Vol. 1, No. 1 (2026) | pp. 68-81 | DOI: 10.5281/zenodo.23292852

Research Article

Missingness-Aware Linear Forecasting for Long-Term Multivariate Time Series with Incomplete Inputs

Rohan Deshmukh[1*], Dr. Arvind Kulkarni[2], Dr. Mohd Faizal Ibrahim[3]

[1] *School of Computing Science and Engineering (Research Scholar), VIT Bhopal University, Kothri Kalan, Sehore, Madhya Pradesh, India*
Email: rohan.deshmukh2024@vitbhopal.ac.in

[2] *School of Computing Science and Engineering, VIT Bhopal University, Kothri Kalan, Sehore, Madhya Pradesh, India*
Email: arvind.kulkarni@vitbhopal.ac.in

[3] *Faculty of Computing and Informatics, Multimedia University (MMU), Cyberjaya, Selangor, Malaysia*
Email: faizal.ibrahim@mmu.edu.my

Received: 18 July 2026 | Revised: 26 August 2026 | Accepted: 21 September 2026 | Published: 26 September 2026
DOI: 10.5281/zenodo.23292852

How to cite this article: Rohan Deshmukh, Arvind Kulkarni, Mohd Faizal Ibrahim (2026). Missingness-Aware Linear Forecasting for Long-Term Multivariate Time Series with Incomplete Inputs. *International Journal of Advanced Computing and Machine Intelligence (IJACMI)*, 1(1), pp. 68-81. DOI: 10.5281/zenodo.23292852

Abstract

Linear models are strong and inexpensive baselines for long-term time-series forecasting, but they are usually evaluated on complete input windows, whereas deployed sensor streams contain isolated dropouts and multi-hour outages. This paper examines how standard linear forecasters degrade when their inputs are incomplete and proposes MALin, a missingness-aware linear forecaster that adds three lightweight elements to a RevIN-linear model: instance normalisation computed from observed entries only, a linear branch driven by the missingness mask, and training on windows corrupted with random point and block missingness. MALin and seven baselines (naive repeat, ridge regression, DLinear with zero filling and with interpolation, NLinear, RLinear, and RLinear with missingness-augmented training) were evaluated on the ETTh1, ETTh2, and ETTm1 datasets at horizons of 96 to 720 steps, under point and block missingness at rates of 10–70%, with identical fixed test masks for all models. Interpolation made point missingness largely harmless, but 50% block missingness still increased the MSE of interpolated DLinear and RLinear by 30% and 33% on average. MALin reduced the error under 30–70% block missingness by 3.8% on average relative to RLinear and by up to 9.5% on ETTm1, without a systematic loss on complete inputs. Diebold–Mariano tests with Holm correction found MALin significantly better than RLinear in 24 of 48 block-missingness settings and significantly worse in none of the 108 settings tested. The gains were concentrated on ETTh1 and ETTm1, while MALin was slightly but not significantly worse on ETTh2. An ablation showed that missingness-augmented training and observed-only normalisation account for most of the improvement, whereas the mask branch helps only when gaps are zero-filled.

Keywords: *time-series forecasting; missing data; linear models; robustness; instance normalisation; data augmentation*

1.0 Introduction

Long-term forecasting of multivariate sensor data supports planning in power systems, transport, and industrial monitoring. Benchmark studies of the problem almost always assume that the lookback window is complete. Deployed sensors do not cooperate with this assumption: individual readings are dropped by unreliable links, and whole channels go silent for hours during maintenance or communication outages. A forecaster that was trained on complete windows must then be fed something, and in practice the gaps are usually zero-filled, carried forward, or interpolated before the model sees them.

Over the last few years, the long-term forecasting literature has shown that small linear models are surprisingly hard to beat on the standard benchmarks. Zeng et al. [2] showed that a single linear layer over the lookback window (with or without a trend–seasonal decomposition) matched or outperformed several Transformer forecasters, and Li et al. [3] showed that reversible instance normalisation [5] combined with one linear map (RLinear) is a strong baseline across many datasets. These models are cheap to train, fast at inference, and easy to deploy on modest hardware, which is exactly where incomplete data are most common.

Work on forecasting with missing inputs has taken a different route. Recent methods add graph message passing [12], state-space backbones with prototype memories [13], gated cross-variate fusion with joint imputation [14], information-bottleneck objectives with attention [15], or dual observation/value streams [16]. In the evaluations reported by these papers, linear models such as DLinear enter only as mask-blind baselines, fed with zero-filled or separately imputed inputs. What has not been checked is whether the linear model itself can be made aware of missingness with very little extra machinery, and how far that closes the gap.

© 2026 The Author(s). Published by WhiteField Academic Press

This paper takes up that question. We propose MALin, a missingness-aware linear forecaster with three simple ingredients: (i) instance normalisation whose statistics are computed from observed entries only, (ii) a second linear map that takes the binary missingness mask as input and corrects the value branch, and (iii) training on windows that are randomly corrupted with point-wise and block-wise missingness. The model adds one L×H weight matrix to RLinear and has no attention, recurrence, or imputation network.

We evaluate MALin and seven baselines on three ETT datasets (ETTh1, ETTh2, ETTm1) [1] at four horizons, under point (MCAR) and block (outage-like) missingness at rates from 10% to 70%, using identical fixed test masks for every model. The study is designed to answer three questions:

- RQ1: How much do standard linear forecasters degrade under point and block missingness when gaps are filled by zeros or linear interpolation?
- RQ2: Does a missingness-aware linear model reduce this degradation, and at what cost on complete inputs?
- RQ3: Which of the three ingredients is responsible for the change?

The contributions are modest and, we believe, practically useful. First, we provide a controlled robustness evaluation of linear long-term forecasters under two missingness patterns, with seed repetitions and Diebold–Mariano tests on per-window losses. Second, we describe a lightweight missingness-aware linear model and isolate the effect of each of its components through a one-at-a-time ablation. Third, we report the outcome as observed: the benefits concentrate in block-wise missingness and are dataset-dependent, point-wise missingness is already handled well by interpolation, and the ablation shows that the training procedure, rather than the mask branch, carries most of the gain. All code, configurations, raw metrics, and per-window losses are released.

2.0 Related Work

2.1 Long-term time-series forecasting

The ETT benchmark was introduced with Informer [1], which targeted the quadratic cost of attention on long sequences. A series of deep forecasters followed, including TimesNet [7], which reshapes series into 2D period-based tensors, PatchTST [4], which splits each channel into patches and processes channels independently, and iTransformer [6], which treats each variate as a token. Zeng et al. [2] questioned whether attention was needed at all: their DLinear and NLinear models, which apply a single linear layer per window after a moving-average decomposition or last-value subtraction, were competitive with the Transformers of the time. Li et al. [3] analysed linear mapping further and found that reversible instance normalisation (RevIN) [5] together with one linear layer (RLinear) is a strong reference model. Channel independence and instance normalisation, both used by PatchTST and RLinear, are now standard design choices.

