COMPONENT-AWARE IMPUTATION REFINEMENT FOR TIME SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series foundation models generally require imputing missing historical values before forecasting. Because imputers vary across series and missingness patterns, forecast quality depends on the initial imputer. We propose component-aware imputation refinement to reduce downstream forecast error and dependence on imputer choice. Starting from an existing imputation, the method applies robust STL decomposition and constructs a local trend bridge from gap boundaries, a seasonal reference from the median of observed same-phase history, and a residual reference. Window-level structural descriptors select the correction action and magnitude, while updates are restricted to originally missing positions. In paired evaluations across 18 datasets, 16 base imputers, and four zero-shot forecasters, mean sMAPE decreases by 3.70%, and the mean within-dataset-and-forecaster standard deviation across imputers decreases by 40.51%. The results indicate improved average forecast accuracy and reduced dependence on base-imputer choice within the evaluated scope.

1 INTRODUCTION

Time series foundation models acquire transferable forecasting capabilities through large-scale pretraining and perform zero-shot forecasting from the historical context of a target series (Das et al., 2024; Ansari et al., 2025). Missing historical values are typically imputed to form a complete input before a forecasting model predicts future values. Imputed values and original observations jointly determine the local trajectories, periodic variation, and fluctuations presented to the model. The choice of a base imputation method therefore matters for downstream forecasting.

Time series imputation offers a broad range of methods, including interpolation, statistical modeling, deep learning, and generative modeling. Several methods achieve strong performance in their respective settings (Cao et al., 2018; Du et al., 2023; Tashiro et al., 2021). This variety also makes selection difficult in practice. Imputation performance varies with data characteristics and missingness patterns, and the selected method affects downstream task performance (Du et al., 2024). Users thus face uncertainty about which method suits a particular forecasting task. Comparing candidates requires additional computation and evaluation, while adopting a readily deployable method may leave forecast quality dependent on the initial choice. Reducing dependence on careful selection of the base imputer consequently has practical value.

Post-imputation refinement offers a way to reduce this dependence on method selection. Outputs from different base imputers must combine with existing observations to form the historical context of a forecaster. Within a particular gap, an initial estimate may still deviate from the surrounding temporal structure. Figure 1 illustrates two constructed cases: attenuated periodic variation and a local level shift. Recurring variation and local trajectories in the surrounding history provide references for further refinement. These relationships motivate a reusable correction mechanism that operates on imputation outputs and can accompany different base imputation procedures.

We propose component-aware imputation refinement for time series foundation models, applying selective corrections to an existing imputation. The method decomposes the completed history into trend, seasonal, and residual components. Trend connections across gap boundaries and same-phase components at originally observed positions define references; window-level structural descriptors select corrections and control their magnitude. Updates are confined to originally missing positions, with the initial imputation retained as an available output. The method can follow different base

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

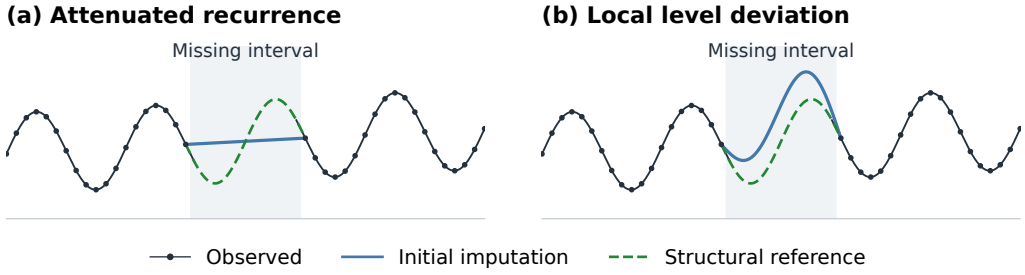


Figure 1: Constructed examples of two local structural deviations in an initial imputation. (a) Periodic variation is attenuated within the missing interval. (b) Estimates within the missing interval exhibit a local level shift. Black points denote observations, blue curves initial imputations, and green dashed curves illustrative references based on surrounding historical structure. Shading marks missing intervals. Both cases illustrate how local historical structure can provide references for post-imputation refinement. All curves are constructed examples; the horizontal axis represents historical time.

imputation procedures and supply the refined history directly to the target forecaster. Through this reusable refinement step, we aim to reduce the impact of the base imputation choice on forecast quality, allowing users to build forecasting workflows around existing or readily deployable imputers.

Our contributions are threefold. First, we make future forecast error the primary objective of imputation refinement and use paired forecast errors and between-imputer differences to assess prediction and selection dependence separately. Second, we develop one mechanism combining trend bridges, seasonal phase medians, and selective residual correction, and characterize its local updates. Third, complete evaluation across 18 datasets, 16 base imputers, 20 historical windows, and four zero-shot forecasters shows a 3.70% reduction in mean sMAPE and a 40.51% reduction in mean between-imputer standard deviation.

2 Related work

2.1 Time series imputation

Base imputers draw on different information: simple and univariate filling (Moritz & Bartz-Beielstein, 2017), conditional relations or neighboring samples (van Buuren & Groothuis-Oudshoorn, 2011; Troyanskaya et al., 2001), and temporal state dynamics (Kalman, 1960; Shumway & Stoffer, 1982). Their outputs can therefore differ even for the same missing history.

Learned imputers use recurrence, consistency, long-range context, attention, graphs, or low-rank spatiotemporal structure (Yoon et al., 2017; Shen et al., 2018; Cao et al., 2018; Liu et al., 2019; Bansal et al., 2021; Du et al., 2023; Cini et al., 2022; Nie et al., 2024). Adversarial, latent-variable, optimal-transport, and diffusion models instead represent uncertainty or generate candidate completions (Yoon et al., 2018; Mattei & Frellsen, 2019; Fortuin et al., 2020; Muzellec et al., 2020; Tashiro et al., 2021; Liu et al., 2023; Fons et al., 2025); MOMENT also supports masked-patch reconstruction (Goswami et al., 2024).

Benchmarks report variation across series and missingness conditions (Du et al., 2024; Kazijevs & Samad, 2023; Toye et al., 2025). This motivates our complementary setting: retain an already chosen imputer and apply the same post-imputation operator to its completed history, rather than replace the imputer or select a new one for each case.

2.2 TIME SERIES IMPUTATION FOR DOWNSTREAM TASKS

Imputation reconstruction error measures recovery of hidden historical values; our target is error on future forecasts. An imputed segment also changes the context seen by a predictor, so its numerical reconstruction quality need not rank imputers by predictive utility (Morvan et al., 2021). We therefore evaluate the forecast from the original completion against the forecast from its refined counterpart under identical conditions.

Task-aware work includes joint imputation and prediction (Cao et al., 2018), predictive-performance-guided repair (Krishnan et al., 2017), and combining imputations according to their effect on learning a forecaster from incomplete training targets (Wang et al., 2024). GRU-D and NeuMiss instead adapt the predictor to incomplete inputs (Che et al., 2018; Morvan et al., 2020). Our setting keeps a pretrained forecaster fixed: a single historical-input correction follows each base imputer, changes only originally missing positions, and is judged by downstream forecast error.

