where the gain is larger.

B PER-BACKBONE AND PER-PROTOCOL BREAKDOWN

Table 7 aggregates over all five datasets, four horizons and three seeds, so each entry is a broad average rather than a single cell. CMoS is positive under every protocol and largest under tail missingness ($+3.8\%$). CycleNet carries the only negative entry (P1, -0.1%), and its per-dataset gains are likewise near zero or slightly negative on ETTm1 and Weather. iTransformer is positive everywhere, with its gains concentrated under block and tail missingness, which is where the injected pathway has the most structure to convey. DLinear stays between $+0.1\%$ and $+0.2\%$ at every protocol, and its per-dataset gains range from $+0.0\%$ to $+0.5\%$.

Table 7: MSE gain (%) of MAFF per backbone and missingness protocol, aggregated over all five datasets, horizons, and seeds (summed-MSE ratio). Blue marks gains beyond 0.5% and gray marks differences within the $\pm 0.5\%$ error band (no entry exceeds 5%, so bold is not used).

Backbone	P1 (random)	P2 (block)	P3 (group)	P4 (tail)
CMoS	+2.3	+1.6	+3.2	+3.8
CycleNet	-0.1	+0.5	+0.5	+0.8
iTransformer	+0.2	+1.1	+0.4	+0.8
DLinear	+0.1	+0.2	+0.1	+0.2

Read across the rows, the protocols separate more sharply than the backbones. CMoS is largest under tail and correlated-group missingness ($+3.8\%$ and $+3.2\%$) and smallest under block missingness ($+1.6\%$), which is the ordering discussed in Section 5. CycleNet's single negative entry is P1; iTransformer's largest is P2, the protocol where its per-variate injection has the most structure to convey; DLinear moves by at most $+0.2\%$ anywhere in the table.

C GATE THRESHOLD AND FILL STRATEGY

Figure 7 reports the two remaining design choices. Varying the gate threshold τ leaves the results unchanged: the projection downstream absorbs whatever the gate attenuates, so τ is not a hyperparameter worth tuning, and the design needs no tuning of it. Replacing linear interpolation with two adversarial fillers matters more, and the outcome is the one the mechanism predicts: under structural missingness both still beat the mask-blind baseline (P3: $+11.7\%$ and $+8.7\%$; P4: $+12.2\%$ and $+10.8\%$); zero-filling matches the linear-interpolation reference on P3 ($+11.7\%$ against $+11.4\%$) while repeat-last trails it, and under scattered missingness (P1) both fall below the baseline (-5.0% and -5.6%), because the repair restores statistics rather than inventing information. MAFF repairs the statistics and injects the missingness structure; it does not rescue an input that is actively misleading, and no method that reads the input can.

D COMPONENT ABLATION

Table 8 separates the two active components on ETTh1. On CMoS, normalization alone recovers $+2.8\%$ and injection alone $+4.2\%$, while the two together reach $+6.9\%$, close to their sum. Which one matters more depends on the protocol. Injection dominates under scattered missingness ($+5.6\%$ against -0.5%, where runs of 3.5 steps leave the statistics only mildly distorted), and normalization dominates under block missingness ($+3.0\%$ against $+1.1\%$, where most of the window is gone and the statistics are the dominant defect). CycleNet shows the same ordering at roughly a third of the magnitude, and its normalization-only arm under P1 is the one entry in the table that is both negative and outside the $\pm 0.5\%$ band.

One reading matters here: the baseline rows show that the two backbones start from different error levels (0.55 to 0.87 for CMoS, 0.48 to 0.80 for CycleNet), so the percentages measure each back-