2.2 Learning with missing inputs

Rubin's taxonomy [17] distinguishes data missing completely at random (MCAR), at random, and not at random. For supervised prediction rather than parameter estimation, Josse et al. [19] showed that imputing with a constant before learning is consistent when the missingness is not informative, and that approaches which let the learner see where values are missing can also handle informative missingness. Le Morvan et al. [18] argued that an impute-then-regress procedure can be Bayes optimal for almost any imputation function when the downstream learner is powerful enough, which shifts attention from perfect imputation to the capacity of the predictor that follows it. Both results rely on flexible learners. A linear forecaster is not flexible in this sense: it cannot tell an interpolated value from an observed one. This motivates giving the mask to the linear predictor as an explicit input.

2.3 Imputation and forecasting with missing values

Deep imputation methods for time series include GRU-D [8], which uses decay mechanisms driven by the mask and the time since the last observation, BRITS [9], which uses bidirectional recurrent dynamics, SAITS [10], which uses self-attention trained with masked reconstruction, and CSDI [11], a conditional diffusion model. These methods can be placed in front of any forecaster, but a two-stage pipeline trains the imputer without reference to the forecasting objective and adds a second model to deploy.

End-to-end forecasters for incomplete inputs have appeared more recently. BiTGraph [12] couples biased temporal convolutions with graph learning. S4M [13] processes values and masks as two streams inside an S4

© 2026 The Author(s). Published by WhiteField Academic Press

state-space model with a prototype bank. GinAR [20] handles entirely missing variables with interpolation attention and adaptive graph convolution. CoIFNet [14] concatenates normalised values, the mask, and timestamp embeddings and fuses them with gated cross-timestep and cross-variate modules while jointly imputing. CRIB [15] forecasts directly from partially observed series with an information-bottleneck objective and a consistency regulariser built from extra random masking. Timeflies [16] predicts whether future points will be observed in addition to their values. In the experiments reported by CoIFNet, CRIB, and Timeflies, DLinear appears as a baseline that does not receive the mask. The question of how a linear forecaster performs once it is given the mask, observed-only normalisation, and missingness-aware training has, to our knowledge, not been examined separately; this is the gap the present study addresses. We do not claim that MALin competes with these heavier models in absolute accuracy, and we do not compare against their published numbers, because their missingness protocols, lookback lengths, and splits differ from ours.

3.0 Research Gap and Objectives

The literature reviewed above leaves a practical gap. Linear forecasters are attractive for deployment because of their cost, yet their behaviour under realistic input missingness has only been reported as a side result, with the mask discarded. It is unclear (a) how much of their degradation is due to the filling strategy rather than the model, (b) whether a few lightweight changes recover most of the lost accuracy, and (c) what such changes cost when data are complete.

The objectives of this study are therefore:

(1) To measure the degradation of DLinear, NLinear, RLinear, and a ridge regression baseline under point and block missingness on three public datasets, with all models evaluated on the same fixed masks.

(2) To design a missingness-aware linear forecaster that adds as little as possible to RLinear, and to quantify its benefit and its cost on complete inputs.

(3) To attribute any improvement to specific components through a one-at-a-time ablation, separating the effect of missingness-aware training from that of the architectural changes.

4.0 Proposed Methodology

4.1 Problem formulation

Let $\mathbf{X} \in \mathbb{R}^{L \times C}$ be a lookback window of L time steps and C channels, and let $\mathbf{M} \in \{0,1\}^{L \times C}$ be its observation mask, with $M_{t,c} = 1$ when $X_{t,c}$ is observed. The task is to predict the next H steps $\mathbf{Y} \in \mathbb{R}^{H \times C}$ from the observed part of $\mathbf{X}$ and from $\mathbf{M}$. We assume that the training targets are fully observed and that missingness affects only the model input. This matches the deployment setting we target, in which a model trained on a cleaned historical archive is run on live data with gaps. Missing input entries are set to zero before they reach any model, so no model can use values it is not supposed to see.

4.2 MALin

MALin operates channel-independently with weights shared across channels, as in DLinear, NLinear, and RLinear [2,3]. For one channel, write $\mathbf{x} \in \mathbb{R}^{L}$ for its values and $\mathbf{m} \in \{0,1\}^{L}$ for its mask. The model has three components.

Observed-only instance normalisation. RevIN [5] normalises each window by its own mean and standard deviation and restores them after prediction. With missing data, statistics computed over a filled window are biased toward the fill values: zero filling pulls the mean toward the training mean and shrinks the variance, and interpolation shrinks the variance within gaps. MALin computes the statistics from observed entries only:

$$\mu = \frac{\sum_t m_t x_t}{\sum_t m_t}, \qquad \sigma = \sqrt{\frac{\sum_t m_t \,(x_t - \mu)^2}{\sum_t m_t} + \varepsilon}, \qquad z_t = \gamma \frac{x_t - \mu}{\sigma} + \beta, \tag{1}$$

where γ and β are learnable per-channel affine parameters and $\varepsilon = 10^{-5}$. If a channel has fewer than two observed values in the window, the standard deviation falls back to 1 (the training-set scale), and with no observed value the mean falls back to 0 (the training-set mean).

© 2026 The Author(s). Published by WhiteField Academic Press

Filling in normalised space. Missing entries of $\mathbf{z}$ are filled by linear interpolation between the nearest observed neighbours, with nearest-value extension at the window edges:

$$\tilde{\mathbf{z}} = Interp\left(\mathbf{z} \odot \mathbf{m}, \mathbf{m}\right). \tag{2}$$

Mask branch. A linear forecaster cannot distinguish an interpolated value from an observed one, so the forecast of a window with a long interpolated gap is computed as if the straight line inside the gap were real. MALin adds a second linear map from the missingness indicator $\mathbf{1} - \mathbf{m}$ to the horizon:

$$\hat{\mathbf{y}}_{\text{norm}} = \mathbf{W}_x \tilde{\mathbf{z}} + \mathbf{b} + \mathbf{W}_m \left(\mathbf{1} - \mathbf{m}\right), \qquad \hat{\mathbf{y}} = \sigma \frac{\hat{\mathbf{y}}_{\text{norm}} - \beta}{\gamma} + \mu, \tag{3}$$

with $\mathbf{W}_x, \mathbf{W}_m \in \mathbb{R}^{H \times L}$ and $\mathbf{b} \in \mathbb{R}^H$. Because the branch acts on $\mathbf{1} - \mathbf{m}$ and $\mathbf{W}_m$ is initialised at zero, MALin starts training as RLinear with observed-only normalisation and, on a complete window, its output depends only on $\mathbf{W}_x$. The branch lets the model learn position-dependent corrections, for example a systematic offset when the most recent hours of a channel are missing. MALin has $2LH + H + 2C$ parameters, compared with $LH + H + 2C$ for RLinear. Figure 1 summarises the model.