2.3 FOUNDATION MODELS FOR ZERO-SHOT TIME SERIES FORECASTING

Forecasting research has long represented trend, seasonality, frequency, patches, and variate structure within prediction models (Livera et al., 2011; Oreshkin et al., 2020; Wu et al., 2021; Zhou et al., 2022; Zeng et al., 2023; Wu et al., 2023; Nie et al., 2023; Liu et al., 2024a). Our decomposition serves a different role: it constructs references for refining a completed historical input before a fixed model forecasts.

Zero-shot forecasters arise from several pretraining designs (Ansari et al., 2024; Woo et al., 2024; Liu et al., 2024b; Garza et al., 2023), with cross-domain resources such as Monash and GIFT-Eval (Godahewa et al., 2021; Aksu et al., 2024). We evaluate one input-side operator with four fixed models: patch-based TimesFM, group-attention Chronos-2, distributional Sundial, and dynamically patched Kairos (Das et al., 2024; Ansari et al., 2025; Liu et al., 2025; Feng et al., 2025). Their distinct forecasting designs test whether the same refinement can be useful beyond one predictor; none participates in computing the correction.

3 PROBLEM FORMULATION

3.1 PROBLEM SETTING

Let $x \in \mathbb{R}^L$ denote a historical sequence of length L preceding the forecast origin, and let $y \in \mathbb{R}^H$ denote the target over the next H time steps. The missingness mask $M \in \{0,1\}^L$ equals one at missing positions and zero at observed positions, defining $O = \{i : M_i = 0\}$ and $G = \{i : M_i = 1\}$. Given historical observations x_O and mask M, a base imputer I produces a complete sequence $z = I(x_O, M)$, with $z_i = x_i$ for $i \in O$. The initial estimates z_G are the quantities to be refined.

Refinement operates within the historical window and can use all observations preceding the forecast origin, including observations on both sides of an internal gap. Hidden historical values x_G and future targets y do not enter the refinement computation; y is used to evaluate forecast error after prediction.

Let F_m be the forecasting map of the target time series foundation model indexed by m, and let A_ϕ be a refinement operator with configuration ϕ. Given period parameter p, the initial forecast $\widehat{y}^0$, refined history z^+, and refined forecast $\widehat{y}^+$ are defined as

$$\widehat{y}^0 = F_m(z), \qquad z^+ = A_\phi(z, M, p; m), \qquad \widehat{y}^+ = F_m(z^+). \tag{1}$$

3.2 IMPUTATION REFINEMENT OBJECTIVE

The primary objective of refinement is lower future forecast error for the target forecaster. Given completed history z from a base imputer I, the operator changes only missing entries G and preserves $z_i^+ = z_i$ for every $i \in O$. The same rule follows different base imputers. If both their mean forecasting error and between-imputer differences decline, the forecasting pipeline depends less on

the initial choice. With forecasting loss ℓ_{pred} and operator A_ϕ under fixed configuration ϕ, define the risk difference relative to the unrefined input:

$$\mathcal{R}_m(\phi; I) = \mathbb{E}[\ell_{\text{pred}}(F_m(A_\phi(z, M, p; m)), y)], \qquad \Delta\mathcal{R}_m(\phi; I) = \mathcal{R}_m(\phi; I) - \mathcal{R}_m(\text{id}; I). \quad (2)$$

Here $A_{\text{id}}(z, M, p; m) = z$, the expectation covers evaluation cases, and F_m returns a P50 point forecast under the experimental protocol. A negative risk difference means lower forecasting error. We neither minimize this risk for an individual case nor use future targets to train the correction; Section 4 specifies a fixed rule from historical structure. The experiments separately measure dispersion across base-imputer errors within each dataset–forecaster cell to assess selection dependence, without optimizing that dispersion at inference time.

4 Component-aware imputation refinement

4.1 Method overview

One refinement operator processes each channel from finite completed history z, the original mask M, an externally supplied period p, and target model identifier m. The period determines STL decomposition and same-phase search; the model identifier only selects fixed trend-enhancement thresholds. Decisions use the current history, not future targets or forecaster states. Appendix B collects exact numerical and boundary conventions.

Figure 2 organizes the computation in three stages. First, trend, seasonal, and residual references are built from the initial imputation. Next, descriptors of the current window select retention or a correction action. Finally, component strengths turn the selected directions into increments written only at originally missing entries. Reference construction supplies possible directions, the gate chooses which to use, and strength control sets their magnitudes.

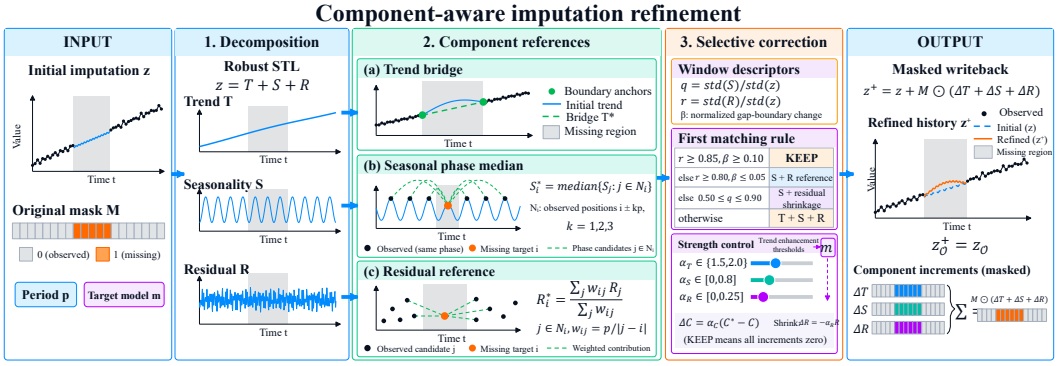


Figure 2: Framework of component-aware imputation refinement. Robust STL decomposes the initial imputation z to construct trend bridges, seasonal phase medians, and weighted residual references. Window descriptors select an ordered action branch, component coefficients control update magnitudes, and masked writeback produces the refined history z^+ by modifying only originally missing entries. Model identifier m configures trend-enhancement thresholds. Gray shading marks missing intervals, and curves are schematic.

Robust STL decomposes the initial imputation into trend, seasonal, and residual components (Cleveland et al., 1990; Hyndman & Athanasopoulos, 2021). Two-sided trend anchors and same-phase components at originally observed positions supply candidate references; residual shrinkage uses a zero target. References depend on the initial decomposition, and window descriptors guide action selection under assumptions of local continuation and phase recurrence. The action set includes retaining the input.

For $z = T + S + R$, the selected action specifies component increments $(\Delta_T, \Delta_S, \Delta_R)$. Retaining the initial imputation returns the input unchanged; the other actions modify the selected components at missing positions. The complete update is

$$z^+ = z + M \odot (\Delta_T + \Delta_S + \Delta_R). \tag{3}$$

4.2 COMPONENT REFERENCES

Robust STL with supplied period p decomposes the completed history into T, S, R (Hyndman & Athanasopoulos, 2021). This separates local level changes, repeated structure, and remaining variation. All three are estimates from z; a component at an observed position can still depend on neighboring imputed values. We use robust fitting with seasonal span 7; Appendix B gives the remaining smoother and iteration settings.