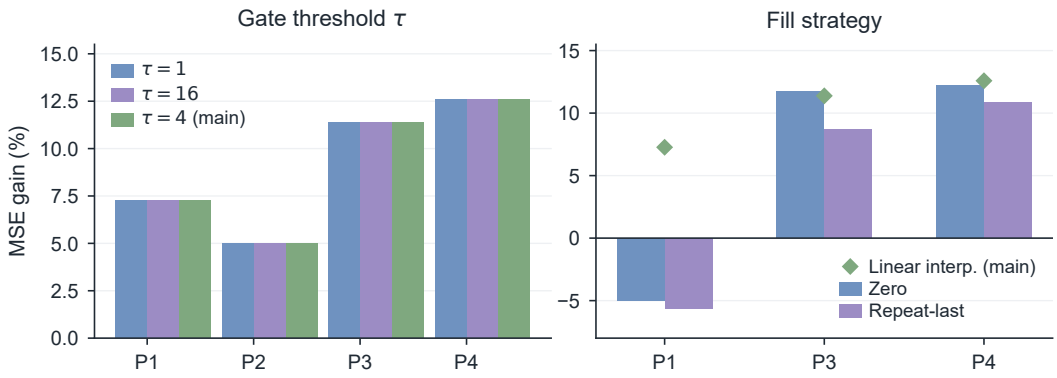

Figure 7: Gate threshold (left) and fill strategy (right) on CMoS, ETTh1, h=96. The three τ settings produce bars of the same height, so the left panel reads as a single bar per group. The fill panel shows the three protocols for which both adversarial fillers were run.

Table 8: Component ablation on ETTh1 (horizons 96 and 336, 3 mask seeds). Baseline rows give the mean test MSE of the mask-blind arm; the other rows give the percentage gain over that baseline. *Norm only* forwards the mask for statistics without injecting features; *injection only* injects features while the backbone keeps its filler-contaminated normalization.

Backbone	Arm	P1	P2	P3	P4
CMoS	Baseline MSE	0.5455	0.8737	0.5963	0.6479
	Norm only	−0.5	+3.0	+2.6	+5.6
	Injection only	+5.6	+1.1	+6.3	+5.3
	Full MAFF	+5.1	+4.0	+8.9	+10.4
CycleNet	Baseline MSE	0.4787	0.7953	0.5047	0.5302
	Norm only	−0.7	+1.2	+1.7	+1.5
	Injection only	+0.9	+1.1	+1.3	+0.9
	Full MAFF	+0.2	+2.1	+3.0	+2.3

bone's own improvement and are not, and are not meant to be, a comparison between the two halves of the table.

E Missingness-strength sweep

Table 9 sweeps the missingness ratio from 0.1 to 0.5; the main experiments use 0.3. Tail missingness gains monotonically with strength, while block missingness shrinks because the blocks overlap as the budget grows and much of the window is already empty; random missingness is comparatively flat, and correlated-group missingness is flat across the sweep at +8.9% (Figure 6, right). The practical reading is that the advantage of MAFF scales with the recoverable structure in the window, not with raw missingness.

The last column is the one to read if the goal is comparability with the main experiments: it gives each cell relative to the ratio-0.3 reference. Tail missingness gains most from a larger budget (+5.1 points at ratio 0.5) and block missingness loses most (−1.5 points).

F Prediction-level behavior

The three figures below come from the single CMoS run on ETTh1 whose aggregate numbers appear in Table 2: look-back 96, horizon 96, mask seed 2024, all four protocols, 2,785 test windows each. Figure 8 shows one forecast per protocol, each taken at the window whose per-window gain is closest to that protocol's median, so the panels are representative rather than selected. Figure 9 breaks the error down by forecast step, and Figure 10 gives the per-window picture.

Table 9: Missingness-strength sweep on CMoS, ETTh1 (horizons 96 and 336, 3 mask seeds). The last column compares against the ratio-0.3 reference from the main experiments. Correlated-group missingness (P3) is omitted.