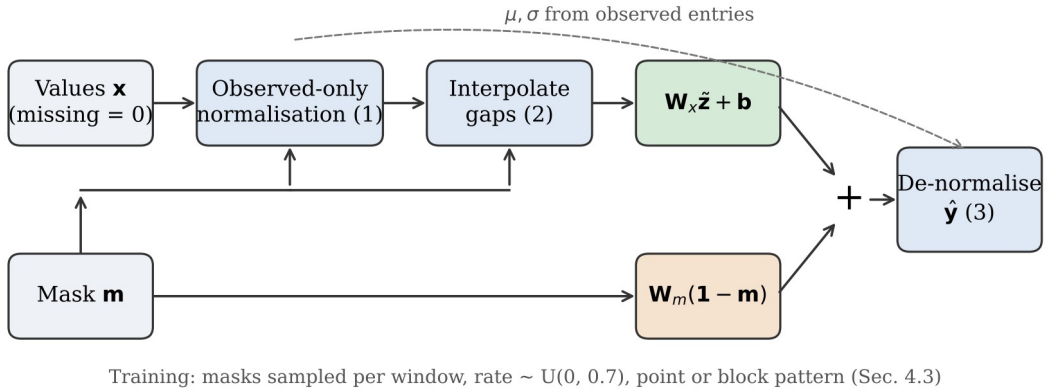


Figure 1. Structure of MALin. Values and mask enter observed-only normalisation and interpolation (value branch); the mask also drives a linear correction branch. The two branches are summed and de-normalised with the observed statistics.

4.3 Missingness-augmented training

The mask branch can only learn from masked inputs, so the training windows are corrupted on the fly. For every training window we draw a missing rate $r \sim \mathcal{U}\left(0, r_{\max}\right)$ with $r_{\max} = 0.7$ and choose the point pattern or the block pattern with probability 0.5 each:

- **Point:** each entry is dropped independently with probability r.
- **Block:** for each channel, $round\left(rL/\bar{\ell}\right)$ gaps with lengths drawn uniformly from $[\ell_{\min}, \ell_{\max}]$ are placed at random positions, where $\bar{\ell}$ is the mean gap length. We use 4–24 steps for hourly data and 16–96 steps for 15-minute data, which corresponds to outages of 4–24 hours in both cases.

A new mask is sampled for every window in every epoch, and the loss is the mean squared error against the complete target:

$$\mathcal{L} = \frac{1}{HC} \left\| \hat{\mathbf{Y}} - \mathbf{Y} \right\|_F^2. \tag{4}$$

One model is trained per dataset and horizon, and the same model is used for every test condition, including complete inputs. No information about the test missing rate or pattern is given to the model.

4.4 Inference

At inference time the observed values and their mask are passed through Eqs. (1)–(3). The computation is two matrix–vector products per channel plus the normalisation and the interpolation, so the cost is linear in L and H.

© 2026 The Author(s). Published by WhiteField Academic Press

5.0 Dataset and Preprocessing

5.1 Data

We use the ETT-small datasets released with Informer [1] in the official repository (github.com/zhouhaoyi/ETDataset) under the CC BY-ND 4.0 licence. Each file records seven variables from an electricity transformer: six load features (HUFL, HULL, MUFL, MULL, LUFL, LULL) and the oil temperature (OT). ETTh1 and ETTh2 are sampled hourly and ETTm1 and ETTm2 every 15 minutes; the two numbered versions come from two different transformers. We use ETTh1, ETTh2, and ETTm1; ETTm2 was also downloaded, but its runs could not be completed within the compute budget and it is excluded from all results. The files contain 17,420 (hourly) and 69,680 (15-minute) rows covering 1 July 2016 to 26 June 2018 (SHA-256 checksums are listed in the reproducibility package). We verified that the files contain no missing values and no duplicated timestamps, so all missingness in this study is synthetic and controlled.

5.2 Splits and normalisation

We follow the standard chronological protocol used by Informer, DLinear, and PatchTST [1,2,4]: the first 12 months are used for training, the next 4 months for validation, and the following 4 months for testing (8,640/2,880/2,880 rows for hourly data and 34,560/11,520/11,520 rows for 15-minute data). As in the reference implementation, the validation and test segments start L steps early so that the first window's lookback lies in the preceding split; targets never cross a split boundary. Each variable is standardised with the mean and standard deviation of the training split only. Windows are extracted with stride 1 and the full test set is evaluated (no batch dropping). Table 1 lists the resulting window counts.

Table 1. Dataset statistics after the standard chronological split (lookback L = 336). Window counts are the number of test samples per horizon; the last columns give the realised fraction of missing entries in the fixed test masks.

Dataset	Interval	Rows (file)	Train / Val / Test rows	Test windows (H=96 / 192 / 336 / 720)	Realised missing % (P30 / P70 / B30 / B70)
ETTh1	1 h	17,420	8,640 / 2,880 / 2,880	2,785 / 2,689 / 2,545 / 2,161	29.9 / 70.5 / 30.2 / 70.1
ETTh2	1 h	17,420	8,640 / 2,880 / 2,880	2,785 / 2,689 / 2,545 / 2,161	29.9 / 70.5 / 30.2 / 70.1
ETTm1	15 min	69,680	34,560 / 11,520 / 11,520	11,425 / 11,329 / 11,185 / 10,801	29.6 / 70.2 / 30.2 / 70.1

5.3 Missingness simulation for testing

Test masks are generated once per dataset, pattern, and rate over the whole test segment with a fixed seed, and every model and horizon is evaluated on the same masks. Windows inherit the segment-level mask, so overlapping windows share the same gaps, as they would during a real outage. We evaluate the point and block patterns described in Section 4.3 at rates of 10%, 30%, 50%, and 70%, plus the complete-input condition, giving nine test conditions. The realised missing fractions of the test masks were within 0.5 percentage points of the nominal rates (Table 1). The mask generator does not depend on the data values, so both patterns are MCAR in Rubin's sense [17]; the block pattern is MCAR at the level of gap placement but produces temporally contiguous gaps.

6.0 Baseline Models

All baselines receive the same zeroed inputs and masks as MALin. Unless stated otherwise, missing entries are filled by the same linear interpolation (Eq. 2, applied to the raw standardised values) before the model sees them, which is a strong and common default.

- **Naive (repeat):** repeats the last value of the interpolated lookback window.
- **Ridge:** channel-independent ridge regression [21] from the interpolated lookback to the horizon, solved in closed form on clean training windows; the penalty is chosen from $\{10^{-1}, \ldots, 10^{4}\}$ on the clean validation set.