Trend reference. An interior gap has historical support at both boundaries. We use the adjacent decomposed trend values as anchors for a local bridge. Figure 6(a) illustrates the reference. For gap $[a, b]$ of length $n = b - a + 1$,

$$T^*_{a+u-1} = T_{a-1} + \frac{u}{n+1}(T_{b+1} - T_{a-1}), \qquad u = 1, \ldots, n. \tag{4}$$

Prefix and suffix gaps lack anchors on both sides, so we retain $T^*_i = T_i$. For an interior gap, the bridge uses the two bordering trend values to propose a local continuation. This assumption is useful when the endpoints represent the low-frequency trajectory within the gap; unobserved turning points in a long gap can reduce reference accuracy.

Phase references. For a missing position i, we inspect three cycles in each direction and retain originally observed, same-phase positions within the historical window:

$$N_i = \{j = i \pm kp : k \in \{1, 2, 3\},\ 1 \le j \le L,\ M_j = 0\}. \tag{5}$$

$$S^*_i = \mathrm{median}_{j \in N_i} S_j, \qquad R^*_i = \frac{\sum_{j \in N_i} w_{ij} R_j}{\sum_{j \in N_i} w_{ij}}, \quad w_{ij} = \frac{p}{|j - i|}. \tag{6}$$

An empty N_i leaves the corresponding reference equal to the original component. The seasonal median limits the influence of isolated candidate outliers; with an even number of candidates, the two central values are averaged. Residual references use distance weights to emphasize nearer cycles. Seasonal correction applies in every non-retention branch, while the selected action updates residuals toward their reference or shrinks them toward zero. All references are computed from the initial decomposition.

The three references serve different roles. Trend bridging proposes low-frequency continuation through a gap; the seasonal median supplies a typical repeated phase; the residual reference preserves local variation consistent with nearby cycles. They are candidate update directions derived from current history, not guaranteed estimates of hidden truth. When boundary anchors or originally observed same-phase points are unavailable, the corresponding reference retains the original component, and the gate may still retain the whole input.

4.3 ACTION SELECTION

Action selection operates at the window level, whereas writeback remains pointwise at missing entries. Window descriptors ask whether seasonal or residual variation is substantial and whether the initial imputation connects smoothly to gap boundaries. Gaps in a window therefore share an action, while each missing entry retains its own trend and phase references.

$$q = \frac{\mathrm{std}(S)}{\sigma}, \qquad r = \frac{\mathrm{std}(R)}{\sigma}. \tag{7}$$

Ratios q, r are computed from the current window and used for action selection. STL components can be correlated, so either ratio can exceed one and their sum varies across windows.

Partition G into maximal contiguous gaps $[a, b]$. Each available left boundary contributes $|z_a - z_{a-1}|$, and each available right boundary contributes $|z_{b+1} - z_b|$. For the multiset B of these contributions, define

$$\beta = \frac{1}{|B|\sigma} \sum_{u \in B} u, \qquad \beta = 0 \text{ if } B = \varnothing. \tag{8}$$

The average covers all available boundaries and describes changes at gap connections relative to the window scale. Interpretation of a boundary change depends on local structure, so the gate uses β together with component amplitudes.

Actions are selected in order: retain the input if $r \geq 0.85$ and $\beta \geq 0.10$; otherwise use seasonal and residual references if $r \geq 0.80$ and $\beta \leq 0.05$; otherwise correct seasonality and shrink residuals if $0.50 \leq q \leq 0.90$; in the remaining cases, correct trend, seasonality, and residuals. The first satisfied condition determines the action. Table 3 specifies the component increments.

The ordered gate first retains high-residual windows with large boundary changes. High-residual windows with smoother boundaries use seasonal and residual references; intermediate seasonal amplitude retains trend while correcting seasonality and shrinking residuals; remaining windows use all three components. Retention takes precedence over updates, so later conditions cannot overwrite an input already judged unsuitable for correction. Thresholds are fixed; Appendices A and B give the exact values and trend-enhancement conditions.

4.4 Correction magnitudes

After action selection, seasonal and residual strengths are computed from component amplitudes and boundary variation, lying in $[0, 0.8]$ and $[0, 0.25]$. They move components toward their references, or move the residual toward zero in the shrinkage branch. Trend bridging uses strength 1.5, rising to 2.0 when boundary variation and both reference discrepancies meet fixed thresholds. TimesFM uses stricter thresholds only for this enhancement decision; the other branches and component formulas are shared. Appendix B gives exact formulas and numerical conventions.

Seasonal and residual strengths do not exceed one, so reference updates interpolate between the original component and its reference. Trend strength exceeds one and can move beyond the bridge. This design makes repeated-structure corrections gradual while allowing stronger low-frequency correction under the gate. Extrapolation can also amplify an inaccurate reference, so enhancement requires the three-component action and additional thresholds. Appendix C characterizes local update properties; paired forecasting experiments determine predictive effects.

4.5 Computation

Algorithm 1 gives the execution skeleton. All references come from one initial decomposition, and the window gate acts before writeback. KEEP returns the initial imputation; other actions use the component increments in Appendix Table 3. Missing entries are written into a copy of the input, so observed entries cannot change through component reconstruction. Appendices A and B contain neighbor search, gate pseudocode, and boundary cases.

For robust-decomposition cost $C_{\mathrm{STL}}(L, p)$, total time is $C_{\mathrm{STL}}(L, p) + O(L + K|G|)$ and space is $O(L)$. Here $K = 3$: each missing position has at most six phase candidates. Feature and gap processing use linear scans. Refinement adds no internal forecasting calls.

5 Experiments

5.1 Experimental setup

We evaluate 16 registered historical base-imputation implementations across 18 datasets, covering simple filling, interpolation, state-space and Gaussian-process methods, MICE, and existing MOMENT, SAITS, and NuwaTS implementations. The forecasters are Chronos-2, TimesFM

Algorithm 1 Component-aware imputation refinement

Input: completed history z, original mask M, period p, model identifier m.
Output: refined history z^+ with $z^+_O = z_O$.

$G \leftarrow \{i : M_i = 1\}$; $z^+ \leftarrow \mathrm{copy}(z)$.
if $G = \varnothing$ **then**
    return z^+
end if
$(T, S, R) \leftarrow \mathrm{RobustSTL}(z, p)$.
$(T^*, S^*, R^*) \leftarrow \mathrm{References}(T, S, R, M, p)$.
$(q, r, \beta, d_T, d_R) \leftarrow \mathrm{Descriptors}(z, M, T, S, R, T^*, R^*)$.
$(a, \alpha_T, \alpha_S, \alpha_R) \leftarrow \mathrm{SelectActionAndStrength}(q, r, \beta, d_T, d_R, m)$.
if $a = \mathrm{KEEP}$ **then**
    return z^+
end if
$(\Delta_T, \Delta_S, \Delta_R) \leftarrow \mathrm{ComponentUpdates}(a, T, S, R, T^*, S^*, R^*, \alpha_T, \alpha_S, \alpha_R)$.
$z^+_G \leftarrow z_G + \Delta_{T,G} + \Delta_{S,G} + \Delta_{R,G}$
return z^+

2.5, Kairos-50M, and Sundial. The original first 20 historical windows per dataset, five block-missingness patterns, and nominal ratios of 10%, 20%, and 30% produce 5,400 scenes and 86,400 base imputations. Each imputation is forecast directly and after refinement under all four models, yielding 345,600 paired comparisons and 691,200 logical prediction records with no technical failures. Appendix D gives implementation details.