Protocol	Ratio	Baseline MSE	MAFF MSE	Gain (%)	Δ vs. r=0.3
P1	0.1	0.5417	0.5122	+5.4	+0.31
P1	0.2	0.5431	0.5142	+5.3	+0.18
P1	0.4	0.5497	0.5227	+4.9	−0.23
P1	0.5	0.5566	0.5305	+4.7	−0.44
P2	0.1	0.6725	0.6337	+5.8	+1.73
P2	0.2	0.7747	0.7440	+4.0	−0.08
P2	0.4	0.9629	0.9312	+3.3	−0.75
P2	0.5	1.0396	1.0132	+2.5	−1.50
P4	0.1	0.5646	0.5284	+6.4	−3.97
P4	0.2	0.5971	0.5519	+7.6	−2.82
P4	0.4	0.6979	0.6094	+12.7	+2.30
P4	0.5	0.7608	0.6433	+15.4	+5.07

That per-window picture is more nuanced than the aggregates, and we report it because the aggregates would otherwise hide it. The fraction of windows on which the gain is negative is 15.3% under P1, 36.2% under P2, 10.5% under P3 and 12.7% under P4, so under block missingness more than a third of the individual windows regress even though the protocol-level average is positive. The gain is larger in the windows where the baseline is least accurate: the correlation between the log of the baseline's per-window error and the per-window gain is 0.13, 0.20, 0.22 and 0.31 for P1–P4, a consistent association that nonetheless leaves most of the per-window variation unexplained, which is why the paper reports cell-level aggregates rather than per-window averages.

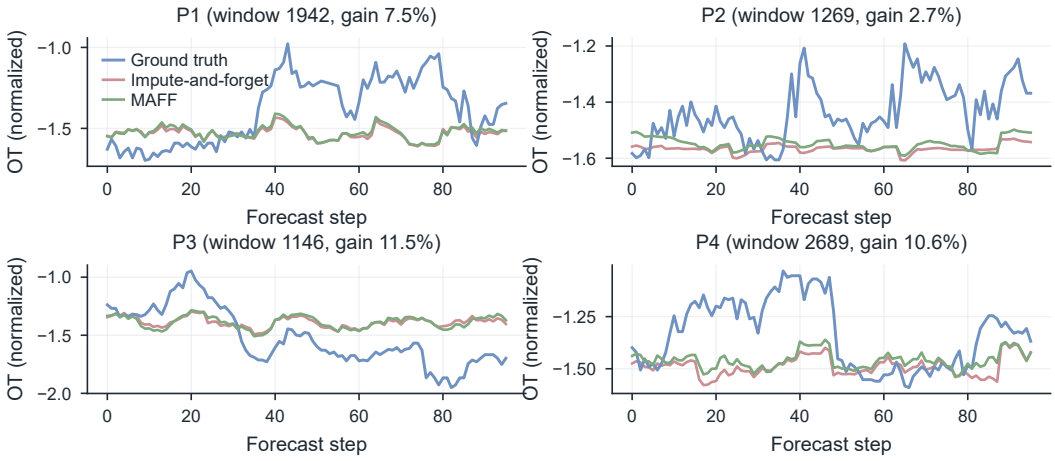

Figure 8: Forecast examples on CMoS, ETTh1, one per protocol, each at the window whose per-window gain is closest to that protocol's median. Blue is the ground truth, rose the impute-and-forget baseline, green MAFF. The two forecasts are close in absolute terms, consistent with the aggregate gains in Table 2.

The per-window view is the most granular data in the paper, and its scatter is the reason the paper reports aggregate gains: these per-window numbers are far wider than the cell-level ones and should not be compared with them directly.