© 2026 The Author(s). Published by WhiteField Academic Press

- **DLinear** and **DLinear-Zero** [2]: moving-average decomposition (kernel 25) and two linear layers, fed with interpolated or zero-filled inputs. The zero-filled version reflects the simplest practice in deployed pipelines.
- **NLinear** [2]: last-value subtraction and one linear layer.
- **RLinear** [3]: RevIN and one linear layer.
- **RLinear-MT:** RLinear trained with the same missingness-augmented procedure as MALin (Section 4.3) but with standard RevIN and no mask branch. This baseline isolates the effect of the training procedure from that of the architectural changes. DLinear, NLinear, and RLinear were re-implemented in PyTorch [22] following the official code of their authors (cure-lab/LTSF-Linear and plumprc/RTSF), using shared weights across channels. As a check on the implementation, our clean-input DLinear MSE on ETTh1 at $H = 96$ was 0.375 and that of the naive repeat baseline was 1.294; Zeng et al. [2] report 0.375 and 1.295 for the same setting.

7.0 Experimental Setup

All neural models use lookback $L = 336$, the setting used for linear models by Zeng et al. [2], and horizons $H \in \{96, 192, 336, 720\}$. They are trained with Adam [23], batch size 256, at most 8 epochs, and early stopping with patience 2 on the validation MSE. The learning rate is halved after every epoch, as in the LTSF-Linear code. The initial learning rate (0.005) was selected from {0.001, 0.005} on the ETTh1 validation set at $H = 96$ for DLinear, RLinear, and MALin, and 0.005 was best for all three; it was then used for every trainable model. Models trained on clean windows are selected on the clean validation set. Models trained with missingness augmentation are selected on a validation set corrupted with a fixed mixture of point and block masks at rates from 0 to 0.7, which matches their training distribution. The test set is never used for any choice.

All main comparisons use seed 0 for every model. To check stability, RLinear, RLinear-MT, and MALin were additionally trained with seeds 1 and 2 on the two hourly datasets. Naive and Ridge are deterministic. The ablation variants were run with seed 0 on ETTh1 and ETTh2. These limits on repetitions and on the ablation scope were set by the available CPU time. We report MSE and MAE on the standardised scale, averaged over all test windows, horizon steps, and channels, which is the convention of the long-term forecasting literature and makes our clean-input numbers directly comparable with published ones.

Statistical comparisons use the Diebold–Mariano (DM) test [24] on the per-window MSE differences between MALin and each competitor, using the seed-0 runs. Consecutive windows overlap, so their losses are strongly autocorrelated; we therefore estimate the long-run variance with a Newey–West (Bartlett) estimator [25] using $L + H$ lags, a conservative choice that covers the full overlap of input and target. P-values are adjusted with the Holm procedure [26] within each comparison family (108 tests per competitor: 3 datasets × 4 horizons × 9 conditions).

All experiments were run on a CPU-only virtual machine (2 vCPUs, 7 GB RAM, Linux 6.18) with Python 3.13, PyTorch 2.14.1 (CPU execution), NumPy 2.5, and pandas 3.0. Each run used one CPU thread; two runs were executed in parallel.

8.0 Results

8.1 Degradation of standard linear forecasters (RQ1)

Table 2 reports the test MSE averaged over the four horizons and Table 3 the corresponding MAE. Figure 2 plots the same averages against the missing rate. Three observations stand out.

First, the filling strategy matters more than the choice of linear model. Zero filling is the worst option by a wide margin: averaged over the three datasets, the MSE of DLinear-Zero rises by 177% at 70% point missingness and by 109% at 50% block missingness, relative to its own clean-input error. The same DLinear architecture fed with linearly interpolated inputs degrades by only 5.5% at 70% point missingness.

Second, point missingness is benign once gaps are interpolated. With isolated missing entries, the interpolated value is close to the true one, and every interpolation-based baseline stays within a few percent of its clean-input MSE up to 50% missingness (Table 2).

© 2026 The Author(s). Published by WhiteField Academic Press

Third, block missingness remains harmful even with interpolation. At 50% block missingness, the MSE of interpolated DLinear increases by 30.3% on average (range 6.4–53.1%) and that of RLinear by 33.3% (range 9.0–56.9%). At 70% the increase reaches 53.4% for DLinear and 58.9% for RLinear. A straight line across a gap of one or more days is a poor substitute for the daily load cycle, and a linear forecaster that weights all lookback positions equally cannot discount it.

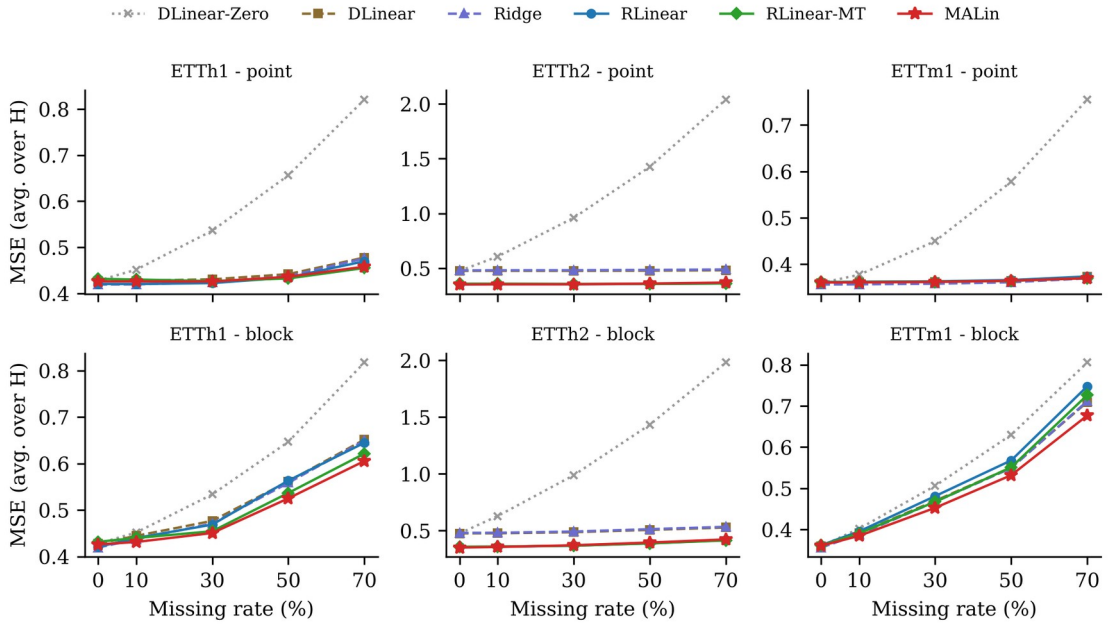


Figure 2. Test MSE (averaged over horizons) against the missing rate for point (top) and block (bottom) missingness. The leftmost point of each curve is the complete-input condition.

8.2 Effect of missingness awareness (RQ2)

Table 2. Test MSE averaged over the four horizons (96, 192, 336, 720). P = point missingness, B = block missingness, numbers = missing rate (%). Best value per dataset and condition in bold (Naive excluded). Lower is better.

