Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Better Imputation Does Not Guarantee Better Forecasting: Transmission Efficiency in Two-Step Time Series Pipelines

Yu Chang
Shaoxing University
Qi Li
Shaoxing University
Sheng Feng
Shaoxing University
Huiqiao Hou
Xi'an University of Architecture and Technology
Yujie Guo
yujie.guo@usx.edu.cn
Shaoxing University

Research Article

Keywords:

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11420039/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Better Imputation Does Not Guarantee Better Forecasting: Transmission Efficiency in Two-Step Time Series Pipelines

Yu Chang[1], Qi Li[1], Sheng Feng[1], Huiqiao Hou[2], Yujie Guo[1*]

[1]Institute of Artificial Intelligence, Shaoxing University, Shaoxing, 312000, Zhejiang, China.
[2]Xi'an University of Architecture and Technology, Xi'an, Shaanxi, China.

*Corresponding author(s). E-mail(s): yujie.guo@usx.edu.cn;

Abstract

Reconstruction accuracy often guides imputer selection for time series forecasting. We examine this practice on four datasets under a historical-window protocol. Inputs exclude observations after the forecast origin; learned imputers are fitted only on training data and frozen during validation and testing. The experiment comprises **2,880** unique configurations and **8,640** training runs across three forecasting seeds (**42**, **123**, **2026**), five imputers and three forecasters: long short-term memory (LSTM), a decomposition-based linear model (DLinear), and a patch-based time series transformer (PatchTST). Forecasting errors are averaged across seeds within each configuration before matched comparisons with Mean imputation. Among **2,127** valid comparisons, median relative changes are **−2.90%** for forecasting mean absolute error (MAE) and **−41.22%** for reconstruction MAE. The median transmission ratio is **0.0449**. Reconstruction improves while forecasting deteriorates in **19.51%** of valid comparisons, with model-specific rates of **34.41%** for LSTM, **6.91%** for DLinear and **17.21%** for PatchTST. Across all **2,304** non-Mean configurations, **74.65%** retain the same forecasting-error-change sign across the three seeds. Reconstruction accuracy alone is therefore insufficient to assess downstream forecasting performance under these settings.

Keywords: Time series forecasting, Missing data imputation, Transmission efficiency, Performance inversion, Two-step pipeline

1 Introduction

Missing observations are common in industrial monitoring, air-quality prediction and other forecasting applications [29]. A standard response is a two-step pipeline: reconstruct the incomplete history, then forecast from the completed input [1, 20]. Imputers are often compared using reconstruction mean absolute error (MAE) or root mean squared error [9, 30]. These metrics assess recovery of missing observations; the downstream objective is accurate prediction of future targets.

Existing forecasting studies show that the most accurate imputer can be suboptimal for a downstream predictor [1, 20]; broader prediction studies also document diminishing returns from improved imputation [15]. Benchmarks provide complementary evidence across methods and missingness settings [9]. These findings leave a practical gap between ranking imputers and assessing the reliability of a reconstruction improvement for a forecasting pipeline. An average ranking or transmission slope does not show how often an individual improvement reverses direction, how large

the forecasting change is, or how stable that outcome remains across training seeds and missingness conditions.

This is a data-quality question with operational consequences: reconstructed histories are useful to a forecasting system when their improvements translate into better future predictions. Choosing an imputer by reconstruction error alone can favour a method that increases downstream error. We address this gap through matched changes in reconstruction and forecasting error under a common forecast-time information constraint. We quantify performance inversion, its variation across pipelines, and the trade-offs between reconstruction-based and downstream-validation-based selection. Scaling and learned imputers are fitted on training data; inputs contain no observations after the forecast origin.

We make three contributions.

1. **Paired transmission diagnostics.** Relative to Mean imputation, the configuration-level ratio τ compares changes in forecasting error (FE) and imputation error (IE). Relative changes, a scale-free elasticity and denominator-sensitivity checks complement the ratio.
2. **An empirical characterisation of performance inversion.** Four datasets, five imputers and LSTM, DLinear and PatchTST forecasters yield 2,880 configurations and 8,640 runs. Among 2,127 valid comparisons, the median relative IE and FE changes are -41.22% and -2.90%; reconstruction improves while forecasting worsens in 19.51%. We examine variation by model, dataset and missingness condition, together with forecasting-seed and leave-one-dataset-out sensitivity.
3. **A downstream selection assessment.** We compare validation-based, fixed-imputer and reconstruction-based choices on 576 tasks, reporting forecasting regret and the frequency of outcomes worse than Mean.

Together, these analyses connect reconstruction-based data-quality assessment to the forecasting outcomes and selection decisions it is intended to support.

2 Related Work

2.1 Missing data mechanisms

Rubin's taxonomy distinguishes missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR) [17, 22]. For time series, pointwise and temporally clustered gaps create different reconstruction tasks, and multiple mechanisms can coexist [13, 34]. We evaluate pointwise MCAR, time-structured Block-MCAR and a synthetic tail-dependent MNAR condition. Block-MCAR is independent of covariate values and is therefore not a MAR mechanism (Section 3.3.1).

2.2 Time series imputation

Statistical and classical methods include Mean, spline interpolation and KNN [6, 26]. Their performance depends on both the missingness pattern and the downstream task; simple methods remain competitive in several evaluations [23, 25, 27]. BRITS uses bidirectional recurrent dynamics [4], whereas SAITS uses diagonally masked self-attention [7]. Both exploit temporal and cross-feature information within an available history.

Probabilistic approaches include GP-VAE [10] and the conditional score-based diffusion imputer CSDI [24]. CSDI learns a conditional distribution of missing values and can generate multiple reconstructions. PyPOTS supports standardised implementations [8], and TSI-Bench evaluates a wider set of imputers across datasets and downstream tasks [9]. Our method set provides contrasting statistical, recurrent and attention-based inputs for paired transmission analysis.

2.3 Time series forecasting models

LSTM models sequential dependencies through gated recurrent states [11, 14]. DLinear decomposes the input into trend and seasonal residuals and forecasts each through a linear map [32]. PatchTST encodes subsequence patches with channel-independent attention [21]. These provide recurrent, linear and attention-based forecasting settings. End-to-end models that directly handle incomplete observations offer an alternative to the decoupled pipeline [5]. We study the latter with fixed architecture and training settings.

2.4 The two-step paradigm and error transmission

Prior forecasting studies show that lower reconstruction error does not always produce lower forecasting error [1, 20]. Suggested explanations include changes in temporal dependence and reconstructed variability [3, 20]. More generally, pointwise reconstruction metrics do not fully describe distributional fidelity or downstream utility [2, 29], and comparable imputation rankings remain an active research topic [19].

Le Morvan and Varoquaux [15] quantify diminishing returns from imputation accuracy to prediction accuracy using regression slopes in tabular regression. Their average transmission measure differs from our configuration-level ratio and inversion diagnostics. Related theory establishes Bayes consistency of impute-then-regress procedures for almost all smooth imputation functions under a universally consistent learner and the stated assumptions [16]. Our experiments examine finite-sample forecasting outcomes.

Wang et al. [28] use generalised representers to estimate downstream effects of candidate imputations without retraining the neural predictor, including missing values used as forecasting labels, and combine strategies according to estimated gains. We instead train the compared two-step pipelines with observed forecast targets and diagnose paired changes in historical reconstruction and forecasting errors. The resulting ratios, inversion frequencies and seed checks complement task-oriented evaluation and benchmark rankings [9].

3 Methodology

3.1 Datasets

We use four datasets spanning energy, air quality, industrial monitoring and weather, with 8,250–52,696 preprocessed time steps.

ETTh1

The hourly electricity-transformer series has 17,420 time steps, seven features and no natural missingness [33].

KDD-Beijing

The hourly Beijing subset of KDD Cup 2018 contains 10,898 time steps from January 2017 to March 2018 [12]. Six indicators ($PM_{2.5}$, PM_{10}, NO_2, CO, O_3, SO_2) at 35 stations give 210 station–indicator features. Natural gaps are excluded from artificial masking and error ground truth, but filled in forecasting inputs.