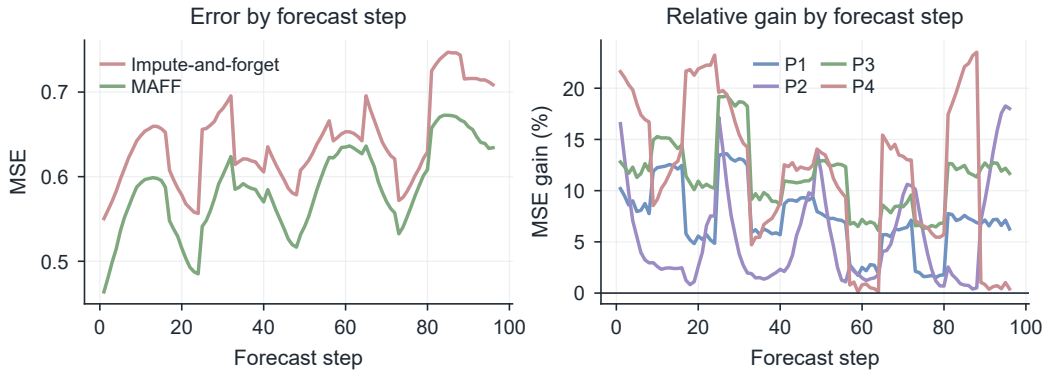

Figure 9: Error by forecast step, averaged over the four protocols (left) and the relative gain per protocol (right). The gain is not confined to the beginning of the horizon: MAFF is ahead at every one of the 96 steps under all four protocols, with the margin smallest under P2 and P4.

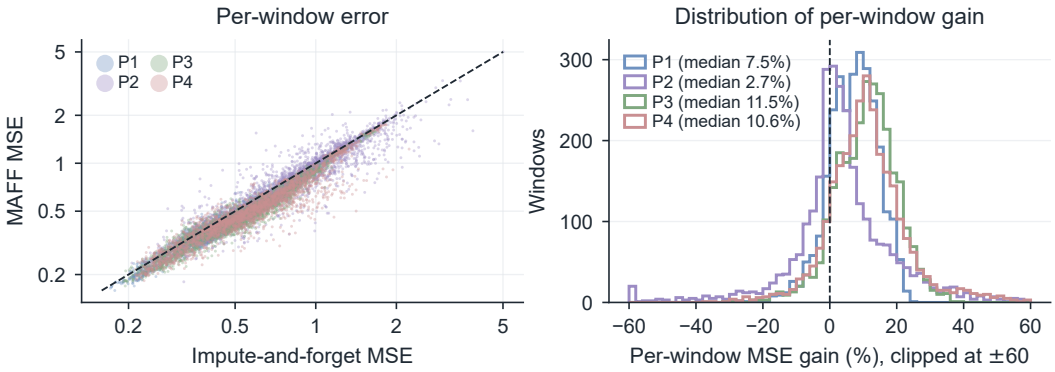

Figure 10: Per-window error on CMoS, ETTh1. Left: each point is one of the 2,785 test windows, on log axes, against the diagonal of equal error; points above the diagonal are windows on which MAFF is more accurate. Right: distribution of the per-window gain, clipped at $\pm 60\%$; the medians are 7.5%, 2.7%, 11.5% and 10.6% for P1–P4. The distributions sit above zero with a left tail of windows on which the baseline is already accurate, which is where the gain is smallest.

G Training protocol

Every run uses the same recipe: Adam (Kingma & Ba, 2015) with a learning rate of 10^{-4} halved every epoch and batch size 32, at most 10 epochs with early stopping on a patience of 3, $d_{\text{model}}{=}512$, $d_{\text{ff}}{=}2048$, two encoder layers and dropout 0.05, a look-back of 96 and horizons of 96, 192, 336 and 720. Each configuration is repeated over three mask seeds (2024, 2025 and 2026) with the training seed fixed at 2021. Constructing the wrapper's modules advances the global random stream, so the two arms of a pair also differ in their dropout draws; the clean-data control bounds the size of that difference. Gains are computed from summed MSE as $(\text{off} - \text{full})/\text{off}$, and the same definition is used for MAE. The projection the wrapper adds is identity-initialized with a scale that starts near zero, which is what keeps the *none* differences within $\pm 0.9\%$ for three of the four backbones once they are pooled over horizons.

H Extended related work

Imputation. Classical methods fill gaps by chained equations (van Buuren & Groothuis-Oudshoorn, 2011) or by multi-directional recurrence over irregular streams (Yoon et al., 2019). Deep imputation has since moved to adversarial training (Yoon et al., 2018), conditional diffusion (Tashiro et al., 2021), self-attention over the observed

positions (Du et al., 2023); the survey of Wang et al. (2025) covers the space in more detail. These methods are all trained to produce an accurate completed series, and they are evaluated on the accuracy of the filled values. What matters for us is narrower: the filled values only have to stop contaminating the statistics that normalize the window, which is why the repair here can be parameter-free and why it composes with whichever filler a practitioner already uses.