Dataset	Model	clean	P10	P30	P50	P70	B10	B30	B50	B70
ETTh1	Naive	1.321	1.323	1.316	1.342	1.342	1.347	1.294	1.356	1.213
	Ridge	**0.419**	**0.420**	0.424	0.438	0.476	0.437	0.471	0.558	0.649
	DLinear-Zero	0.428	0.451	0.536	0.656	0.821	0.452	0.533	0.647	0.817
	DLinear	0.428	0.428	0.431	0.442	0.478	0.444	0.477	0.562	0.651
	NLinear	0.429	0.429	0.431	0.443	0.478	0.449	0.480	0.571	0.652
	RLinear	0.421	0.420	**0.422**	0.435	0.470	0.440	0.469	0.563	0.644
	RLinear-MT	0.432	0.430	0.427	**0.432**	**0.455**	0.440	0.454	0.537	0.621
	MALin	0.426	0.426	0.426	0.436	0.458	**0.431**	**0.450**	**0.525**	**0.605**
ETTh2	Naive	0.539	0.542	0.540	0.543	0.543	0.538	0.537	0.567	0.582
	Ridge	0.481	0.482	0.482	0.485	0.489	0.482	0.492	0.513	0.534
	DLinear-Zero	0.476	0.607	0.961	1.426	2.039	0.629	0.988	1.433	1.980
	DLinear	0.476	0.477	0.476	0.478	0.481	0.477	0.487	0.507	0.528
	NLinear	0.357	0.358	0.356	0.358	0.364	0.359	**0.368**	0.393	0.419
	RLinear	0.357	0.357	0.356	**0.357**	**0.362**	0.359	**0.368**	0.389	0.416
	RLinear-MT	0.359	0.359	0.357	0.357	**0.362**	0.360	**0.368**	**0.388**	**0.415**
	MALin	**0.351**	**0.354**	**0.354**	0.361	0.370	**0.356**	0.372	0.396	0.423
ETTm1	Naive	1.271	1.272	1.274	1.277	1.282	1.260	1.269	1.263	1.318
	Ridge	**0.356**	**0.357**	**0.358**	**0.361**	**0.370**	0.388	0.467	0.549	0.712

© 2026 The Author(s). Published by WhiteField Academic Press

Dataset	Model	clean	P10	P30	P50	P70	B10	B30	B50	B70
	DLinear-Zero	0.360	0.377	0.450	0.578	0.755	0.402	0.506	0.630	0.805
	DLinear	0.360	0.360	0.361	0.365	0.373	0.391	0.470	0.551	0.710
	NLinear	0.368	0.368	0.369	0.371	0.379	0.402	0.487	0.573	0.754
	RLinear	0.362	0.362	0.363	0.366	0.374	0.395	0.481	0.568	0.748
	RLinear-MT	0.361	0.362	0.362	0.364	0.370	0.391	0.467	0.551	0.726
	MALin	0.361	0.361	0.362	0.365	**0.370**	**0.385**	**0.453**	**0.532**	**0.677**

Table 3. Test MAE averaged over the four horizons. Notation as in Table 2.

Dataset	Model	clean	P10	P30	P50	P70	B10	B30	B50	B70
ETTh1	Naive	0.737	0.737	0.735	0.742	0.744	0.742	0.730	0.744	0.715
	Ridge	0.432	0.433	0.437	0.447	0.471	0.445	0.461	0.506	0.534
	DLinear-Zero	0.440	0.464	0.525	0.597	0.676	0.462	0.522	0.586	0.670
	DLinear	0.440	0.440	0.443	0.451	0.473	0.452	0.467	0.509	0.537
	NLinear	0.438	0.438	0.440	0.448	0.469	0.451	0.464	0.508	0.529
	RLinear	0.429	0.430	0.432	0.441	0.464	0.443	0.457	0.504	0.525
	RLinear-MT	0.432	0.432	0.431	**0.437**	**0.456**	0.441	**0.449**	0.492	**0.516**
	MALin	**0.429**	**0.429**	**0.431**	0.439	0.458	**0.436**	**0.449**	**0.486**	**0.516**
ETTh2	Naive	0.481	0.482	0.482	0.483	0.483	0.481	0.480	0.492	0.499
	Ridge	0.472	0.473	0.473	0.475	0.481	0.474	0.483	0.494	0.505
	DLinear-Zero	0.470	0.541	0.696	0.868	1.061	0.546	0.700	0.861	1.040
	DLinear	0.470	0.470	0.471	0.472	0.477	0.471	0.480	0.491	0.502
	NLinear	0.399	0.399	0.399	**0.400**	**0.405**	0.401	**0.407**	0.420	0.434
	RLinear	0.400	0.400	0.399	**0.400**	**0.405**	0.402	0.408	**0.419**	**0.433**
	RLinear-MT	0.403	0.402	0.401	0.401	**0.405**	0.403	0.408	**0.419**	**0.433**
	MALin	**0.395**	**0.396**	**0.397**	0.403	0.410	**0.398**	0.407	0.424	0.435
ETTm1	Naive	0.698	0.698	0.699	0.701	0.701	0.695	0.701	0.700	0.718
	Ridge	0.377	0.378	0.379	0.381	0.388	0.395	0.434	0.474	0.546
	DLinear-Zero	0.382	0.403	0.465	0.546	0.640	0.415	0.488	0.563	0.652
	DLinear	0.382	0.382	0.384	0.386	0.392	0.399	0.438	0.476	0.547
	NLinear	0.385	0.385	0.385	0.387	0.392	0.402	0.443	0.483	0.558
	RLinear	0.379	0.379	0.380	0.382	0.388	0.396	0.438	0.479	0.555
	RLinear-MT	0.377	0.377	**0.377**	**0.379**	**0.384**	0.393	0.431	0.472	0.548
	MALin	**0.376**	**0.376**	**0.377**	**0.379**	**0.384**	**0.389**	**0.425**	**0.465**	**0.530**

MALin reduces the block-missingness degradation on ETTh1 and ETTm1 (Table 2). On ETTh1 its horizon-averaged MSE at 30%, 50%, and 70% block missingness is 0.450, 0.525, and 0.605, against 0.469, 0.563, and 0.644 for RLinear (reductions of 4.1%, 6.7%, and 6.1%). On ETTm1 the corresponding values are 0.453, 0.532, and 0.677 against 0.481, 0.568, and 0.748 (reductions of 5.8%, 6.3%, and 9.5%). Averaged over all three datasets and the 30–70% block rates, MALin lowers MSE by 3.8% relative to RLinear and by 10.7% relative to interpolated DLinear.

ETTh2 is a clear exception. On this dataset, RLinear, RLinear-MT, and NLinear all outperform MALin under block missingness, by about 1–2% (e.g. 0.396 for MALin versus 0.389 for RLinear and 0.388 for RLinear-MT at B50). ETTh2 is also the dataset with the smallest block degradation for all instance-normalised models: RLinear loses 9% at B50 on ETTh2, compared with 34% on ETTh1 and 57% on ETTm1. When interpolation already

© 2026 The Author(s). Published by WhiteField Academic Press

preserves most of the information, the extra mask parameters and the masked training add variance without much to correct.