SWaT

We use 495,000 contiguous one-second observations from 2015-12-22 16:30:00 to 2015-12-28 09:59:59 [18]. Arithmetic means over non-overlapping 60-second windows give 8,250 minute-level time steps. Training-variance selection retains 40 of the 51 sensor features.

Weather

The Jena station series contains 52,696 ten-minute records over 21 variables from 1 January 2020 to 1 January 2021, with no natural missingness [31].

3.2 Historical-window evaluation protocol

All experiments use the same forecast-origin information constraint. For an origin t, the forecasting input is $x_{t-L:t-1}$ with $L = 96$, and the target is the original series $y_{t:t+h-1}$, where $h \in \{24, 48, 96, 192\}$. No target observation is supplied to an imputer. An input history may extend into a preceding chronological split, but its target lies entirely within its assigned training, validation or test split. Training and validation origins advance by one time step; test origins advance by 24.

We split each series chronologically in a 7 : 1 : 2 ratio, without shuffling the split order. `StandardScaler` is fitted on training data only and frozen downstream. For KDD-Beijing and SWaT, columns whose training variance is at most 10^{-5} are removed. Weather retains supplied finite values, including zeros; no whole-sequence interpolation or backward filling is performed before splitting. Naturally missing values remain unknown until input-window imputation, and are never replaced to construct FE targets.

Learned imputers are fitted only on incomplete training data. Their parameters are frozen when completing validation and test histories. Deep imputers receive 96-step training samples with

stride 96, appending the final boundary sample when needed, and subsequently process each historical input window. Means and KNN donors are also fitted only on training data. Observed input entries are preserved. Any NaNs remaining after method-specific reconstruction are replaced with the frozen observed training-column mean. For Spline, this also handles features without observations in the input history; finite KNN or learned-imputer outputs are retained.

Bidirectional reconstruction within the available history is allowed: an earlier missing entry can use a later observation in the same window, provided both precede the forecast origin. Thus, the constraint concerns forecast-time information availability, rather than instantaneous recovery of every missing entry.

3.3 Missing data simulation

3.3.1 Missingness conditions

We construct three missingness conditions separately within each chronological split. Natural gaps are excluded from artificial masking because their truth is unknown. A mask is generated once per mechanism–rate condition and shared across methods, models, horizons and forecasting seeds.

MCAR

Known entries are sampled uniformly without replacement. The target count is the integer part of the requested rate times the number of known entries.

Block-MCAR

Randomly positioned 24-step blocks, spanning all features over the same candidate time interval, are selected until the target number of known entries has been masked; the final selection may be truncated to meet the target count. Blocks are sampled independently of covariate values, defining a time-structured MCAR condition.

MNAR

Within each split and feature, entries at or below the $r/2$ quantile and at or above the $1 - r/2$ quantile are selected, with random adjustment to the target mask count. These full-split quantiles define a synthetic missingness experiment; neither the thresholds nor the concealed values are supplied to the imputer or forecaster.

3.3.2 Missing rates and seed control

The rates are 10%, 30%, 50% and 70%. Missingness and imputer-training seeds are 42. All four datasets use forecasting seeds 42, 123 and 2026. Before each forecasting configuration, Python, NumPy and PyTorch RNGs, including CUDA RNGs, are reset to the forecasting seed; cuDNN deterministic mode is enabled and benchmarking is disabled. Imputed inputs are cached and reused across forecasting models, horizons and forecasting seeds. Each learned imputer is fitted once per upstream condition at seed 42; the three-seed analysis varies only forecasting training, conditional on the fixed masks and imputed inputs.

AI-assisted code development and manuscript preparation

The authors used Codex for language editing and translation, and to assist with developing and revising experiment and statistical-analysis code. The authors reviewed the text, code and numerical outputs, checked statistics by recomputation from configuration-level records, and take full responsibility for the published content.

3.4 Imputation methods

Mean

For feature j, let Ω_j^{tr} denote observed training positions after artificial masking. Missing input entries are filled by

$$\hat{x}_{t,j} = \frac{1}{|\Omega_j^{\mathrm{tr}}|} \sum_{u \in \Omega_j^{\mathrm{tr}}} x_{u,j}. \tag{1}$$

These column means are frozen for validation and testing. Mean is the reference anchor for the transmission analysis.

Range-constrained Spline

We use SciPy's `UnivariateSpline` independently within each available history, with its default smoothing setting (`s=None`). Let $n_{\min}$ denote the minimum observed count across features in the window. The common degree is three when $n_{\min} \geq 4$, and $n_{\min} - 1$ when $2 \leq n_{\min} < 4$. When

$n_{\min} < 2$, the spline step is skipped and within-history filling is used. Splines are evaluated only between the first and last observed points. Boundary gaps use the nearest observation; residual gaps use within-history backward/forward filling. At missing positions only, reconstructed values are constrained to the observed minimum and maximum of that feature in the same history. If spline fitting fails or produces non-finite outputs, we use local linear interpolation. Observed entries are restored, and a feature without any observed history uses the frozen training mean. All bounds are derived from the available input history, not future targets. This range-constrained implementation is used on all four datasets.

KNN

A distance-weighted $k = 5$ nearest-neighbour imputer is fitted on incomplete training rows and frozen. Each query row uses observed cross-feature values and the training donors. The missingness-aware distance is

$$D(x_a, x_b) = \sqrt{\frac{C}{|O|} \sum_{j \in O} (x_{a,j} - x_{b,j})^2}, \qquad (2)$$

where C is the number of features and O contains features observed in both rows. The distance is undefined when O is empty. Undefined donor distances are ignored; if no donor has a defined distance for a missing feature, the implementation uses its observed training-column mean. Query rows do not become validation or test donors.

BRITS

The bidirectional recurrent imputation model [4] captures temporal and cross-feature dependencies. It is fitted on 96-step training samples and applied without refitting to each available historical input window. Its backward branch can use later observations within that history, but no post-origin observations.

SAITS

The self-attention-based imputation model [7] reconstructs missing entries using attention over the available history. It follows the same training-only fitting and frozen inference protocol as BRITS. Structural parameters and dataset-specific budgets for both deep imputers are listed in B.

3.5 Forecasting models

We use a two-layer LSTM with hidden size 64 and a direct multi-horizon output head [14]; DLinear with a 25-step moving average and individual feature-wise component maps [32]; and supervised PatchTST with RevIN, 16-step patches and stride eight [21]. Settings are fixed across imputers within each dataset. Tables 14 and 15 give the complete architectural and training configurations.

3.6 Error metrics and transmission diagnostics

Both IE and FE are evaluated using MAE and mean squared error (MSE) on the series standardised with training-fitted parameters. Let W_h denote test input windows retained for horizon h. IE is computed only at artificially masked, truth-known positions within the test segment that occur in W_h. Each feature's error is averaged over its selected window occurrences, then the available feature errors are averaged with equal weight. Overlapping occurrences of the same time point count separately. Longer horizons retain fewer windows, so IE depends on h; it is not averaged across forecast horizons before forming the ratio. FE averages feature-wise errors over original, unmasked forecast targets in those windows; naturally unknown targets are excluded. Imputed targets are never used as FE ground truth.

For dataset d, forecaster g, imputer ϕ, missingness condition m, rate r and horizon h, let $\overline{\mathrm{FE}} = \frac{1}{3} \sum_{s \in \{42,123,2026\}} \mathrm{FE}_s$ denote the forecasting error averaged over the three seeds within a configuration. The primary contrasts are

$$\Delta \mathrm{IE}_{d,\phi,m,r,h} = \mathrm{IE}_{d,\phi,m,r,h} - \mathrm{IE}_{d,\mathrm{Mean},m,r,h}, \quad (3)$$

$$\Delta \overline{\mathrm{FE}}_{d,g,\phi,m,r,h} = \overline{\mathrm{FE}}_{d,g,\phi,m,r,h} - \overline{\mathrm{FE}}_{d,g,\mathrm{Mean},m,r,h}. \quad (4)$$

IE has no forecasting-seed dependence because the inputs are fixed. The configuration-level transmission ratio is

$$\tau(d, g, \phi, m, r, h) = \frac{\Delta \overline{\mathrm{FE}}_{d,g,\phi,m,r,h}}{\Delta \mathrm{IE}_{d,\phi,m,r,h}}, \quad |\Delta \mathrm{IE}| > \epsilon, \qquad (5)$$

with $\epsilon = 0.01$ by default. Otherwise the ratio is undefined, although the trained configuration remains in the error tables. We examine

$\epsilon \in \{0.001, 0.005, 0.01, 0.02, 0.05\}$ and MSE-based errors as sensitivity checks. Mean rows provide the matched within-configuration reference and are not counted as comparisons. In the seed-specific sensitivity analysis, $\Delta\mathrm{FE}_{d,g,\phi,m,r,h,s} = \mathrm{FE}_{d,g,\phi,m,r,h,s} - \mathrm{FE}_{d,g,\mathrm{Mean},m,r,h,s}$ replaces $\Delta\overline{\mathrm{FE}}$. Thus averaging precedes sign classification in the primary analysis, whereas the seed-consistency analysis compares the individual-run signs.