Backbones. The four architectures we wrap span the families that the long-horizon forecasting literature has converged on. Autoformer (Wu et al., 2021) and TimesNet (Wu et al., 2023) model series structure in the frequency and period dimensions; TimeMixer (Wang et al., 2024) mixes across scales after decomposition; SCINet (Liu et al., 2022a) and ModernTCN (Luo & Wang, 2024) treat the series as a signal to be filtered; DLinear (Zeng et al., 2023) is all-MLP; iTransformer (Liu et al., 2024) inverts the attention axis; CycleNet (Lin et al., 2024) memorizes periodic residuals; and CMoS (Si et al., 2025) builds correlations between temporal chunks. Most of these architectures were designed and tuned for complete inputs, and none of them exposes a place where a missingness signal is expected. That is the gap MAFF targets, and it is why the injection point has to be chosen per architecture rather than fixed once.

Normalization and missingness in forecasting. Instance normalization is what makes a forecaster robust to level and scale shifts, and it has been extended in several directions: reversible normalization of the look-back window (Kim et al., 2022), slice-level statistics (Liu et al., 2023), a frequency-adaptive treatment of non-stationarity (Ye et al., 2024), and distribution-shift correction applied to the predicted output (Fan et al., 2023). Each of these computes its statistics from the window as given, which is the assumption that missingness breaks.

Forecasting from incomplete series has recently begun to be studied directly. S4M (Peng et al., 2025) pairs a state-space backbone with missing-value forecasting, GinAR (Yu et al., 2024) targets variable-level absence, and CRIB (Yang et al., 2025) argues against imputation altogether, reporting that an unsupervised imputation step can corrupt the data distribution and degrade prediction, and proposing instead to predict from the partially observed series with a consistency-regularized information bottleneck. Our claim is narrower and our intervention is different. We keep the impute-and-forecast pipeline that practice already uses and identify the specific route by which fillers do damage: they enter the instance-normalization statistics, and that entry can be repaired without touching the backbone or adding capacity. CRIB's diagnosis is about the accuracy of the imputed values, ours is about the statistics computed from them, and the two are complementary. A better filler helps both; the repair here helps even when the filler is as crude as zero-filling or repeat-last, which is what the fill-strategy panel of Figure 7 shows.

I MAE RESULTS

Table 10 repeats the breakdown of Table 7 for absolute error, and Tables 11–14 give the absolute MAE of both arms for all four backbones. The ordering across backbones is the same, and on CMoS the gain is largest under tail and correlated-group missingness and smallest under block missingness, so the effect is not an artifact of a single loss geometry. The per-dataset MAE gains on CMoS are $+3.0\%$ (ETTh1), $+1.6\%$ (ETTm1), $+0.6\%$ (ETTm2), $+0.6\%$ (Weather) and $+0.5\%$ (ETTh2).

Table 10: MAE gain (%) of MAFF per backbone and missingness protocol, aggregated over all five datasets, horizons, and seeds (summed-MAE ratio), on the same cells as Table 7. Blue marks gains beyond 0.5% and gray marks differences within the $\pm 0.5\%$ error band.

Backbone	P1 (random)	P2 (block)	P3 (group)	P4 (tail)
CMoS	$+1.5$	$+0.8$	$+1.7$	$+1.7$
CycleNet	-0.1	$+0.1$	$+0.2$	$+0.4$
iTransformer	-0.0	$+0.5$	$+0.2$	$+0.5$
DLinear	$+0.1$	$+0.3$	$+0.1$	$+0.1$

Table 11 is the absolute counterpart of Table 2. The off/full gap follows the same shape: it is widest on ETTh1 and under block missingness, narrows with the horizon in the same way, and is the only CMoS table in which a cell turns slightly negative, on Weather under tail missingness (-0.1%).