Under point missingness, MALin and RLinear are statistically indistinguishable in almost all settings (Table 5: 2 of 48 tests significant in favour of MALin, none against). MALin's mean MSE change relative to RLinear over all point conditions is −0.04%.

On complete inputs MALin does not pay a systematic price, contrary to the concern that training on corrupted windows would hurt clean accuracy. Its horizon-averaged clean MSE is 0.426, 0.351, and 0.361 on the three datasets, versus 0.421, 0.357, and 0.362 for RLinear (mean change −0.2%, range −1.5% to +1.3%). Ridge regression is the best model on clean ETTh1 and ETTm1. This is consistent with the observation of Zeng et al. [2] that well-regularised linear maps are strong on these benchmarks, and it confirms that the baselines in this study are competitive rather than weak.

Table 4. Test MSE per horizon on complete inputs (clean) and under 50% block missingness (B50). Best value per row and condition in bold.

Dataset	H	DLinear clean	Ridge clean	RLinear clean	RLinear-MT clean	MALin clean	DLinear B50	Ridge B50	RLinear B50	RLinear-MT B50	MALin B50
ETTh1	96	0.375	**0.370**	0.378	0.390	0.390	0.537	0.539	0.544	0.512	**0.502**
	192	0.410	**0.404**	0.412	0.427	0.420	0.546	0.546	0.554	0.529	**0.517**
	336	0.443	**0.433**	0.436	0.443	0.441	0.564	0.560	0.565	0.540	**0.529**
	720	0.483	0.470	0.456	0.467	**0.452**	0.599	0.588	0.590	0.565	**0.551**
ETTh2	96	0.299	0.301	**0.280**	0.282	0.281	0.340	0.344	0.322	**0.321**	0.327
	192	0.401	0.397	0.346	**0.346**	**0.346**	0.435	0.432	0.382	**0.379**	0.392
	336	0.482	0.485	0.372	0.376	**0.370**	0.508	0.513	**0.401**	0.402	0.407
	720	0.722	0.740	0.428	0.431	**0.409**	0.744	0.764	**0.450**	**0.450**	0.456
ETTm1	96	0.300	**0.299**	0.304	0.301	0.300	0.546	0.544	0.556	0.535	**0.516**
	192	0.336	**0.334**	0.338	0.337	0.337	0.544	0.544	0.558	0.543	**0.525**
	336	0.370	**0.368**	0.371	0.377	0.376	0.545	0.545	0.560	0.550	**0.531**
	720	0.432	**0.423**	0.434	0.431	0.430	0.568	0.564	0.598	0.576	**0.556**

Table 4 breaks the results down by horizon for the clean and B50 conditions. On ETTh1 and ETTm1, MALin has the lowest B50 error at every horizon, and on ETTh2 it is within 0.006–0.013 MSE of the best model. On clean inputs MALin is best at the two longest ETTh2 horizons and at H = 720 on ETTh1.

© 2026 The Author(s). Published by WhiteField Academic Press

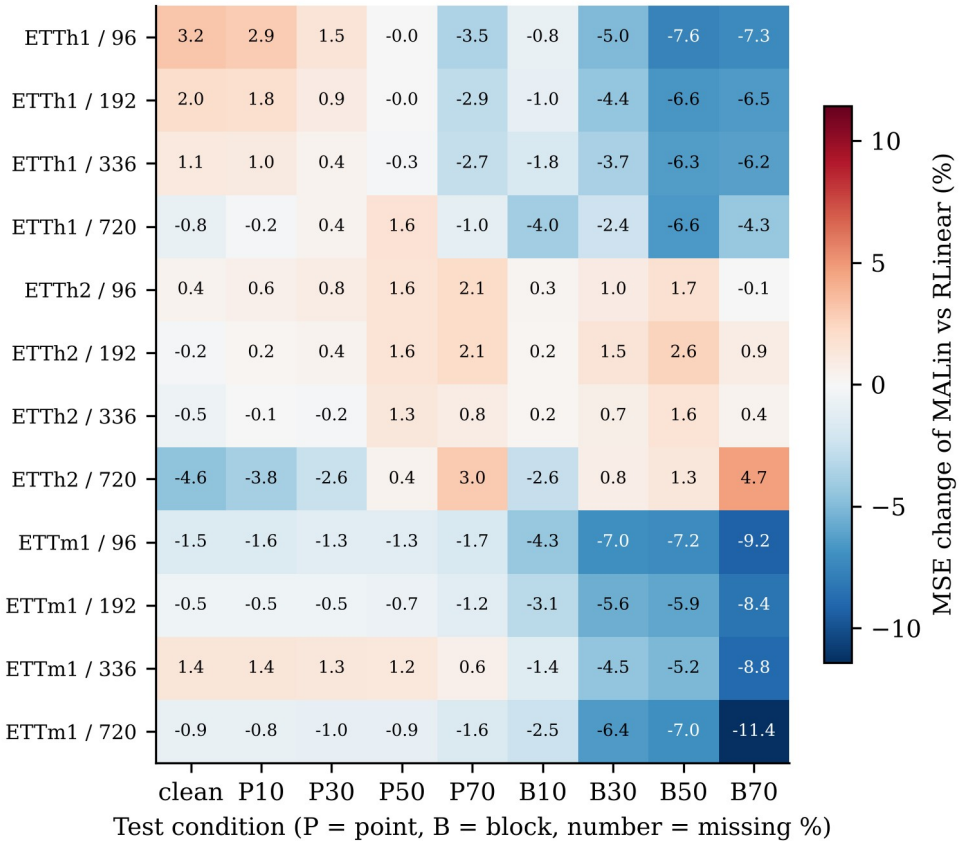


Figure 3. Relative MSE change (%) of MALin with respect to RLinear for every dataset, horizon, and test condition. Blue cells indicate lower error for MALin.

8.3 Statistical analysis

Table 5 summarises the Diebold–Mariano tests. Against RLinear, MALin is significantly better in 24 of the 48 block-missingness settings and significantly worse in none after Holm correction. The significant improvements are concentrated on ETTh1 and ETTm1; the small losses on ETTh2 are not significant under the conservative long-run variance estimator. Against interpolated DLinear and Ridge, MALin is significantly better in 17 and 18 block settings respectively, again without a significant loss. Against RLinear-MT, which shares MALin's training procedure, only 8 of 48 block settings show a significant improvement, and the median MSE change is −1.8%. In other words, about half of MALin's block-missingness gain over RLinear is already obtained by training RLinear on masked windows.

Table 5. Diebold–Mariano tests on per-window MSE differences (Newey–West variance, L + H lags; Holm-adjusted p < 0.05 within each comparison family of 108 tests). Each test is one dataset × horizon × condition. Negative change means MALin has lower MSE.