Pure performance inversion is $\Delta\mathrm{IE} < 0$ and $\Delta\overline{\mathrm{FE}} > 0$ among valid ratios. Global inversion is $\tau < 0$, also including reverse cases with $\Delta\mathrm{IE} > 0$ and $\Delta\overline{\mathrm{FE}} < 0$. A positive ratio can represent either two improvements or two deteriorations, so its sign alone is not proof of downstream benefit. To describe the magnitude of pure inversion, we additionally examine $100\Delta\overline{\mathrm{FE}}/\overline{\mathrm{FE}}_{\mathrm{Mean}}$ within those cases. A tolerance sweep requires this relative increase to exceed $\delta \in \{0, 0.1, 0.5, 1, 2, 5\}\%$, retaining the original valid-ratio denominator. The scale-free companion is

$$\eta = \frac{\Delta\overline{\mathrm{FE}}/\overline{\mathrm{FE}}_{\mathrm{Mean}}}{\Delta\mathrm{IE}/\mathrm{IE}_{\mathrm{Mean}}}. \tag{6}$$

Relative changes and ratios are computed within each configuration before aggregation. η and τ share a sign by construction when their baselines are positive. We emphasise medians, interquartile range (IQR) widths and trimmed means, while retaining arithmetic means as descriptive contrasts.

3.7 Experimental matrix and aggregation

Each of the 2,880 unique configurations in Table 1 is run with three forecasting seeds, giving 8,640 records. The main analysis pairs the three-seed mean FE of each non-Mean imputer with the corresponding Mean FE, yielding 2,304 comparisons before denominator filtering. All methods retain the same weighting across datasets, models, mechanisms, rates and horizons. Sample standard deviations (SDs) use $n - 1$ in the variance denominator. For an aggregate method summary, errors are first averaged over conditions within each seed; we then report the mean and sample SD of these three seed-specific aggregate errors. Configuration-level SDs are provided separately. Ranking comparisons retain exact tied minima and test whether the best-IE and best-FE method sets intersect. Rankings use test errors to describe agreement; an operational selection procedure would use validation data.

Table 1 Experimental matrix. Three forecasting seeds (42, 123, 2026) give 8,640 runs over 2,880 unique configurations. Missingness and imputer-training seeds are fixed at 42.

Dimension	Values	Count
Dataset	ETTh1, KDD-Beijing, SWaT, Weather	4
Missingness	MCAR, Block-MCAR, MNAR	3
Missing rate	10%, 30%, 50%, 70%	4
Imputer	Mean, Spline, KNN, BRITS, SAITS	5
Forecaster	LSTM, DLinear, PatchTST	3
Horizon	24, 48, 96, 192	4

3.7.1 Forecasting-seed analysis

For each of 2,304 non-Mean configurations we compare the negative, zero or positive sign of $\Delta\mathrm{FE}_s$ across the three forecasting seeds. All-three agreement requires all signs to match. Mean pairwise disagreement averages the three seed-pair disagreement indicators within each configuration, then averages across configurations. The proportion with any disagreement is reported separately. These sign measures include every non-Mean configuration, whereas pure-inversion rates use the valid-ratio denominator.

3.7.2 Uncertainty and paired comparisons

Confidence intervals use 20,000 group-bootstrap resamples, retaining correlated forecast horizons within each group. Matched comparisons use grouped randomisation tests with Holm adjustment for multiple comparisons; group keys and resampling details are given in F.

Dataset influence check

We repeat the configuration-mean summaries after omitting each dataset in turn, without retraining or changing the denominator rule, to assess each dataset's influence on the pooled result.

3.7.3 Imputer-selection analysis

A selection unit $u = (d, g, m, r, h)$ fixes all factors except the imputer, giving 576 units with five candidates each. We average validation MSE and test FE over the three forecasting seeds within each candidate before selection. The validation rule minimises the observed-target-cell validation MSE recorded for the restored checkpoint; testing uses the feature-macro errors defined above. Fixed-imputer rules keep the candidate constant. Lowest test IE-MAE and IE-MSE are additional reconstruction-optimal diagnostic references. The validation rule uses no test errors in selecting its candidate.

For test metric $q \in \{\mathrm{MAE}, \mathrm{MSE}\}$ and selected imputer $\hat{\phi}_u$, let $F^*_{u,q} = \min_\phi \overline{\mathrm{FE}}_{u,\phi,q}$. Relative regret is

$$R_{u,q} = 100 \frac{\overline{\mathrm{FE}}_{u,\hat{\phi}_u,q} - F^*_{u,q}}{F^*_{u,q}}.$$

The test-error minimum is used only as an evaluation reference. We also report the proportion of units whose selected test MAE exceeds Mean. This latter measure uses all 576 units and is distinct from pure inversion, which requires a valid IE improvement. No selection score has a tied minimum in the current data.

Selection-rule comparisons use the grouped protocol in F.

3.8 Training and implementation

All forecasting models use Adam, a maximum of 100 epochs, patience 20, minimum validation-MSE improvement 10^{-6} and gradient-norm clipping at 1. Learning rates are 10^{-3} for DLinear and 10^{-4} for LSTM and PatchTST; batch size is 128, except 32 for KDD-Beijing PatchTST. Settings are fixed across imputers and forecasting seeds. Training and early stopping use MSE over observed target cells; reported FE is the macro average of feature-specific errors. Validation-selected weights are restored for testing. The checkpoint convention is detailed in D; A records the computing environment and F describes the accompanying numerical materials.

Table 2 Forecasting errors over three seeds on all four datasets. Each seed-specific method mean averages 144 conditions (models, mechanisms, rates and horizons). SD is the sample SD of the three seed-specific means.

Dataset	Imputer	MAE mean ± SD	MSE mean ± SD
ETTh1	Mean	0.5909 ± 0.0006	0.6893 ± 0.0015
ETTh1	Spline	0.5581 ± 0.0013	0.6423 ± 0.0035
ETTh1	KNN	0.5978 ± 0.0005	0.7002 ± 0.0014
ETTh1	BRITS	0.5702 ± 0.0011	0.6527 ± 0.0031
ETTh1	SAITS	0.5795 ± 0.0010	0.6764 ± 0.0032
KDD-Beijing	Mean	0.5991 ± 0.0010	0.8662 ± 0.0034
KDD-Beijing	Spline	0.5913 ± 0.0034	0.8730 ± 0.0041
KDD-Beijing	KNN	0.5899 ± 0.0021	0.8535 ± 0.0023
KDD-Beijing	BRITS	0.5902 ± 0.0016	0.8517 ± 0.0046
KDD-Beijing	SAITS	0.5873 ± 0.0011	0.8476 ± 0.0023
SWaT	Mean	0.2730 ± 0.0017	0.3298 ± 0.0013
SWaT	Spline	0.2492 ± 0.0018	0.3225 ± 0.0019
SWaT	KNN	0.2751 ± 0.0027	0.3306 ± 0.0022
SWaT	BRITS	0.2549 ± 0.0035	0.3136 ± 0.0035
SWaT	SAITS	0.2616 ± 0.0026	0.3242 ± 0.0029
Weather	Mean	0.3180 ± 0.0011	0.2660 ± 0.0018
Weather	Spline	0.2579 ± 0.0011	0.2011 ± 0.0019
Weather	KNN	0.3194 ± 0.0005	0.2665 ± 0.0007
Weather	BRITS	0.2782 ± 0.0010	0.2162 ± 0.0013
Weather	SAITS	0.2720 ± 0.0017	0.2116 ± 0.0025

4 Results and Analysis

All 8,640 records are successful, finite and complete on the experimental grid. Averaging FE across forecasting seeds gives 2,880 configurations and 2,304 non-Mean contrasts. The default rule retains 2,127 valid ratios (92.32%); all configurations remain in the error summaries.