Table 11: Absolute test MAE on the CMoS backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.474 / 0.455	0.499 / 0.487	0.516 / 0.505	0.539 / 0.531
	P2	0.608 / 0.594	0.617 / 0.605	0.625 / 0.616	0.644 / 0.638
	P3	0.497 / 0.470	0.519 / 0.498	0.533 / 0.515	0.557 / 0.542
	P4	0.522 / 0.491	0.542 / 0.520	0.553 / 0.532	0.573 / 0.555
ETTh2	P1	0.381 / 0.379	0.427 / 0.427	0.453 / 0.453	0.461 / 0.461
	P2	0.401 / 0.396	0.443 / 0.439	0.469 / 0.466	0.477 / 0.475
	P3	0.383 / 0.381	0.429 / 0.428	0.455 / 0.454	0.463 / 0.462
	P4	0.409 / 0.405	0.450 / 0.447	0.469 / 0.466	0.474 / 0.471
ETTm1	P1	0.417 / 0.402	0.427 / 0.418	0.446 / 0.435	0.475 / 0.470
	P2	0.526 / 0.519	0.537 / 0.536	0.551 / 0.549	0.575 / 0.577
	P3	0.424 / 0.409	0.434 / 0.425	0.452 / 0.444	0.481 / 0.480
	P4	0.462 / 0.450	0.472 / 0.459	0.484 / 0.474	0.510 / 0.504
ETTm2	P1	0.283 / 0.277	0.317 / 0.315	0.353 / 0.352	0.406 / 0.405
	P2	0.303 / 0.298	0.338 / 0.337	0.373 / 0.373	0.426 / 0.426
	P3	0.284 / 0.278	0.318 / 0.316	0.354 / 0.354	0.407 / 0.407
	P4	0.309 / 0.307	0.339 / 0.338	0.369 / 0.368	0.419 / 0.418
Weather	P1	0.248 / 0.243	0.284 / 0.279	0.315 / 0.314	0.360 / 0.358
	P2	0.249 / 0.249	0.288 / 0.286	0.330 / 0.330	0.382 / 0.381
	P3	0.248 / 0.243	0.283 / 0.279	0.315 / 0.314	0.360 / 0.359
	P4	0.273 / 0.272	0.307 / 0.308	0.335 / 0.335	0.373 / 0.373

Table 12: Absolute test MAE on the CycleNet backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.437 / 0.434	0.459 / 0.460	0.478 / 0.479	0.497 / 0.498
	P2	0.571 / 0.564	0.587 / 0.584	0.598 / 0.595	0.615 / 0.612
	P3	0.452 / 0.441	0.474 / 0.467	0.491 / 0.485	0.510 / 0.505
	P4	0.468 / 0.459	0.486 / 0.481	0.498 / 0.494	0.515 / 0.511
ETTh2	P1	0.362 / 0.361	0.411 / 0.414	0.443 / 0.443	0.454 / 0.454
	P2	0.387 / 0.386	0.434 / 0.433	0.464 / 0.462	0.473 / 0.471
	P3	0.365 / 0.365	0.414 / 0.417	0.446 / 0.445	0.456 / 0.456
	P4	0.404 / 0.400	0.444 / 0.444	0.464 / 0.461	0.470 / 0.467
ETTm1	P1	0.380 / 0.380	0.396 / 0.397	0.415 / 0.416	0.448 / 0.449
	P2	0.501 / 0.502	0.518 / 0.521	0.533 / 0.538	0.556 / 0.562
	P3	0.387 / 0.388	0.404 / 0.406	0.423 / 0.426	0.455 / 0.458
	P4	0.430 / 0.430	0.439 / 0.439	0.450 / 0.450	0.479 / 0.480
ETTm2	P1	0.272 / 0.272	0.311 / 0.311	0.348 / 0.348	0.403 / 0.403
	P2	0.294 / 0.294	0.333 / 0.332	0.369 / 0.369	0.423 / 0.422
	P3	0.273 / 0.273	0.312 / 0.312	0.350 / 0.349	0.405 / 0.404
	P4	0.307 / 0.306	0.337 / 0.336	0.368 / 0.367	0.417 / 0.416
Weather	P1	0.239 / 0.239	0.273 / 0.273	0.308 / 0.308	0.353 / 0.354
	P2	0.249 / 0.250	0.285 / 0.286	0.327 / 0.328	0.379 / 0.380
	P3	0.238 / 0.238	0.273 / 0.273	0.308 / 0.308	0.354 / 0.354
	P4	0.272 / 0.273	0.304 / 0.304	0.331 / 0.331	0.370 / 0.370