Each comparison fixes the scene, base imputation, forecaster, random state, and horizon, changing only whether refinement is applied. Point forecasts are P50, using the sample median for sample-based models. Horizons are 48 for minute/hour data, 30 for daily data, and 8 for weekly data; contexts are at most 2048 steps. Formal inference uses batches of 32 inputs with the same inference contract on both sides. The primary metric is sMAPE, averaged over 300 scenes within each dataset–forecaster–imputer cell and then equally across cells. MAE and RMSE use the mean absolute first difference of consecutive observed historical values in the same scene for normalization.

The experiments answer three connected questions. Does refinement lower future forecast error under fixed inputs and inference protocols (Section 5.2)? Does it narrow the error gap among base imputers without choosing a new winner (Section 5.3)? Do the aggregate changes persist across the evaluated missingness conditions (Section 5.4)? Every pair starts from the same completed history; only the originally missing entries differ after refinement.

5.2 Downstream forecasting performance

Table 1 answers the first question. Overall mean sMAPE decreases from 5.2593 to 5.0645, a reduction of 0.1948 points or 3.70%. All four fixed forecasters show lower means, with relative changes from 3.14% for Sundial to 4.50% for Chronos-2. Because each row averages the same paired scenes and imputers before and after refinement, the change reflects the input correction under that forecaster rather than a switch of forecasting model.

Table 1: Mean sMAPE by forecaster; 86,400 pairs per model and 345,600 pairs overall.

Forecaster	Base	Refined	Gain
Chronos-2	5.1801	4.9468	4.50%
TimesFM 2.5	5.1777	4.9948	3.53%
Kairos-50M	5.2243	5.0324	3.67%
Sundial	5.4552	5.2842	3.14%
Overall	5.2593	5.0645	3.70%

The direction agrees with two secondary error measures: fixed-history-scale-normalized MAE falls from 4.8137 to 4.6968 (2.43%) and RMSE from 5.8371 to 5.7069 (2.23%). The primary conclusion

rests on sMAPE; these measures show that the mean reduction is not specific to one error definition. Appendix E gives scene-level change counts and grouped results.

The reduction is also visible below the model-level average: all 16 imputer marginal means and all 64 forecaster–imputer cell means decline (Appendix Table 14). This is a consistency statement about grouped means, not a claim that every individual forecast improves. The size of the reduction varies across models and imputers, which motivates examining selection dependence separately.

5.3 Dependence on base imputer selection

Figure 3(a) places each imputer's original and refined mean sMAPE side by side; larger shifts for some higher-error starting methods bring the plotted means closer together. To quantify the second question, we compute the population standard deviation and range across 16 imputer means within each dataset–forecaster cell, then average over 72 such cells. Figure 3(b) shows standard deviation falling from 0.4571 to 0.2719 (40.51%) and range from 1.7368 to 1.0601 (38.96%). This is reduced sensitivity of aggregate forecast error to which registered imputer supplied the initial history; it does not measure a user's imputer-search cost or imply equal performance for all imputers. Appendix E reports the model-specific values.

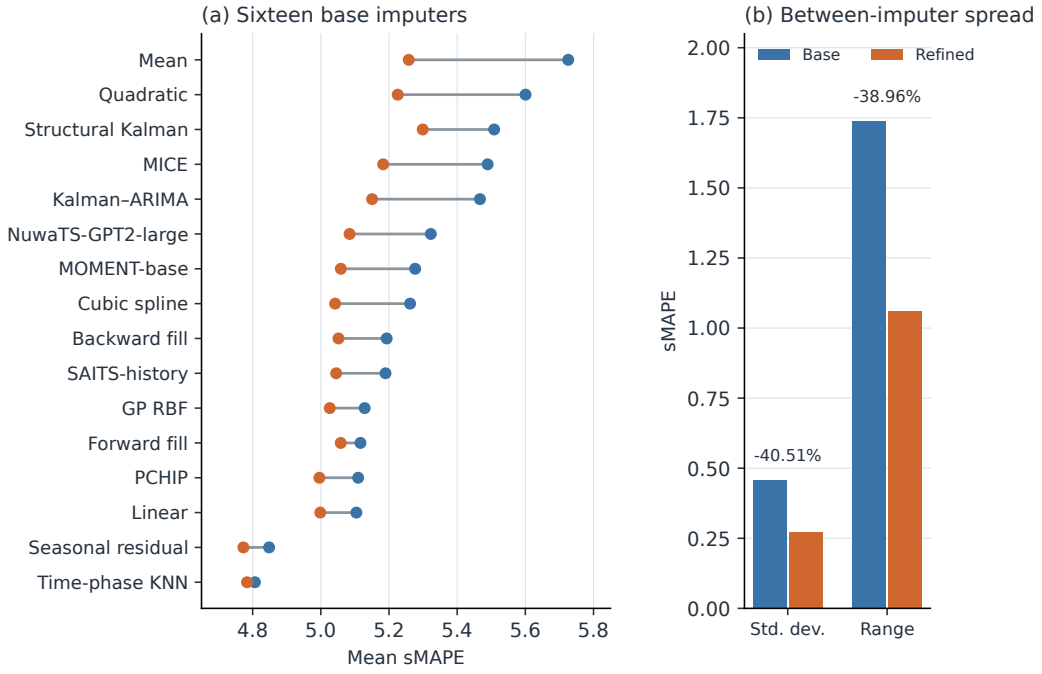


Figure 3: Mean sMAPE for 16 imputers (left): each pair places the refined point on the left and the base point on the right. Mean between-imputer standard deviation and range across 72 dataset–forecaster cells (right): the base and refined bars are the left and right bars in each pair. Blue denotes base and orange denotes refined.

5.4 Performance across missingness conditions

Table 2 and Figure 4 answer the third question: the table gives relative reductions and the paired bars show absolute means on the same scenes. Mean sMAPE declines under each of the five patterns and all three nominal ratios. Pattern-level relative reductions range from 1.89% for periodic missingness to 4.98% for change-region missingness; ratio-level reductions are 1.66%, 3.46%, and 5.69% at 10%, 20%, and 30% missingness. Appendix E.3 contains full values.

Table 2: Relative mean-sMAPE reductions by missingness condition.

Condition	Relative reduction by group
Pattern	Random 3.03%; extreme 4.75%; change 4.98%; volatility 3.70%; periodic 1.89%
Ratio	10%: 1.66%; 20%: 3.46%; 30%: 5.69%

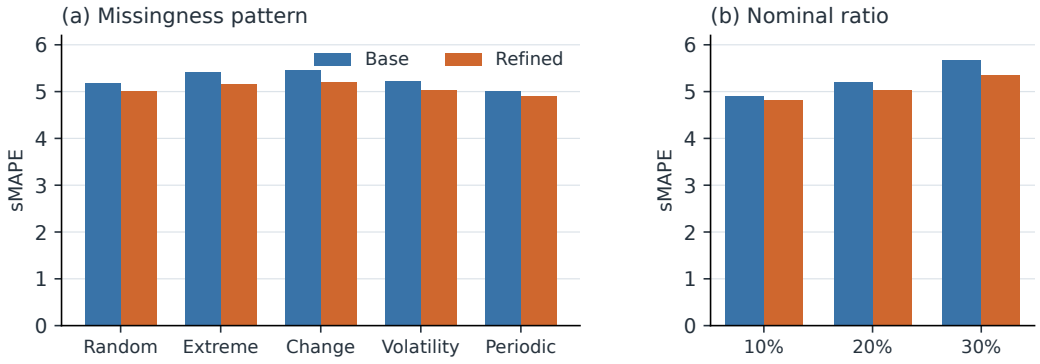


Figure 4: Mean sMAPE by missingness pattern and nominal ratio. Base and refined bars use the same scenes within each group; the base bar is on the left and the refined bar on the right.