4.1 Reconstruction and forecasting performance

Pooled median reconstruction MAEs are 0.318 for Spline, 0.360 for SAITS, 0.438 for BRITS, 0.608 for KNN and 0.632 for Mean. Forecasting rankings vary by dataset and model (Figure 1). Each plotted point pools configuration-level three-seed mean FE over datasets, mechanisms and horizons.

Table 2 shows that Spline has the lowest aggregate forecasting MAE on ETTh1, SWaT and Weather, while SAITS leads on KDD-Beijing. However, the reconstruction-optimal and forecasting-optimal method sets are disjoint in 74/144, 113/144, 74/144 and 41/144 task units, respectively. The objective also matters: on SWaT, Spline gives the lowest MAE (0.2492), whereas BRITS gives the lowest MSE (0.3136). These differences

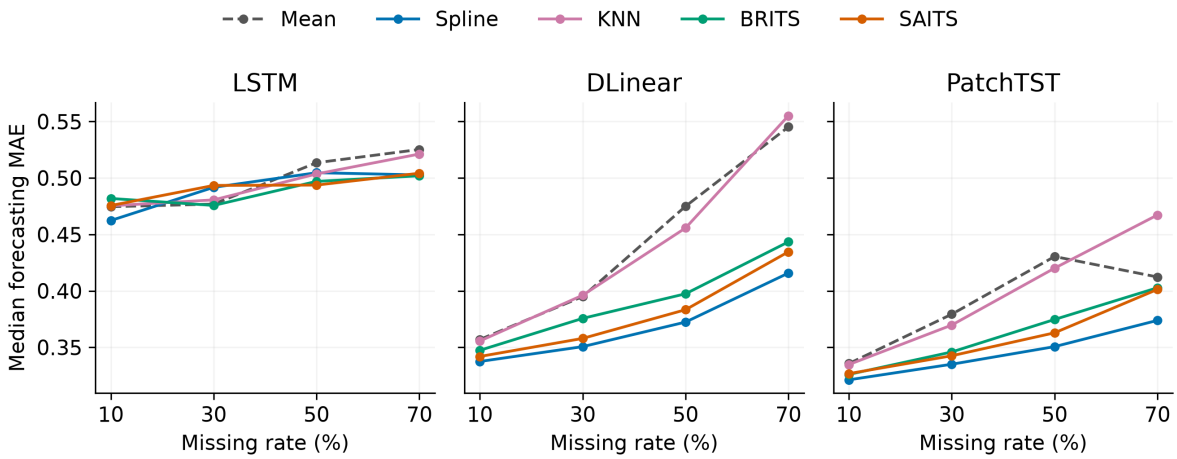


Fig. 1 Median downstream forecasting MAE by forecaster, imputer and missing rate. Errors are first averaged over forecasting seeds 42, 123 and 2026 within every configuration, then pooled over datasets, mechanisms and horizons. The dashed Mean curve is the reference imputer.

motivate condition-specific downstream selection rather than an aggregate imputer ranking.

4.2 Transmission and performance inversion

The median relative IE and FE changes are −41.22% and −2.90%. The median τ is 0.0449 (quartiles 0.0070, 0.1517), and median elasticity is 0.0901. These are separate summaries of paired changes; the quotient of the marginal medians is not the median ratio. Figure 2 includes both valid and denominator-excluded contrasts. Pure inversion occurs in 415/2,127 valid comparisons (19.51%), and global inversion in 21.11%. Among 1,998 comparisons with $\Delta\text{IE} < -0.01$, 20.77% worsen forecasting.

Pure inversion occurs on every dataset, from 10.08% on Weather to 24.19% on KDD-Beijing (Table 3). Its median relative FE deterioration is 1.71% (quartiles 0.71%, 3.77%). KDD-Beijing has the smallest median τ (0.0219) and the highest ranking mismatch (113/144), whereas Weather has the lowest mismatch (41/144). The error, ratio and ranking summaries consistently distinguish these datasets. G complements these error comparisons with exploratory measurements of first-difference amplitude and lag-1 autocorrelation recovery.

4.3 Sensitivity and robust summaries

Across MAE thresholds 0.001–0.05, pure-inversion rates range from 18.85% to 19.81% (Table 4; Figure 3). Each threshold defines its own valid population. The MSE check uses squared-error units and gives 19.21% at the numerical threshold 0.01.

The arithmetic mean exceeds the trimmed means and median, reflecting a positive ratio tail (Table 5). Changing the anchor changes the comparison: Spline gives 9.05% pure inversion, while Mean remains the primary reference (Table 6). Holding the default valid population fixed, requiring relative FE deterioration above 0.1%, 0.5%, 1%, 2% and 5% gives rates of 18.76%, 15.75%, 12.65%, 9.03% and 3.76%.

Omitting each dataset in turn leaves pure inversion at 17.98%–22.53% and median τ at 0.0394–0.0588 (Table 17). LSTM retains the highest and DLinear the lowest inversion rate in every omission. The pooled pattern is therefore not attributable to a single dataset in this grid.

4.4 Model, imputer and missingness differences

Pure-inversion rates are 34.41% for LSTM, 6.91% for DLinear and 17.21% for PatchTST, each over 709 valid contrasts (Table 7). The LSTM–DLinear gap is 27.50 percentage points. On Weather, DLinear has no pure inversion in any forecasting seed

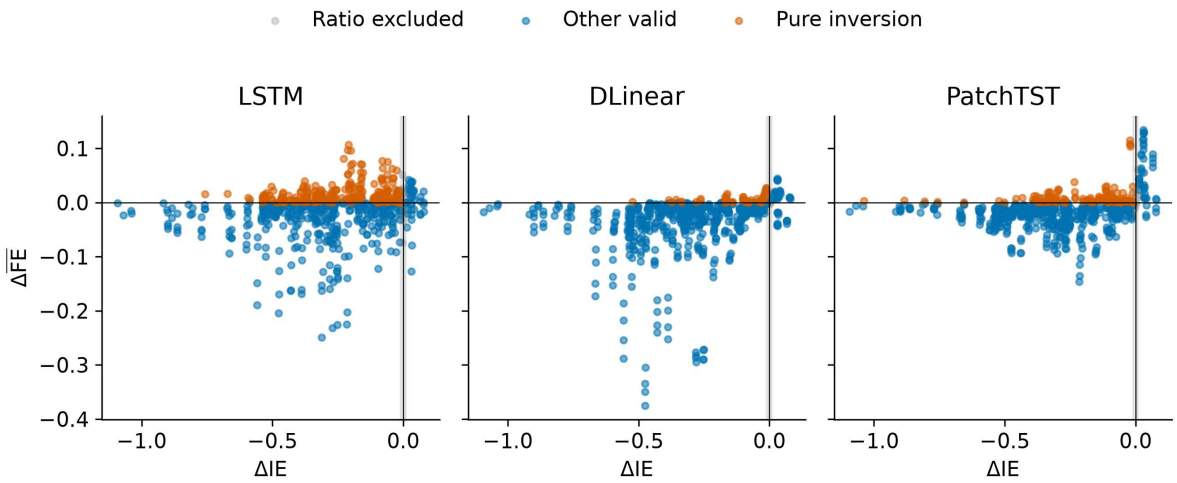


Fig. 2 Configuration-level changes relative to Mean after averaging forecasting errors over three seeds. All 2,304 non-Mean configurations are shown, separated by forecaster. The shaded band marks $|\Delta\text{IE}| \leq 0.01$; orange points denote valid pure inversion.

Table 3 Transmission summaries using three-seed mean forecasting errors; IQR denotes the interquartile range width.