Table 12 gives CycleNet's absolute MAE. Its off runs sit about 0.03 below CMoS's, and the two columns are close everywhere except ETTm1 under block missingness, where the baseline is worse and the gain is correspondingly larger.

Table 13: Absolute test MAE on the iTransformer backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.409 / 0.412	0.437 / 0.441	0.467 / 0.464	0.498 / 0.497
	P2	0.510 / 0.505	0.535 / 0.530	0.556 / 0.550	0.585 / 0.576
	P3	0.417 / 0.414	0.448 / 0.446	0.474 / 0.461	0.499 / 0.504
	P4	0.438 / 0.444	0.471 / 0.464	0.485 / 0.473	0.498 / 0.506
ETTh2	P1	0.347 / 0.349	0.400 / 0.398	0.440 / 0.433	0.449 / 0.453
	P2	0.371 / 0.372	0.425 / 0.422	0.457 / 0.455	0.465 / 0.462
	P3	0.346 / 0.349	0.401 / 0.398	0.437 / 0.436	0.461 / 0.450
	P4	0.391 / 0.393	0.436 / 0.430	0.458 / 0.451	0.471 / 0.465
ETTm1	P1	0.376 / 0.375	0.401 / 0.405	0.434 / 0.428	0.462 / 0.465
	P2	0.474 / 0.471	0.496 / 0.491	0.516 / 0.511	0.543 / 0.539
	P3	0.379 / 0.379	0.404 / 0.407	0.434 / 0.436	0.470 / 0.472
	P4	0.419 / 0.419	0.435 / 0.436	0.456 / 0.454	0.486 / 0.487
ETTm2	P1	0.266 / 0.267	0.309 / 0.314	0.354 / 0.348	0.410 / 0.407
	P2	0.285 / 0.285	0.329 / 0.328	0.365 / 0.367	0.420 / 0.422
	P3	0.267 / 0.268	0.310 / 0.315	0.351 / 0.349	0.409 / 0.407
	P4	0.308 / 0.304	0.337 / 0.337	0.371 / 0.368	0.421 / 0.421
Weather	P1	0.215 / 0.215	0.256 / 0.258	0.299 / 0.300	0.350 / 0.351
	P2	0.248 / 0.248	0.285 / 0.285	0.326 / 0.327	0.379 / 0.381
	P3	0.214 / 0.213	0.255 / 0.256	0.298 / 0.298	0.349 / 0.350
	P4	0.258 / 0.256	0.293 / 0.290	0.324 / 0.323	0.366 / 0.365

Table 13 gives iTransformer's absolute MAE, whose off runs have the lowest error level of the four backbones. The table shows the same concentration as its MSE counterpart: the widest gaps are on block and tail missingness, and the largest regression is on Weather under random missingness (-0.4%).

The four tables above give the same absolute MAE breakdown. The pattern tracks the MSE tables. CMoS reaches $+4.3\%$ on ETTh1 under tail missingness and turns slightly negative only on Weather under P4 (-0.1%). CycleNet and DLinear stay within $\pm 0.5\%$ on all but a few cells, the largest regression being -0.7% for CycleNet on ETTm1 under block missingness. iTransformer is again strongest on block and tail missingness.

J Cost and mask statistics

Table 15 gives the numbers behind Figure 3, and Table 16 the numbers behind Figure 4. Together they cover the two quantities that the results section reports without a per-cell table: the training-time ratio of the wrapper and the run lengths the gate sees.