6 Conclusion

This work addressed one forecasting question: after a base imputer has supplied a complete historical context, can a fixed correction improve forecasts while reducing dependence on which imputer was chosen? Our component-aware operator uses robust decomposition to form a gap-boundary trend bridge and observed same-phase seasonal and residual references. Window descriptors select an action and its strength, and only originally missing entries are written back. The resulting history can be used by an unchanged time series foundation model.

The complete paired panel contains 345,600 comparisons across 18 datasets, 16 base imputers, 20 historical windows, and four forecasters. Mean sMAPE declines by 3.70%, while mean between-imputer standard deviation within dataset–forecaster cells declines by 40.51%. The first result addresses forecast accuracy; the second addresses selection dependence. The effects vary by forecaster, imputer, and missingness condition, and the aggregate reductions do not imply improvement for every input.

The practical implication is a reusable step after an existing or readily deployable imputer, without future labels, case-specific training, or changing the forecaster. Its references require informative local boundaries and observed same-phase history; when these are weak, the gate may retain the original completion. This distinction between a proposed correction and its measured forecasting effect keeps the method applicable as an input-side refinement rather than a replacement for imputation or forecasting.

AI use statement

Generative AI assisted with discussion of the research method and experimental design; implementation and inspection of research code; checking and interpreting experimental results; manuscript organization, drafting, translation, and mathematical exposition; literature search and reference formatting; and figure generation, revision, plotting, and document-building code. Method illustrations are constructed examples; empirical tables and figures are calculated from stored prediction records. The authors take responsibility for the final manuscript and its claims.

REPRODUCIBILITY STATEMENT

Section 3 defines the problem and Section 4 specifies the fixed operator. Appendices A–C provide pseudocode, parameters, and properties; Appendices D and E document data settings, inference, metrics, and grouped results for the complete panel. All tables use the same paired records, reporting zero technical failures and all gains, deteriorations, and equal outcomes.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A Benchmark For General Time Series Forecasting Model Evaluation. *arXiv preprint arXiv:2410.10393*, 2024. URL https://arxiv.org/abs/2410.10393.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2403.07815.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From Univariate to Universal Forecasting. *arXiv preprint arXiv:2510.15821*, 2025. URL https://arxiv.org/abs/2510.15821.

Parikshit Bansal, Prathamesh Deshpande, and Sunita Sarawagi. Missing Value Imputation on Multidimensional Time Series. *Proceedings of the VLDB Endowment, 14(11):2533–2545*, 2021. URL https://vldb.org/pvldb/vol14/p2533-bansal.pdf.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems 31*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/734e6bfcd358e25ac1db0a4241b95651-Abstract.html.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports, 8:6085*, 2018. URL https://www.nature.com/articles/s41598-018-24271-9.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=kOu3-S3wJ7.

Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics, 6(1):3–73*, 1990. URL https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/stl-a-seasonal-trend-decomposition-procedure-based-on-loess.pdf.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning, PMLR 235:10148–10167*, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Wenjie Du, David Côté, and Yan Liu. SAITS: Self-Attention-based Imputation for Time Series. *Expert Systems with Applications, 219:119619*, 2023. URL https://arxiv.org/abs/2202.08516.

Wenjie Du, Jun Wang, Linglong Qian, Yiyuan Yang, Zina Ibrahim, Fanxing Liu, Zepu Wang, Haoxin Liu, Zhiyuan Zhao, Yingjie Zhou, Wenjia Wang, Kaize Ding, Yuxuan Liang, B. Aditya Prakash, and Qingsong Wen. TSI-Bench: Benchmarking Time Series Imputation. *arXiv preprint arXiv:2406.12747*, 2024. URL https://arxiv.org/abs/2406.12747.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Sihan Lu, Shuqi Gu, Lintao Ma, Xingyu Lu, and Kan Ren. Kairos: Toward Adaptive and Parameter-Efficient Time Series Foundation Models. *arXiv preprint arXiv:2509.25826*, 2025. URL https://arxiv.org/abs/2509.25826.

Elizabeth Fons, Alejandro Sztrajman, Yousef El-Laham, Luciana Ferrer, Svitlana Vyetrenko, and Manuela Veloso. LSCD: Lomb–Scargle Conditioned Diffusion for Time series Imputation. In *Proceedings of the 42nd International Conference on Machine Learning, PMLR 267:17411–17436*, 2025. URL https://proceedings.mlr.press/v267/fons25a.html.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Raetsch, and Stephan Mandt. GP-VAE: Deep Probabilistic Time Series Imputation. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR 108:1651–1661*, 2020. URL https://proceedings.mlr.press/v108/fortuin20a.html.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. TimeGPT-1. *arXiv preprint arXiv:2310.03589*, 2023. URL https://arxiv.org/abs/2310.03589.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash Time Series Forecasting Archive. In *Advances in Neural Information Processing Systems 34, Datasets and Benchmarks Track*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/eddea82ad2755b24c4e168c5fc2ebd40-Paper-round2.pdf.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A Family of Open Time-series Foundation Models. In *Proceedings of the 41st International Conference on Machine Learning, PMLR 235:16115–16152*, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 3rd edition, 2021. URL https://otexts.com/fpp3/.

Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting, 22(4):679–688*, 2006. URL https://www.sciencedirect.com/science/article/pii/S0169207006000239.

Rudolf E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering, 82(1):35–45*, 1960. URL https://doi.org/10.1115/1.3662552.

Maksims Kazijevs and Manar D. Samad. Deep imputation of missing values in time series health data: A review with benchmarking. *Journal of Biomedical Informatics, 144:104440*, 2023. URL https://www.sciencedirect.com/science/article/pii/S1532046423001612.

Sanjay Krishnan, Michael J. Franklin, Ken Goldberg, and Eugene Wu. BoostClean: Automated Error Detection and Repair for Machine Learning. *arXiv preprint arXiv:1711.01299*, 2017. URL https://arxiv.org/abs/1711.01299.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. PriSTI: A Conditional Diffusion Framework for Spatiotemporal Imputation. In *Proceedings of the 39th IEEE International Conference on Data Engineering, pp. 1927–1939*, 2023. URL https://doi.org/10.1109/ICDE55515.2023.00150.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *Proceedings of the 41st International Conference on Machine Learning, PMLR 235:32369–32399*, 2024b. URL https://proceedings.mlr.press/v235/liu24cb.html.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A Family of Highly Capable Time Series Foundation Models. *arXiv preprint arXiv:2502.00816*, 2025. URL https://arxiv.org/abs/2502.00816.

Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. NAOMI: Non-Autoregressive Multiresolution Sequence Imputation. In *Advances in Neural Information Processing Systems 32*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/50c1f44e426560f3f2cdcb3e19e39903-Abstract.html.

Alysha M. De Livera, Rob J. Hyndman, and Ralph D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association, 106(496):1513–1527*, 2011. URL https://robjhyndman.com/publications/complex-seasonality/.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets. In *Proceedings of the 36th International Conference on Machine Learning, PMLR 97:4413–4423*, 2019. URL https://proceedings.mlr.press/v97/mattei19a.html.

Steffen Moritz and Thomas Bartz-Beielstein. imputeTS: Time Series Missing Value Imputation in R. *The R Journal, 9(1):207–218*, 2017. URL https://journal.r-project.org/articles/RJ-2017-009/.

Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gael Varoquaux. NeuMiss networks: differentiable programming for supervised learning with missing values. In *Advances in Neural Information Processing Systems 33*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/42ae1544956fbe6e09242e6cd752444c-Abstract.html.

Marine Le Morvan, Julie Josse, Erwan Scornet, and Gael Varoquaux. What's a good imputation to predict with missing values? In *Advances in Neural Information Processing Systems 34*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5fe8fdc79ce292c39c5f209d734b7206-Abstract.html.

Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing Data Imputation using Optimal Transport. In *Proceedings of the 37th International Conference on Machine Learning, PMLR 119:7130–7140*, 2020. URL https://proceedings.mlr.press/v119/muzellec20a.html.

Tong Nie, Guoyang Qin, Wei Ma, Yuewen Mei, and Jian Sun. ImputeFormer: Low Rankness-Induced Transformers for Generalizable Spatiotemporal Imputation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2260–2271*, 2024. URL https://doi.org/10.1145/3637528.3671751.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcoL.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Lifeng Shen, Qianli Ma, and Sen Li. End-to-End Time Series Imputation via Residual Short Paths. In *Proceedings of the 10th Asian Conference on Machine Learning, PMLR 95:248–263*, 2018. URL https://proceedings.mlr.press/v95/shen18a.html.

Robert H. Shumway and David S. Stoffer. An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm. *Journal of Time Series Analysis, 3(4):253–264*, 1982. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9892.1982.tb00349.x.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Information Processing Systems 34*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cfe8504bda37b575c70ee1a8276f3486-Abstract.html.

Adedolapo Aishat Toye, Asuman Celik, and Samantha Kleinberg. Benchmarking Missing Data Imputation Methods for Time Series Using Real-World Test Cases. In *Proceedings of the 6th Conference on Health, Inference, and Learning, PMLR 287:480–501*, 2025. URL https://proceedings.mlr.press/v287/toye25a.html.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics, 17(6):520–525*, 2001. URL https://academic.oup.com/bioinformatics/article/17/6/520/272365.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software, 45(3):1–67*, 2011. URL https://www.jstatsoft.org/v45/i03/.

Zhixian Wang, Linxiao Yang, Liang Sun, Qingsong Wen, and Yi Wang. Task-oriented Time Series Imputation Evaluation via Generalized Representers. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f88264fcc54775ee1706116e90fe351a-Abstract-Conference.html.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified Training of Universal Time Series Forecasting Transformers. In *Proceedings of the 41st International Conference on Machine Learning, PMLR 235:53140–53164*, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems 34*, 2021. URL https://papers.neurips.cc/paper_files/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02186.

Jinsung Yoon, William R. Zame, and Mihaela van der Schaar. Estimating Missing Data in Temporal Data Streams Using Multi-directional Recurrent Neural Networks. *arXiv preprint arXiv:1711.08742*, 2017. URL https://arxiv.org/abs/1711.08742.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing Data Imputation using Generative Adversarial Nets. In *Proceedings of the 35th International Conference on Machine Learning, PMLR 80:5689–5698*, 2018. URL https://proceedings.mlr.press/v80/yoon18a.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence, 37(9):11121–11128*, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of the 39th International Conference on Machine Learning, PMLR 162:27268–27286*, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A REFERENCE CONSTRUCTION AND ACTION SELECTION

Algorithm 2 Observed phase-neighbor reference

Input: Seasonal and residual components S, R, mask M, period p; $K = 3$.
Initialize $S^* \leftarrow S$, $R^* \leftarrow R$.
For each $i \in G$, collect observed same-phase indices N_i using Eq. 5.
If $N_i \neq \varnothing$: $S_i^* \leftarrow \mathrm{median}_{j \in N_i} S_j$.
If $N_i \neq \varnothing$: $R_i^* \leftarrow \sum_{j \in N_i} w_{ij} R_j / \sum_{j \in N_i} w_{ij}$.
Return (S^*, R^*).

References are computed from the initial decomposition before any writeback. A previously updated missing entry never becomes a trusted candidate for another update, so neighbor construction is independent of missing-index traversal order. The trend reference also starts as a copy of T; only gaps with two available boundary anchors receive a bridge.

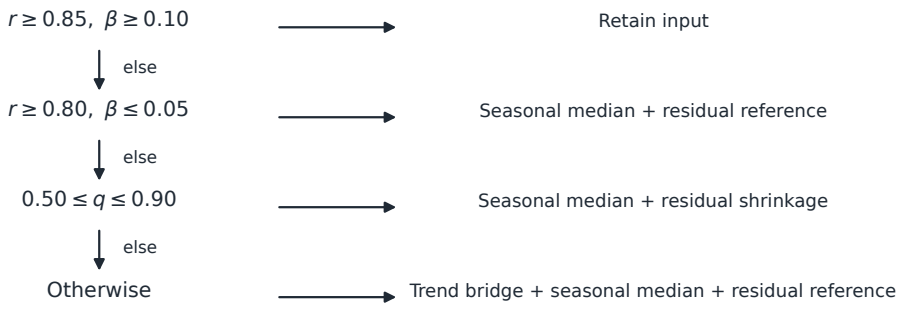


Figure 5: Ordered action selection. All non-retention branches apply seasonal median correction; the branch determines trend correction and residual treatment.

Algorithm 3 Ordered gate and trend enhancement

If $r \geq 0.85$ and $\beta \geq 0.10$: $a \leftarrow$ KEEP.
Else if $r \geq 0.80$ and $\beta \leq 0.05$: $a \leftarrow$ SR.
Else if $0.50 \leq q \leq 0.90$: $a \leftarrow$ SHRINK.
Else: $a \leftarrow$ TR.
Set $\alpha_T = 1.5$ and $(\tau_\beta, \tau_T, \tau_R) = (0.05, 0.03, 0.035)$.
For a TimesFM model prefix, replace thresholds by $(0.10, 0.05, 0.06)$.
For the three-component action only, if $\beta \geq \tau_\beta$, $d_T \geq \tau_T$, and $d_R \geq \tau_R$:
    Set $\alpha_T = 2.0$.
Return (a, α_T).

Table 3: Component updates for each refinement action. Zero denotes no change to a component; it does not set the component to zero. Updates are written only at originally missing positions.