Dataset	Valid N	τ median	τ IQR	$\Delta\overline{\text{FE}}$ median	ΔIE median	Pure inversion
ETTh1	528	0.0628	0.2680	−0.0089	−0.1251	20.83%
KDD-Beijing	525	0.0219	0.0658	−0.0063	−0.3210	24.19%
SWaT	558	0.0629	0.1544	−0.0105	−0.2335	22.58%
Weather	516	0.0544	0.1323	−0.0214	−0.4718	10.08%
Overall	2127	0.0449	0.1447	−0.0112	−0.2759	19.51%

Table 4 Denominator-threshold and error-metric sensitivity. Forecasting errors are averaged over three seeds before forming contrasts.

Metric	ϵ	Valid N	Pure inversion	Global inversion
MAE	0.001	2226	19.81%	21.61%
MAE	0.005	2154	19.50%	21.17%
MAE	0.010	2127	19.51%	21.11%
MAE	0.020	2043	19.09%	20.51%
MAE	0.050	1830	18.85%	19.40%
MSE	0.010	2139	19.21%	24.54%

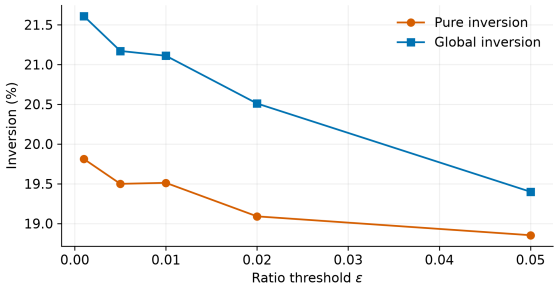


Fig. 3 Pure and global inversion rates as the MAE denominator threshold varies. Each point uses the valid comparisons at that threshold.

(Table 16). Figure 4 shows model–imputer interactions: PatchTST has 27.5% pure inversion with KNN (38/138), versus 12.6% with Spline (24/191) and 13.5% with BRITS (26/192). These measured differences identify settings for downstream validation.

Unmatched pure-inversion rates are 26.67% for Block-MCAR, 24.60% for MCAR and 8.60% for simulated MNAR. Denominator filtering excludes 153/768 Block-MCAR contrasts (138 KNN, 12 SAITS, 3 Spline), versus 12/768 for each other mechanism, all KNN. Thus KNN has $576 - 138 - 12 - 12 = 414$ valid contrasts.

Restricting mechanisms to their common 615 valid configurations gives 26.67% inversion for Block-MCAR (164/615), 25.04% for MCAR

Table 5 Robust summaries of the default MAE transmission ratio.

Summary	Value
Arithmetic mean	0.1148
5% trimmed mean	0.0907
10% trimmed mean	0.0768
Median	0.0449
First quartile	0.0070
Third quartile	0.1517

Table 6 Anchor sensitivity. Each anchor is excluded from its own comparison set; denominator validity is recomputed relative to that anchor.

Anchor	Valid N	τ median	Pure inversion	Global inversion
Mean	2127	0.0449	19.51%	21.11%
Spline	2253	0.0636	9.05%	30.09%

Table 7 Pure inversion by stratum with 95% horizon-group bootstrap intervals (20,000 draws).

Stratum	Valid N	Pure inversion	95% CI (%)
DLinear	709	6.91%	[3.94, 10.31]
LSTM	709	34.41%	[29.49, 39.63]
PatchTST	709	17.21%	[13.14, 21.55]
BRITS	576	17.01%	[12.67, 21.70]
KNN	414	21.98%	[16.18, 28.26]
SAITS	564	21.99%	[16.67, 27.48]
Spline	573	17.80%	[12.78, 23.04]
Block-MCAR	615	26.67%	[21.07, 32.51]
MCAR	756	24.60%	[20.11, 29.37]
MNAR	756	8.60%	[5.82, 11.64]

(154/615) and 7.80% for MNAR (48/615). Corresponding median ratios are 0.0611, 0.0226 and 0.0788, and median relative FE changes are −1.44%, −2.01% and −7.01%. Matching reduces the Block-MCAR–MCAR gap from 2.06 to 1.63 percentage points and retains the lower MNAR inversion rate.

All three paired model comparisons have Holm-adjusted $p < 0.001$ (Table 8). BRITS has less pure inversion than SAITS on their common valid conditions (adjusted $p = 0.043$); Spline–SAITS and Spline–KNN differences do not pass the six-comparison adjustment.

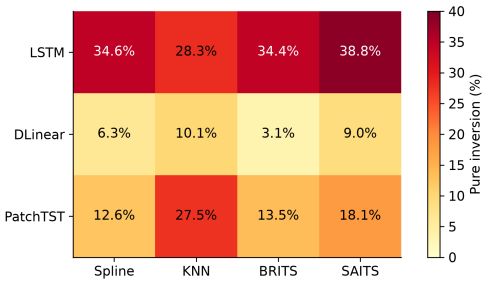


Fig. 4 Pure inversion rates by forecasting model and non-Mean imputer. Each cell uses its own valid-ratio denominator after averaging forecasting errors over three seeds.

4.5 Forecasting-seed consistency

Across 2,304 non-Mean configurations, 1,720 (74.65%) retain the same FE-change sign across three seeds; 25.35% disagree in at least one pair. Mean pairwise disagreement is 16.91% (Table 9).

Seed-specific pure-inversion rates are 20.36%, 20.17% and 19.79%; classification after seed averaging gives 19.51%. Pure inversion occurs in all seeds for 200 configurations and in at least one for 677. E gives dataset–model rates.

All-three FE-sign agreement is 95.05% for DLinear (730/768), 83.59% for PatchTST (642/768) and 45.31% for LSTM (348/768). Among 709 valid contrasts per model, all-three pure-inversion counts are 39, 77 and 84, or 5.50%, 10.86% and 11.85%, respectively. These are 39/49, 77/122 and 84/244 of the corresponding mean-classified pure inversions. Aggregate persistence therefore coexists with substantial configuration-level forecasting variability, particularly for LSTM.

4.6 Implications for imputer selection

Table 10 compares choices over 576 tasks. The validation rule minimises three-seed mean validation MSE; test-IE rules provide reconstruction-optimal diagnostic references. Regret is excess test error over the best of five candidates for the same task and metric.

Compared with fixed Spline, validation-MSE selection reduces test-MAE deterioration relative to Mean from 20.66% to 11.11% (difference −9.55 percentage points, 95% interval $[-13.19, -6.08]$, adjusted $p < 0.001$). MSE regret falls from 3.87% to 1.98% (difference −1.89 points, interval $[-2.87, -0.97]$, adjusted $p < 0.001$). Fixed

Table 8 Paired pure-inversion comparisons on the common valid configurations. Differences are A minus B in percentage points; intervals resample whole horizon groups, and two-sided grouped randomisation p-values are Holm-adjusted over six comparisons.

A vs B	Matched N	Difference (pp)	95% CI (pp)	Adjusted p
LSTM vs DLinear	709	27.50	[21.55, 33.57]	< 0.001
LSTM vs PatchTST	709	17.21	[11.13, 23.31]	< 0.001
PatchTST vs DLinear	709	10.30	[5.63, 15.05]	< 0.001
Spline vs SAITS	564	−3.90	[−8.69, 0.71]	0.242
Spline vs KNN	414	−5.07	[−11.84, 1.46]	0.242
BRITS vs SAITS	564	−6.03	[−10.82, −1.42]	0.043

Table 9 Forecasting-error-change signs over all non-Mean configurations, including those excluded from ratio summaries. Pairwise disagreement averages the three seed pairs.

Dataset	Units	All-three agree	Any disagree	Pairwise disagree
Overall	2,304	74.65%	25.35%	16.91%
ETTh1	576	76.74%	23.26%	15.57%
KDD-Beijing	576	68.75%	31.25%	20.83%
SWaT	576	72.57%	27.43%	18.29%
Weather	576	80.56%	19.44%	12.96%

Spline instead has lower MAE regret: 1.88% versus 2.40% (difference 0.52 points, interval $[-0.39, 1.45]$, adjusted $p = 0.794$). The validation rule selects Spline 211 times, BRITS 133, SAITS 109, KNN 73 and Mean 50.