MALin vs.	Condition group	Tests	MALin significantly better	MALin significantly worse	Not significant	Median MSE change (%)
DLinear	Clean	12	4	0	8	-0.5
DLinear	Point	48	6	0	42	-1.5
DLinear	Block	48	17	0	31	-5.4
Ridge	Clean	12	2	0	10	+0.5
Ridge	Point	48	6	0	42	-0.6
Ridge	Block	48	18	0	30	-5.3
RLinear	Clean	12	0	0	12	-0.3
RLinear	Point	48	2	0	46	+0.1
RLinear	Block	48	24	0	24	-3.8
RLinear-MT	Clean	12	2	0	10	-0.4
RLinear-MT	Point	48	2	1	45	+0.0
RLinear-MT	Block	48	8	0	40	-1.8

© 2026 The Author(s). Published by WhiteField Academic Press

8.4 Ablation study (RQ3)

Table 6. One-at-a-time ablation on ETTh1 (seed 0; MSE averaged over H = 96 and 192, the horizons completed within the compute budget; Point/Block = average over the four missing rates of that pattern). RLinear is shown for reference.

Variant	clean	Point (avg.)	Block (avg.)	B70
MALin (full)	0.405	0.414	0.491	0.608
− mask branch	0.406	0.412	0.491	0.609
− observed-only normalisation	0.406	0.414	0.500	0.626
zero fill instead of interpolation	0.442	0.447	0.460	0.545
− missingness-augmented training	0.395	0.417	0.521	0.670
RLinear (reference)	0.395	0.414	0.518	0.653

The ablation (Table 6) was completed on ETTh1 for H = 96 and 192. It gives a more mixed picture of MALin's components than the design rationale suggested, and we report it as observed.

- **Missingness-augmented training is the main source of robustness.** Removing it restores the clean-input accuracy of RLinear (0.395) but raises the block-averaged MSE from 0.491 to 0.521 and the B70 MSE from 0.608 to 0.670.
- **Observed-only normalisation helps under heavy block missingness.** Without it, the B70 MSE rises from 0.608 to 0.626, while the clean and point results are unchanged.
- **The mask branch has no measurable effect in this configuration.** With interpolated inputs, removing $\mathbf{W}_m$ changes no metric by more than 0.002.
- **Zero filling combined with the mask branch is better under block missingness but worse elsewhere.** It reaches 0.460 block-averaged MSE and 0.545 at B70, the best values in the table. However, it costs 0.037 MSE on clean inputs and 0.033 on point missingness. With zero filling, the mask branch is the only way for the model to know which zeros are real, so it is used. With interpolation, the interpolated values already carry most of what the mask would add.

These results indicate that the essential ingredients for a robust linear forecaster are training on masked windows and normalising with observed statistics. The mask branch is only useful when the filling strategy leaves information for it to recover. Practitioners who expect mostly block outages may prefer the zero-fill variant, and those who need clean-input accuracy may prefer interpolation.

8.5 Error analysis

To see where MALin helps, we related the per-window losses on ETTh1 (H = 96) to the longest gap inside each lookback window, using the saved per-window test losses of the seed-0 runs. At 50% block missingness, MALin has a lower MSE than RLinear in 91.8% of the 2,785 test windows. For windows whose longest gap is 48–95 hours (2,544 windows), the mean MSE falls from 0.557 to 0.513 (−8.0%). For windows with a gap of at least 96 hours (241 windows), the reduction is 2.9%. At 30% block missingness, MALin is better in 69.6% of windows, with reductions of 6.0% for gaps of 24–47 hours and 4.0% for gaps of 48–95 hours.

The windows where MALin loses most are not random. At 30% block missingness, the three largest losses relative to RLinear come from consecutive windows (indices 1557, 1560, and 1561) that share the same 57-hour gap; MALin's MSE in these windows is about 0.55–0.58, against 0.46–0.48 for RLinear. Overlapping windows share gaps, so a single unfavourable outage affects a run of neighbouring forecasts. At 50% block missingness, the largest losses occur in windows with gaps of 82–96 hours, where the margin is about 0.02 MSE. Long gaps covering several daily cycles therefore remain difficult, and in a minority of such windows the learned correction moves the forecast in the wrong direction.

8.6 Computational cost

MALin has $2LH + H + 2C$ trainable parameters: 64,622 at H = 96 and 484,574 at H = 720 with L = 336, compared with 32,366 and 242,654 for RLinear. Inference requires two matrix products per channel plus the normalisation and interpolation, both linear in L. In our CPU runs, one MALin training epoch on ETTh1 took roughly twice as long as one RLinear epoch (for example, about 4.4 s for RLinear and 8.8 s for MALin at H = 96 with one CPU thread), mainly because of mask sampling. The models for missing-value forecasting discussed in Section 2.3 use attention, graph, or state-space layers; we did not run them, so we make no direct cost comparison.

© 2026 The Author(s). Published by WhiteField Academic Press

9.0 Discussion

The results support a simple practical message. If a linear forecaster must run on inputs with gaps, the first decision is not the architecture but the filling strategy, and zero filling should be avoided. The second decision is to train on masked windows. This alone, without any architectural change, accounts for about half of MALin's gain over RLinear under block missingness, and it costs little on complete inputs. The third step, observed-only normalisation, brings a further gain when gaps are long and frequent.

The ablation also tempers the architectural part of our proposal. The mask branch, which motivated the model, is inert when the input is interpolated. A plausible reading is that, for a linear model with shared weights, interpolation already conveys the position and length of a gap implicitly through the straight-line segment, and a mask-dependent additive correction has little left to explain. The branch becomes useful once that implicit signal is removed (zero filling). This is in line with the theoretical point made by Josse et al. [19] and Le Morvan et al. [18]: what matters is that the predictor has access to the information about missingness in a form it can use, not that the imputation is perfect.

The ETTh2 results are a reminder that robustness gains are dataset-dependent. On ETTh2, instance-normalised linear models were already robust to block gaps, and MALin was slightly worse than RLinear. A reasonable deployment rule is therefore to validate the missingness-aware model against plain RLinear on a validation set masked to resemble the expected outages, and to keep the simpler model when the difference is negligible.

10.0 Limitations

- **Synthetic missingness.** All missingness was simulated and is MCAR in placement. Real outages may depend on the signal itself (for example, sensor failures under high load), and such informative missingness was not evaluated.
- **Missing inputs only.** Training targets were complete. Settings in which the training archive itself contains gaps were not studied.
- **Benchmark scope.** Three ETT datasets were evaluated. The ETTm2 runs, and larger benchmarks such as Weather and Electricity, were not completed within the CPU budget.
- **Limited repetitions.** Main comparisons use one training seed, and the ablation covers only ETTh1 at two horizons. Seed-to-seed variation of the linear models is typically small on these datasets, but our significance tests capture only the variation across test windows, not across training runs.
- **No deep missing-value baselines.** We did not run deep missing-value forecasters (BiTGraph, S4M, CoIFNet, CRIB) or a Transformer forecaster such as PatchTST, because of CPU time. MALin is therefore positioned as an improvement over linear baselines, not as a competitor to those models.
- **Fixed hyperparameters.** The augmentation range ($r_{\max} = 0.7$, 50/50 pattern mix) and block lengths were fixed in advance and not tuned.