Refinement action	**Trend** Δ_T	**Seasonal** Δ_S	**Residual** Δ_R
Keep initial imputation	0	0	0
Seasonal–residual	0	$\alpha_S(S^* - S)$	$\alpha_R(R^* - R)$
Seasonal + shrinkage	0	$\alpha_S(S^* - S)$	$-\alpha_R R$
Three components	$\alpha_T(T^* - T)$	$\alpha_S(S^* - S)$	$\alpha_R(R^* - R)$

B PARAMETERS AND BOUNDARY CONVENTIONS

Strength control additionally measures how far the current trend and residuals lie from their references at missing positions:

$$d_T = \frac{\sum_{i \in G} |T_i^* - T_i|}{|G|\sigma}, \qquad d_R = \frac{\sum_{i \in G} |R_i^* - R_i|}{|G|\sigma}. \tag{9}$$

When G is empty, set both discrepancies to zero. Quantities d_T, d_R measure the mean distances of the current trend and residual components from their respective references, providing window-level inputs to trend enhancement.

Let $v = \min(1, q/\max(r, 0.01))$. For $q > 0.01$, set $\alpha_S = \mathrm{round}_2(0.4 + 0.4v)$; otherwise set $\alpha_S = 0$. Define $b_S = 0.15v$ if $r > 0.01$, and $b_S = 0$ otherwise. The residual coefficient is

$$\alpha_R = \mathrm{round}_2\big(\mathrm{clip}(0.1 + b_S - \min(0.2\beta, 0.1), 0, 0.3)\big). \tag{10}$$

The seasonal contribution increases the weight on residual references, while the boundary term decreases it. Dynamic coefficients are rounded to two decimal places. The explicit clipping bound is 0.3; the formula actually yields at most 0.25 because $b_S \le 0.15$. Both reference-based residual updates and zero-target shrinkage use this same coefficient.

The trend coefficient is $\alpha_T = 1.5$. Under the three-component action, it increases to 2.0 when $\beta \ge \tau_\beta(m)$, $d_T \ge \tau_T(m)$, and $d_R \ge \tau_R(m)$ all hold. Default thresholds are $(0.05, 0.03, 0.035)$; identifiers with a TimesFM prefix use $(0.10, 0.05, 0.06)$. The TimesFM configuration therefore requires larger boundary changes and reference discrepancies before enhancement. These three empirical thresholds control the triggering region for model-specific trend enhancement; the remaining action rules and strength formulas are shared.

For $\alpha_T > 1$, the update extends beyond the reference along the bridge direction. For example, $\alpha_T = 1.5$ gives $T^+ = 1.5T^* - 0.5T$ at missing positions. A stronger update also amplifies reference errors; Section C analyzes this behavior.

Amplitude ratios q, r and discrepancies d_T, d_R are computed for each window; the neighborhood radius $K = 3$ and gate thresholds are prescribed hyperparameters. Coefficients α_S, α_R follow the feature-dependent formulas, and α_T takes 1.5 or 2.0. Scale fallback applies when the population standard deviation is exactly zero. Coefficients use NumPy rounding to two decimal places.

Table 4: Fixed thresholds and coefficient conventions.

Quantity	**Value or condition**
Phase radius	3 periods in each direction
Default trend enhancement	$(0.05, 0.03, 0.035)$
TimesFM trend enhancement	$(0.10, 0.05, 0.06)$
Trend coefficient	1.5; 2.0 when all enhancement tests hold
Seasonal coefficient	0 if $q \le 0.01$; otherwise $\mathrm{round}_2(0.4 + 0.4v)$
Residual coefficient	Equation 10; two-decimal rounding

STL settings. The numerical environment uses statsmodels 0.14.6. Decomposition specifies the period and robust fitting, with seasonal span 7, degree 1 for all three smoothers, and jump 1 for all three. The trend span is the smallest odd integer greater than $1.5p/(1 - 1.5/7)$, and the low-pass span is the smallest odd integer greater than p. Robust fitting uses two inner iterations and fifteen outer iterations.

The numerical interface requires $L \ge 4$, at least one original observation, and an integer period $2 \le p \le \lfloor L/2 \rfloor$. The period is specified from dataset frequency and configuration. Table 4 summarizes empirical hyperparameters, and Table 5 gives conventions for special inputs.

Table 5: Boundary cases under the defined input contract.

Case	Behavior
No missing entries	No entries are replaced.
All entries missing	Reject: at least one original observation is required.
Nonfinite input or invalid period	Reject the input.
Zero standard deviation	Use scale 1 in normalized statistics.
Prefix or suffix gap	Keep its trend reference unchanged.
No observed phase neighbors	Keep the corresponding component reference unchanged.
Residual-shrinkage action	Shrink residuals even if phase references are unavailable.
Multiple channels	The operator itself handles one channel at a time.

C Operator properties and interpretive examples

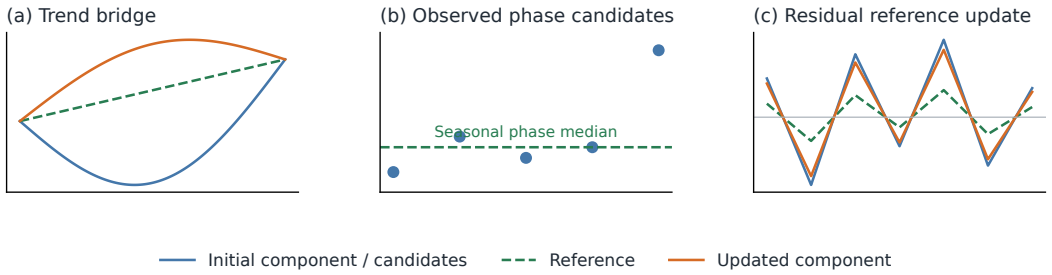


Figure 6: Constructed component corrections: trend follows a bridge, seasonality uses the median of observed same-phase candidates, and residuals move toward a phase reference or zero. These illustrations describe the operator and are not empirical observations.

C.1 Localization by masked writeback

Define $\mathcal{C}(z, M) = \{u : u_i = z_i \text{ for all } i \in O\}$. For any finite candidate v, masked writeback $P_M(v) = (1 - M) \odot z + M \odot v$ belongs to $\mathcal{C}(z, M)$ and is the Euclidean projection of v onto this affine set. Minimize $\sum_i (u_i - v_i)^2$ coordinatewise: observed coordinates are fixed at z_i, and free missing coordinates attain their minimum at $u_i = v_i$. This provides a geometric interpretation of observation preservation in the writeback step.

C.2 Reference bounds and zero-increment fallback

For nonempty N_i, the seasonal median lies between the candidate extrema, and normalized residual weights are positive and sum to one. Thus, for $C \in \{S, R\}$,

$$\min_{j \in N_i} C_j \le C_i^* \le \max_{j \in N_i} C_j. \tag{11}$$

For $0 \le \alpha_C \le 1$, the reference update $C_i^+ = (1 - \alpha_C)C_i + \alpha_C C_i^*$ lies in the convex hull of C_i and the candidate values. If $C_i^* = C_i$, its increment is exactly zero in the component-update formula. Residual shrinkage uses $R_i^+ = (1 - \alpha_R)R_i$ instead, so unavailable neighbors do not prevent that action. Finite-precision component reconstruction can introduce roundoff at missing entries even when reference increments vanish; observed values are still copied exactly.