Validation selection has lower MSE regret than lowest test-IE-MAE selection (1.98% versus 3.52%, adjusted $p = 0.0014$); the difference from test-IE-MSE (2.31%) is smaller (adjusted $p = 0.794$). Model-specific outcomes differ. For LSTM, changing from test-IE-MAE to validation selection lowers MAE regret from 4.50% to 1.78%, MSE regret from 6.52% to 2.94%, and MAE deterioration relative to Mean from 40.10% to 17.19% (36.98% with fixed Spline).

For DLinear, validation MSE regret is 0.83%, versus 1.25% for test-IE-MAE and 1.62% for fixed Spline. Its MAE regret is higher (1.67% versus 0.96% and 0.68%), with MAE deterioration frequencies of 8.33%, 5.73% and 8.33%. For PatchTST, validation selection likewise has higher MAE regret (3.76% versus 1.17% and 0.93%), but lower MSE regret (2.17% versus 2.79% and 3.97%) and less MAE deterioration relative to Mean (7.81% versus 14.58% and 16.67%). Thus selection should consider the intended forecasting metric, strong fixed baselines and the frequency of adverse outcomes together.

5 Conclusion

Better historical-window reconstruction does not guarantee better forecasting in the evaluated two-step pipelines. Across 2,880 configurations and three forecasting seeds, pure inversion occurs in 415/2,127 valid comparisons (19.51%). Median relative IE and FE improvements are 41.22% and 2.90%, and median τ is 0.0449. The pattern occurs on every dataset and remains after omitting each dataset in turn. Forecasting-seed sign agreement is 74.65% overall, with substantial model-specific variation.

The practical finding is a selection trade-off. Validation-MSE selection reduces MSE regret and MAE deterioration relative to Mean compared with fixed Spline, while Spline has lower average MAE regret. Imputers should therefore be assessed on downstream validation performance for the intended metric, alongside fixed baselines and the frequency of adverse outcomes.

5.1 Scope and limitations

Experimental coverage

The study covers four datasets with 8,250–52,696 preprocessed steps, fixed masks and learned imputers at seed 42, and three forecasting-training seeds. Larger datasets, independently regenerated masks and refitted imputers would broaden coverage of pipeline variability. Diffusion-based imputers such as CSDI [24] are not evaluated.

Protocol and inference

The protocol permits bidirectional reconstruction within the available history and uses synthetic

Table 10 Imputer selection over 576 task units. Regret is averaged over units relative to each metric's test-error oracle. The last column reports the frequency of test MAE exceeding Mean and its 95% horizon-group bootstrap interval. Lowest test-IE rules are diagnostic references using test reconstruction errors.

Selection rule	MAE regret (%)	MSE regret (%)	MAE worse than Mean (%) [95% CI]
Validation MSE	2.40	1.98	11.11 [7.81, 14.76]
Lowest test IE-MAE	2.21	3.52	20.14 [15.10, 25.52]
Lowest test IE-MSE	2.23	2.31	16.15 [11.98, 20.66]
Fixed Spline	1.88	3.87	20.66 [15.28, 26.22]
Fixed Mean	12.76	15.30	0.00 [0.00, 0.00]

full-split quantile masking for MNAR. Streaming latency and adaptive fitting are outside its scope. Grouped estimates describe the evaluated grid, with remaining dependencies through shared series, masks and anchors. The comparisons and supplementary signal diagnostics are observational; identifying causal mechanisms would require controlled perturbations or ablations. Training uses fixed budgets and validation-selected checkpoints (D).

Statements and Declarations

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author contributions

Yu Chang: Visualization, Writing – original draft, Investigation, Formal analysis, Validation, Software. Qi Li: Resources. Sheng Feng: Data curation. Huiqiao Hou: Project administration. Yujie Guo: Conceptualization, Methodology, Supervision, Writing – review & editing.

Data availability

The third-party datasets should be obtained from their original sources: ETTh1 from the Informer repository [33], KDD Cup 2018 Beijing air-quality data through the Monash Time Series Forecasting Repository [12], Weather from the Autoformer repository [31], and SWaT from iTrust upon completion of its data request and approval [18]. Neither raw nor processed SWaT data are redistributed. The SWaT input used here is minute-level; raw-to-minute aggregation should follow the preprocessing description in Section 3. Numerical analysis code and archived experimental metrics are available at https://github.com/SecretFzzf/shure. The accompanying numerical analysis materials (Online Resource 1) provide the 8,640 seed-specific result rows, three-seed configuration summaries, and analysis scripts for reproducing the reported tables and figures. These materials contain metrics and configuration records rather than raw time-series data. The accompanying source package also contains signal-diagnostic summaries and extraction code; recomputing those summaries uses the archived configuration-level signal metrics, whereas repeating extraction requires the matching data and frozen imputation caches. Intermediate imputation arrays and trained weights are not redistributed. The result tables permit numerical recomputation; exact training-output regeneration additionally depends on the data-processing version, fixed imputed inputs and training environment.

Appendix A Computing environment

The recorded training environment used Python 3.10.20 with PyTorch 2.12.0+cu130 as the deep learning framework (CUDA available). The key library versions were NumPy 2.2.6, pandas 2.3.3, SciPy 1.15.3, scikit-learn 1.7.2, and PyPOTS 1.5. The hardware comprised eight NVIDIA A100-PCIE-40GB GPUs (40 GB of memory per GPU), dual Intel Xeon Gold 6248R processors at 3.00 GHz (24 physical cores per socket, 48 physical cores and 96 logical cores in total, with AVX-512 support),

and 754 GiB of system memory. At the time of the latest environment check, the NVIDIA driver version was 580.142, and `nvidia-smi` reported CUDA compatibility version 13.0. Numerical recomputation of the archived metrics used Python 3.12.14, NumPy 2.3.5, pandas 3.0.1 and SciPy 1.15.3; figure generation used Matplotlib 3.11.2. These analysis versions are distinct from the training environment and are specified in the accompanying analysis requirements.

Appendix B Training-fitted imputation configuration

Deep imputers use 96-step training samples with stride 96, appending the last boundary sample when needed. Training batch size is 32 and inference batch size is 256. Only incomplete training data are supplied to fitting; validation and test histories are reconstructed without refitting. Table 11 lists the recorded budgets and sample counts. Tables 12 and 13 give the shared structural parameters.

Section 3 specifies Mean, range-constrained Spline and KNN, including history-derived bounds, missingness-aware donor distances and frozen training-mean fallbacks.

Appendix C Forecasting model architecture and training configuration

Tables 14 and 15 summarise the forecasting architectures and training settings. The DLinear source follows the official LTSF-Linear implementation; PatchTST uses the official supervised backbone. DLinear uses individual feature-specific component maps. PatchTST shares its projection and head across channels, uses batch normalisation, learnable positional encoding, residual attention, and zero attention/head dropout. It uses the branch without additional series decomposition. The LSTM is a direct multi-horizon recurrent baseline.

Appendix D Forecasting early stopping and random control

Validation MSE is pooled over observed target cells. The improvement tolerance is 10^{-6} and patience is 20 for all datasets. Testing reloads validation-selected weights. ETTh1 applies the tolerance to checkpoint updates and patience; the other datasets apply it to patience and save every strict validation improvement. Forecasting seeds 42, 123 and 2026 are used on every dataset; missingness and imputer fitting remain at seed 42. Python, NumPy, PyTorch and CUDA RNGs are reset for each configuration. cuDNN deterministic mode is enabled and benchmarking disabled. Runtime configuration details are recorded in the accompanying numerical materials.

Across the 8,640 runs, 948 reach the 100-epoch cap, including 340/720 SWaT PatchTST runs and 225/720 Weather PatchTST runs. These are completed, finite runs evaluated with validation-selected weights; reaching the cap alone establishes neither convergence nor a training failure.

Appendix E Fixed-input and model-specific seed checks

Each dataset uses 60 upstream input conditions (three mechanisms, four rates, five imputers). Recorded IE values agree across the three forecasting seeds and models. The complete result grid has no duplicate keys, missing configurations or non-finite error metrics. Table 16 reports seed-specific rates before averaging forecast errors.