Table 15: Per-epoch training-time ratio between the full and off runs of the same configuration, over 240 pairs per backbone. Median, interquartile range and full range are given; the same data is drawn in Figure 3.

Backbone	pairs	median	IQR	range
CMoS	240	0.993	0.814–1.033	0.523–1.547
CycleNet	240	0.998	0.792–1.154	0.466–1.567
iTransformer	240	1.339	1.166–1.431	0.464–2.123
DLinear	240	0.765	0.674–0.997	0.504–1.621

Table 14: Absolute test MAE on the DLinear backbone: mask-blind (off) / mask-aware (full), per dataset, protocol, and horizon (3 mask seeds).

Dataset	Prot.	h96	h192	h336	h720
ETTh1	P1	0.415 / 0.415	0.444 / 0.444	0.469 / 0.473	0.514 / 0.512
	P2	0.546 / 0.545	0.566 / 0.566	0.586 / 0.584	0.619 / 0.619
	P3	0.426 / 0.425	0.454 / 0.454	0.478 / 0.483	0.523 / 0.521
	P4	0.450 / 0.451	0.473 / 0.472	0.492 / 0.496	0.533 / 0.531
ETTh2	P1	0.397 / 0.404	0.481 / 0.478	0.543 / 0.545	0.662 / 0.661
	P2	0.446 / 0.450	0.524 / 0.516	0.582 / 0.585	0.698 / 0.699
	P3	0.399 / 0.406	0.482 / 0.480	0.546 / 0.547	0.664 / 0.663
	P4	0.472 / 0.471	0.545 / 0.529	0.579 / 0.595	0.706 / 0.699
ETTm1	P1	0.377 / 0.375	0.393 / 0.393	0.417 / 0.415	0.452 / 0.454
	P2	0.503 / 0.502	0.520 / 0.518	0.537 / 0.536	0.562 / 0.562
	P3	0.383 / 0.381	0.400 / 0.400	0.424 / 0.422	0.459 / 0.461
	P4	0.430 / 0.429	0.438 / 0.438	0.454 / 0.451	0.486 / 0.487
ETTm2	P1	0.295 / 0.296	0.363 / 0.362	0.430 / 0.429	0.524 / 0.517
	P2	0.325 / 0.326	0.391 / 0.389	0.452 / 0.456	0.548 / 0.543
	P3	0.295 / 0.297	0.364 / 0.362	0.430 / 0.429	0.525 / 0.518
	P4	0.349 / 0.350	0.404 / 0.403	0.464 / 0.464	0.549 / 0.545
Weather	P1	0.258 / 0.258	0.301 / 0.296	0.332 / 0.332	0.383 / 0.382
	P2	0.333 / 0.331	0.370 / 0.361	0.403 / 0.398	0.443 / 0.443
	P3	0.257 / 0.257	0.299 / 0.296	0.331 / 0.333	0.383 / 0.381
	P4	0.311 / 0.308	0.346 / 0.345	0.370 / 0.375	0.408 / 0.407

Table 16: Mask-pathway statistics: the mean longest contiguous missing run, averaged over test windows and channels, and the gate value $g = \sigma((b_c - \tau)/\tau)$ with $\tau = 4$ that it implies. Masks regenerated with the mask generator and seeds used in the experiments (ratio 0.3, look-back 96, three mask seeds).

Dataset	Prot.	mean longest run	gate g
ETTh1	P1	3.49	0.468
	P2	53.86	1.000
	P3	16.29	0.956
	P4	28.41	0.988
ETTh2	P1	3.49	0.468
	P2	53.86	1.000
	P3	16.29	0.956
	P4	28.41	0.988
ETTm1	P1	3.49	0.468
	P2	53.83	1.000
	P3	16.29	0.956
	P4	28.53	0.988
ETTm2	P1	3.49	0.468
	P2	53.83	1.000
	P3	16.29	0.956
	P4	28.53	0.988
Weather	P1	3.48	0.468
	P2	92.33	1.000
	P3	18.10	0.971
	P4	28.52	0.988