11.0 Threats to Validity

Internal validity. All models share the same data pipeline, splits, scaler, test masks, training budget, and early-stopping rule, and the test set was never used for model selection. A possible source of bias is that augmented models were selected on a masked validation set and clean-trained models on a clean one; each model was selected under its own training distribution, which is standard practice but not the only defensible choice.

Construct validity. MSE and MAE on standardised data are the conventional metrics on these benchmarks but weight all seven variables equally, whereas operators may care mainly about oil temperature.

External validity. The ETT datasets come from two transformers in one region. Conclusions about the relative value of the components may not transfer to data with different periodicity or noise levels.

Conclusion validity. Per-window losses are strongly autocorrelated. We used a conservative Newey–West bandwidth of $L + H$ lags and Holm correction, so the reported significant results are, if anything, understated.

12.0 Conclusion

We studied how lightweight linear forecasters behave when their input windows contain missing values, and proposed MALin, a linear forecaster with observed-only normalisation, a mask branch, and missingness-

© 2026 The Author(s). Published by WhiteField Academic Press

augmented training. On three ETT datasets, interpolation made point missingness largely harmless, but block missingness of 50% still raised the error of standard linear models by about a third on average. MALin reduced the block-missingness error by 3.8% on average relative to RLinear over 30–70% block rates, and by up to 9.5% on ETTm1, with no systematic loss on complete inputs and no significant loss in any of the 108 Diebold–Mariano comparisons against RLinear. The gains were concentrated on ETTh1 and ETTm1, and MALin was slightly (not significantly) worse than RLinear on ETTh2. The ablation showed that training on masked windows and observed-only normalisation account for the improvement, while the mask branch helps only when gaps are zero-filled.

13.0 Future Work

Natural extensions are evaluation under informative (signal-dependent) missingness and with gaps in the training targets, completion of the ETTm2, Weather, and Electricity experiments with several seeds, and a direct comparison with deep missing-value forecasters under a common protocol. It would also be worth combining zero filling and interpolation as two input views, so that the mask branch can exploit the zero-filled view while the value branch keeps the clean-input accuracy of interpolation.

References

[1] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond efficient Transformer for long sequence time-series forecasting,” in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 12, pp. 11106–11115, 2021, doi: 10.1609/aaai.v35i12.17325.

[2] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are Transformers effective for time series forecasting?,” in Proc. AAAI Conf. Artif. Intell., vol. 37, no. 9, pp. 11121–11128, 2023, doi: 10.1609/aaai.v37i9.26317.

[3] Z. Li, S. Qi, Y. Li, and Z. Xu, “Revisiting long-term time series forecasting: An investigation on linear mapping,” arXiv:2305.10721, 2023.

[4] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A time series is worth 64 words: Long-term forecasting with Transformers,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2023.

[5] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, “Reversible instance normalization for accurate time-series forecasting against distribution shift,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2022.

[6] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, “iTransformer: Inverted Transformers are effective for time series forecasting,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2024.

[7] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “TimesNet: Temporal 2D-variation modeling for general time series analysis,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2023.

[8] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” Scientific Reports, vol. 8, 2018, doi: 10.1038/s41598-018-24271-9.

[9] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, “BRITS: Bidirectional recurrent imputation for time series,” in Advances in Neural Information Processing Systems, vol. 31, 2018.

[10] W. Du, D. Côté, and Y. Liu, “SAITS: Self-attention-based imputation for time series,” Expert Systems with Applications, 2023, doi: 10.1016/j.eswa.2023.119619.

[11] Y. Tashiro, J. Song, Y. Song, and S. Ermon, “CSDI: Conditional score-based diffusion models for probabilistic time series imputation,” in Advances in Neural Information Processing Systems, vol. 34, 2021.

[12] X. Chen, X. Li, B. Liu, and Z. Li, “Biased temporal convolution graph network for time series forecasting with missing values,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2024.

[13] J. Peng, M. Yang, Q. Zhang, and X. Li, “S4M: S4 for multivariate time series forecasting with missing values,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2025.

[14] K. Tang, J. Zhang, H. Meng, M. Ma, Q. Xiong, F. Lv, J. Xu, and T. Li, “CoIFNet: A unified framework for multivariate time series forecasting with missing values,” arXiv:2506.13064, 2025.

[15] J. Yang, Y. Hu, K. Zhang, L. Niu, Y. Dong, P. S. Yu, and K. Ding, “Revisiting multivariate time series forecasting with missing values,” arXiv:2509.23494, 2025.

[16] Y. Hu, H. Chen, P. Liu, Y. Liu, Z. Dong, and J.-M. Yang, “Existence precedes value: Joint modeling of observational existence and evolving states in time series forecasting,” arXiv:2606.13571, 2026.

[17] D. B. Rubin, “Inference and missing data,” Biometrika, vol. 63, no. 3, pp. 581–592, 1976, doi: 10.1093/biomet/63.3.581.

[18] M. Le Morvan, J. Josse, E. Scornet, and G. Varoquaux, “What’s a good imputation to predict with missing values?,” in Advances in Neural Information Processing Systems, vol. 34, pp. 11530–11540, 2021.

[19] J. Josse, J. M. Chen, N. Prost, G. Varoquaux, and E. Scornet, “On the consistency of supervised learning with missing values,” Statistical Papers, vol. 65, no. 9, pp. 5447–5479, 2024, doi: 10.1007/s00362-024-01550-4.

[20] C. Yu, F. Wang, Z. Shao, T. Qian, Z. Zhang, W. Wei, and Y. Xu, “GinAR: An end-to-end multivariate time series forecasting model suitable for variable missing,” arXiv:2405.11333, 2024.

[21] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” Technometrics, vol. 12, no. 1, pp. 55–67, 1970.

[22] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Advances in Neural Information Processing Systems, vol. 32, pp. 8026–8037, 2019.

© 2026 The Author(s). Published by WhiteField Academic Press

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
[24] F. X. Diebold and R. S. Mariano, “Comparing predictive accuracy,” Journal of Business & Economic Statistics, vol. 13, no. 3, pp. 253–263, 1995.
[25] W. K. Newey and K. D. West, “A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix,” Econometrica, vol. 55, no. 3, pp. 703–708, 1987.
[26] S. Holm, “A simple sequentially rejective multiple test procedure,” Scandinavian Journal of Statistics, vol. 6, no. 2, pp. 65–70, 1979.

© 2026 The Author(s). Published by WhiteField Academic Press. This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. *Authors retain copyright.*

INTERNATIONAL JOURNAL OF ADVANCED COMPUTING
AND MACHINE INTELLIGENCE

Peer-Reviewed · Open Access · Published by WhiteField Academic Press
Noida, India

Scan to visit

www.ijacmi.org
editor@ijacmi.org
submissions@ijacmi.org

Visit our website — www.ijacmi.org

© 2026 The Author(s). Published by WhiteField Academic Press