Appendix F Numerical analysis materials

The numerical materials contain seed-specific metrics, configuration summaries, paired contrasts, ranking and sensitivity outputs, selection results and figure coordinates. Scripts reproduce the tables and figures from these records. ETTh1 errors are archived to six decimal places; the other datasets retain additional precision. Sign classifications use the recorded values.

Table 11 Recorded deep-imputer training budgets (maximum epochs / early-stopping patience).

Dataset	Features	Train windows	SAITS epochs / patience	BRITS epochs / patience
ETTh1	7	128	24 / 8	20 / 6
KDD-Beijing	210	80	40 / 13	36 / 12
SWaT	40	61	32 / 10	28 / 9
Weather	21	385	24 / 8	20 / 6

Table 12 SAITS shared structural parameters.

Parameter	Value
`n_steps`	96
`n_layers`	2
`d_model`	256
`n_heads`	4
`d_k`, `d_v`	64, 64
`d_ffn`	128
`dropout`	0.1
Training / inference batch	32 / 256

Table 13 BRITS shared structural parameters.

Parameter	Value
`n_steps`	96
`rnn_hidden_size`	128
Training / inference batch	32 / 256
Training-window stride	96; append final boundary window

Table 14 Forecasting architectures used on all four datasets.

Model	Architecture
LSTM	Two layers, hidden size 64, no dropout; final hidden state mapped to all hC targets.
DLinear	Moving average of width 25 with replicated boundaries; independent trend and seasonal-residual linear maps for each feature, summed at the output.
PatchTST	RevIN with affine parameters and output denormalisation; patch length 16, stride 8, end padding (12 patches); two encoder layers, four heads, dimension 128, feed-forward width 512, dropout 0.1; shared channel-independent embedding and head.

Resampling and paired comparisons

All bootstrap and randomisation procedures use 20,000 draws and NumPy `RandomState(42)`. Inversion-proportion intervals resample whole dataset–forecaster–imputer–mechanism–rate groups, retaining their valid horizons. Each draw divides the sampled inversion count by the sampled valid count, accommodating groups with one to four valid horizons.

Forecaster comparisons match dataset, imputer, mechanism, rate and horizon; imputer comparisons match dataset, forecaster, mechanism, rate and horizon. They use the common valid intersection. Paired intervals resample whole matched groups with horizon omitted from the key. Two-sided randomisation tests flip all paired binary differences within a group, using the corrected p-value $(b + 1)/20{,}001$. Holm adjustment covers the six specified comparisons. Selection-rule comparisons retain all four horizons in 144 dataset–forecaster–mechanism–rate groups and apply the same bootstrap and sign-flip procedures; their seven tests form a separate Holm-adjusted family.

Dataset influence

Table 17 summarises changes in the pooled results after each dataset omission, retaining the validity threshold 0.01.

Appendix G Exploratory signal diagnostics

The analysis reuses 240 frozen imputation caches, producing 960 dataset–missingness–rate–horizon–imputer summaries and 2,304 non-Mean contrasts after matching forecasters. The main validity rule retains 2,127 contrasts. All 960 configurations reproduce archived IE-MAE and IE-MSE within the recorded numerical tolerances. No model is retrained; forecasting errors are averaged over the same three seeds, conditional on upstream seed 42.

Signal measures and support

For each feature in each eligible 96-step history, let R and $\widehat{R}_m$ be the mean absolute first differences of

Table 15 Forecasting settings shared across the three seeds and five imputers. Adam has zero weight decay and no learning-rate schedule. Gradient-norm clipping is 1.

Model	ETTh1 batch	KDD-Beijing batch	SWaT batch	Weather batch	Learning rate	Epochs / patience
LSTM	128	128	128	128	10^{-4}	100 / 20
DLinear	128	128	128	128	10^{-3}	100 / 20
PatchTST	128	32	128	128	10^{-4}	100 / 20

Table 16 Pure inversion by dataset, model and forecasting seed at $\epsilon = 0.01$. The valid count is the same across seeds within each row.

Dataset	Forecaster	Valid N	Seed 42	Seed 123	Seed 2026
ETTh1	LSTM	176	36.93%	39.77%	30.11%
ETTh1	DLinear	176	10.23%	10.23%	13.07%
ETTh1	PatchTST	176	12.50%	12.50%	13.64%
KDD-Beijing	LSTM	175	34.86%	40.00%	34.86%
KDD-Beijing	DLinear	175	10.29%	8.57%	9.71%
KDD-Beijing	PatchTST	175	34.29%	34.86%	34.29%
SWaT	LSTM	186	48.39%	39.25%	47.31%
SWaT	DLinear	186	5.38%	10.22%	9.14%
SWaT	PatchTST	186	15.59%	15.59%	13.44%
Weather	LSTM	172	27.91%	25.00%	23.26%
Weather	DLinear	172	0.00%	0.00%	0.00%
Weather	PatchTST	172	6.98%	5.23%	7.56%

Table 17 Leave-one-dataset-out influence check. Forecasting errors are averaged over three seeds before contrasts. No additional training is performed.

Excluded dataset	Valid N	Pure count	Pure inversion	Median τ
None	2127	415	19.51%	0.0449
ETTh1	1599	305	19.07%	0.0405
KDD-Beijing	1602	288	17.98%	0.0588
SWaT	1569	289	18.42%	0.0431
Weather	1611	363	22.53%	0.0394

the standardised truth and method m reconstruction. Eligible adjacent pairs have known truth at both endpoints, lie within the test segment and include at least one artificially masked cell. With at least eight such pairs, amplitude distortion is

$$d_{R,m} = \frac{|\widehat{R}_m - R|}{R + 10^{-6}}. \tag{G1}$$

Autocorrelation distortion is $d_{A,m} = |\widehat{\rho}_m - \rho|$, where ρ and $\widehat{\rho}_m$ are lag-1 Pearson correlations over genuinely adjacent pairs with known truth in the test segment. These pairs need not include a masked cell. At least eight pairs and endpoint standard deviations above 10^{-8} are required for truth and all five reconstructions; constant windows are excluded rather than assigned zero correlation. For each diagnostic, common eligible support is used across imputers. Window-level distortions are averaged within each feature and then equally across eligible features, retaining histories with an available full target horizon. Changes are $\Delta D = D_m - D_{\text{Mean}}$; negative values indicate closer recovery than Mean. Minimum eligible window–feature coverage across configurations is 30.94% for amplitude and 74.81% for autocorrelation.

Association and incremental prediction

Table 18 reports pooled Spearman associations with $\Delta\overline{\text{FE}}$. Percentile intervals use 2,000 bootstrap draws (seed 42), stratified by dataset and resampling dataset–missingness–rate groups with all their methods, forecasters and horizons retained. An ordinary least-squares diagnostic predicts $\Delta\overline{\text{FE}}$ from ΔIE and categorical dataset, forecaster, imputer, missingness, rate and horizon terms. Each signal change is added separately.

One-hot encoding and numeric standardisation are fitted only within training folds. Both models use the same five-fold GroupKFold partition of 48 dataset–missingness–rate groups. Pooled out-of-fold R^2 evaluates the configuration-level error-change regression.

Table 18 Exploratory signal diagnostics across 2,127 valid contrasts. Intervals are stratified group-bootstrap 95% intervals. Baseline out-of-fold R^2 is 0.2129; the last column is the change after adding one diagnostic.

Distortion change	Spearman ρ	95% CI	OOF ΔR^2
First-difference amplitude	0.192	[0.046, 0.344]	−0.0040
Lag-1 autocorrelation	0.295	[0.173, 0.410]	−0.0161

Sensitivity and stratified results

Autocorrelation associations are positive within each dataset ($\rho = 0.204$–0.365), whereas amplitude associations are approximately -0.071 for KDD-Beijing and SWaT despite the positive pooled estimate. Locally flat truth can inflate relative amplitude distortion. Using denominator constants 10^{-8}, 10^{-6} and 10^{-4} gives pooled correlations of 0.191, 0.192 and 0.215; restricting support to $R > 10^{-6}$ gives 0.223. Pooled correlations remain positive across these checks; raw amplitude distortion is sensitive to locally flat histories. Model-specific associations are strongest for DLinear/PatchTST, rather than LSTM. Only PatchTST's autocorrelation diagnostic increases out-of-fold R^2 (0.2746 to 0.2955) under this fixed partition. The two pooled diagnostics have positive rank associations but negative incremental out-of-fold R^2 (Table 18).

Reference proportions

Autocorrelation distortion improves in 403/415 pure inversions (97.11%), compared with 1,604/1,712 other valid contrasts (93.69%). Among the latter, 1,583 cases improve reconstruction without increasing forecasting error; 1,571 (99.24%) also improve autocorrelation distortion. Amplitude distortion improves in 345/415 (83.13%), 1,428/1,712 (83.41%) and 1,428/1,583 (90.21%), respectively. Signal-statistic improvements are common across all three groups.

References

[1] Ahn H, Sun K, Kim KP (2022) Comparison of missing data imputation methods in time series forecasting. Computers, Materials & Continua 70(1):767–779. https://doi.org/10.32604/cmc.2022.019369

[2] Boursalie O, Samavi R, Doyle TE (2022) Evaluation methodology for deep learning imputation models. Experimental Biology and Medicine 247(22):1972–1987. https://doi.org/10.1177/15353702221121602

[3] van Buuren S, Oberman HI (2026) Is the most likely value also the best imputation? Paediatric and Perinatal Epidemiology 40(2):278–280. https://doi.org/10.1111/ppe.70081, URL https://doi.org/10.1111/ppe.70081

[4] Cao W, Wang D, Li J, et al (2018) BRITS: Bidirectional recurrent imputation for time series. In: Advances in Neural Information Processing Systems, pp 6775–6785, https://doi.org/10.48550/arXiv.1805.10572

[5] Che Z, Purushotham S, Cho K, et al (2018) Recurrent neural networks for multivariate time series with missing values. Scientific Reports 8(1):6085. https://doi.org/10.1038/s41598-018-24271-9

[6] De Boor C (1978) A Practical Guide to Splines. Springer-Verlag, https://doi.org/10.1007/978-1-4612-6333-3

[7] Du W, Côté D, Liu Y (2023) SAITS: Self-attention-based imputation for time series. Expert Systems with Applications 219:119619. https://doi.org/10.1016/j.eswa.2023.119619

[8] Du W, Yang Y, Qian L, et al (2023) PyPOTS: A Python toolkit for machine learning on partially-observed time series. https://doi.org/10.48550/arXiv.2305.18811, URL https://arxiv.org/abs/2305.18811, arXiv:2305.18811

[9] Du W, Wang J, Qian L, et al (2024) TSI-Bench: Benchmarking time series imputation. arXiv preprint arXiv:2406.12747, https://doi.org/10.48550/arXiv.2406.12747

[10] Fortuin V, Baranchuk D, Raetsch G, et al (2020) GP-VAE: Deep probabilistic time series imputation. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol 108. PMLR, pp 1651–1661, URL https://proceedings.mlr.press/v108/fortuin20a.html

[11] Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: Continual prediction and LSTM. Neural Computation 12(10):2451–2471. https://doi.org/10.1162/089976600300015015

[12] Godahewa R, Bergmeir C, Webb GI, et al (2020) KDD cup dataset (with missing values). Zenodo, https://doi.org/10.5281/zenodo.4656719, URL https://doi.org/10.5281/zenodo.4656719

[13] Gomer B, Yuan KH (2023) A realistic evaluation of methods for handling missing data when there is a mixture of MCAR, MAR, and MNAR mechanisms in the same dataset. Multivariate Behavioral Research 58(5):988–1013. https://doi.org/10.1080/00273171.2022.2158776

[14] Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[15] Le Morvan M, Varoquaux G (2025) Imputation for prediction: beware of diminishing returns. In: International Conference on Learning Representations (ICLR), pp 41174–41213, URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/65e837e76a5308df3d5544aab6196e21-Abstract-Conference.html

[16] Le Morvan M, Josse J, Scornet E, et al (2021) What's a good imputation to predict with missing values? In: Advances in Neural Information Processing Systems, vol 34. Curran Associates, Inc., pp 11530–11540, https://doi.org/10.48550/arXiv.2106.00311

[17] Little RJA, Rubin DB (2019) Statistical Analysis with Missing Data, 3rd edn. John Wiley & Sons, https://doi.org/10.1002/9781119482260

[18] Mathur AP, Tippenhauer NO (2016) SWaT: A water treatment testbed for research and training on ICS security. In: 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater). IEEE, pp 31–36, https://doi.org/10.1109/cyswater.2016.7469060

[19] Näf J, Grzesiak K, Scornet E (2025) How to rank imputation methods? arXiv preprint arXiv:2507.11297, https://doi.org/10.48550/arXiv.2507.11297

[20] Niako N, Melgarejo JD, Maestre GE, et al (2024) Effects of missing data imputation methods on univariate blood pressure time series data analysis and forecasting with ARIMA and LSTM. BMC Medical Research Methodology 24(1):320. https://doi.org/10.1186/s12874-024-02448-3

[21] Nie Y, Nguyen NH, Sinthong P, et al (2023) A time series is worth 64 words: Long-term forecasting with transformers. In: International Conference on Learning Representations, https://doi.org/10.48550/arXiv.2211.14730

[22] Rubin DB (1976) Inference and missing data. Biometrika 63(3):581–592. https://doi.org/10.1093/biomet/63.3.581

[23] Saeyang S, Notodiputro K, Wanishsakpong W (2026) Analysis of imputation methods for missing not at random (MNAR) data: A comparative study of air pollution data in Bangkok, Thailand. Journal of Current Science and Technology 16(2):187. https://doi.org/10.59796/jcst.v16n2.2026.187

[24] Tashiro Y, Song J, Song Y, et al (2021) CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In: Advances in Neural Information Processing Systems, vol 34. Curran Associates, Inc., pp 24804–24816, https://doi.org/10.48550/arXiv.2107.03502

[25] Toye AA, Celik A, Kleinberg S (2025) Benchmarking missing data imputation methods for time series using real-world test cases.

In: Proceedings of the Sixth Conference on Health, Inference, and Learning, Proceedings of Machine Learning Research, vol 287. PMLR, pp 480–501, URL https://proceedings.mlr.press/v287/toye25a.html

[26] Troyanskaya O, Cantor M, Sherlock G, et al (2001) Missing value estimation methods for DNA microarrays. Bioinformatics 17(6):520–525. https://doi.org/10.1093/bioinformatics/17.6.520

[27] Twumasi-Ankrah S, Odoi B, Pels WA, et al (2019) Efficiency of imputation techniques in univariate time series. International Journal of Science, Environment and Technology 8(3):430–453. URL https://www.ijset.net/journal/2342.pdf

[28] Wang Z, Yang L, Sun L, et al (2024) Task-oriented time series imputation evaluation via generalized representers. In: Advances in Neural Information Processing Systems, pp 137403–137431, https://doi.org/10.52202/079017-4365

[29] Wani AA, Abeer F (2026) Comprehensive analysis of missing data imputation in clinical time-series: challenges, risks, and practical solutions. PeerJ Computer Science 12:e3848. https://doi.org/10.7717/peerj-cs.3848

[30] Wongoutong C (2025) Performance comparison of various imputation methods for missing data mechanisms (MAR, MCAR, and MNAR) in a nonstationary time series. International Journal of Mathematics and Mathematical Sciences 2025:3031708. https://doi.org/10.1155/ijmm/3031708

[31] Wu H, Xu J, Wang J, et al (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems 34:22419–22430. https://doi.org/10.48550/arXiv.2106.13008

[32] Zeng A, Chen M, Zhang L, et al (2023) Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 11121–11128, https://doi.org/10.1609/aaai.v37i9.26317

[33] Zhou H, Zhang S, Peng J, et al (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 11106–11115, https://doi.org/10.1609/aaai.v35i12.17325

[34] Zhou Y, Aryal S, Bouadjenek MR (2024) Review for handling missing data with special missing mechanism. arXiv preprint arXiv:2404.04905, https://doi.org/10.48550/arXiv.2404.04905