C.3 Correction magnitude and trend overshoot

For the three-component action, the triangle inequality gives

$$|z_i^+ - z_i| \le \alpha_T |T_i^* - T_i| + \alpha_S |S_i^* - S_i| + \alpha_R |R_i^* - R_i| \quad (i \in G). \tag{12}$$

This bound relates correction magnitude to deviations from all three references. To isolate trend enhancement, set the seasonal increment to zero, $T_i = 10$, $T_i^* = 12$, $R_i = 1$, $R_i^* = 0.5$, and $\alpha_R = 0.2$. Trend coefficients of 1.5 and 2.0 yield total increments of 2.9 and 3.9, respectively. Forecast error also depends on the response of the target forecaster.

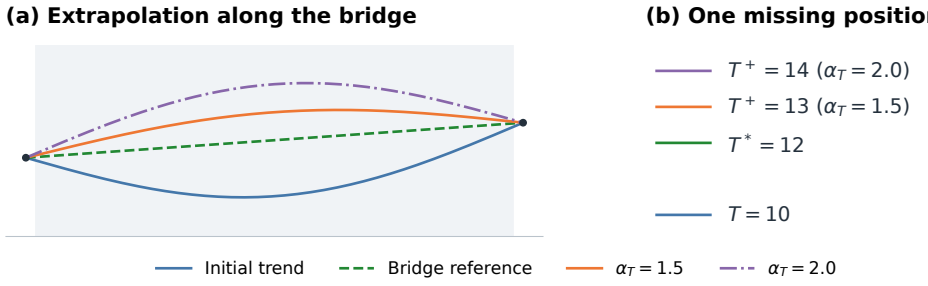


Figure 7: Constructed illustration of trend updates. The bridge provides the correction direction, coefficients 1.5 and 2.0 extend the update beyond the reference, and the identity action retains the initial value. Curves are constructed from the update equations.

C.4 An analytical example of reconstruction and forecast rankings

Consider a linear forecaster $F_W(z) = Wz$ with target $y = Wx$ and error $e = z - x$. Squared reconstruction error is $e^\mathsf{T} e$, while squared forecast error is $e^\mathsf{T} W^\mathsf{T} W e$. Set the true values on two error coordinates to zero, with a common future target zero for both forecasters below. Take $e_a = (1, 0)$ and $e_b = (0, 2)$. Their reconstruction losses are 1 and 4. Forecaster $W_1 = (1, 0)$ gives forecast losses 1 and 0, whereas $W_2 = (0, 1)$ gives 0 and 4 (Figure 8). Remaining historical coordinates can be observed and identical. Thus, even the preferred correction can change with the forecaster. This analytical construction illustrates how two forecasting maps respond differently to the same input errors.

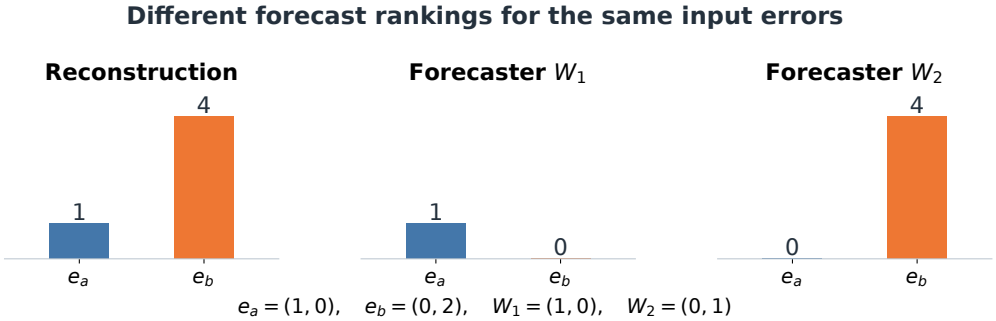


Figure 8: An analytical example of different input rankings. Reconstruction prefers error e_a; forecaster W_1 prefers e_b; forecaster W_2 prefers e_a. Bars show squared losses calculated from the stated vectors.

D Experimental details

D.1 Data scope

We use the 18 univariate datasets listed in Table 6.

For each dataset, we use the original windows ranked 1 through 20 by forecast origin. Table 6 lists target series, sampling frequency, context length, period, and forecast horizon. Each window has five missingness patterns at three ratios, giving 15 scenes.

Table 6: Dataset settings. Each target column defines one univariate series, with 20 windows and 300 missing scenes per dataset. T, H, D, and W denote minutes, hours, days, and weeks.

Dataset	**Target**	**Freq.**	L	p	H
Coastal_T_S_15T	PSAL	15T	2048	96	48
Coastal_T_S_20T	TEMP	20T	2048	72	48
Coastal_T_S_H	PRES_REL	H	2048	24	48
ETTh1	OT	H	2048	24	48
ETTh2	OT	H	2048	24	48
ETTm1	OT	15T	2048	96	48
ETTm2	OT	15T	2048	96	48
OpenElectricity_NEM_5T	demand__QLD1	5T	2048	288	48
Water_Quality_Darwin_15T	DOX2	15T	2048	96	48
current_velocity_15T	TEMP	15T	2048	96	48
current_velocity_20T	ABSIC1	20T	2048	72	48
current_velocity_5T	PRES_REL	5T	2048	288	48
current_velocity_H	ABSIC1	H	2048	24	48
electricity	OT	H	2048	24	48
exchange_rate	OT	D	2048	7	30
national_illness	OT	W	926/958	52	8
traffic	OT	H	2048	24	48
weather	OT	10T	2048	144	48

D.2 Missingness construction

Each missing block spans 50 time steps, with at least 50 observed steps between blocks and a 50-step guard at each end. The block count is obtained by rounding the nominal ratio times context length divided by 50, so realized ratios can differ slightly. Masks are nested across the three ratios within a window and pattern. Random blocks use stable random priorities; extreme blocks prioritize deviation from the window median; change blocks use the difference between medians of the first and second halves; volatility blocks use the median absolute deviation of within-block first differences; periodic blocks follow a regular grid. Structured mask construction uses complete history to locate target intervals, after which masked values are hidden from imputation and refinement. Future targets do not enter mask selection.

D.3 Base imputer implementations

The 16 base imputers comprise mean, forward fill, backward fill, linear interpolation, PCHIP, quadratic interpolation, and cubic spline; seasonal-residual and time-phase KNN; structural Kalman, Kalman–ARIMA, GP RBF, and MICE; plus registered overlapping-window implementations of MOMENT-base, SAITS-history, and NuwaTS-GPT2-large. Both sides of each pair use the same registered actual imputation array. The historical knn label equals mean numerically in all 5,400 scenes, so this panel retains mean only; fixed poly2 and SAITS-history implementations represent their corresponding historical variants. Label reduction does not depend on current forecast gains.

MOMENT, SAITS, and NuwaTS use their registered existing weights and historical overlapping-window imputers; no base imputer is retrained from refinement gains. Statistical and interpolation methods use their registered fixed implementations and parameters. Results apply to these 16 concrete implementations rather than to every variant sharing a method name.

D.4 FORECASTING PROTOCOL

All four forecasters use at most 2048 historical steps, batch size 32, and P50 point forecasts. Base and refined inputs from the same scene share horizon, random state, model snapshot, and complete inference contract; Kairos restores its initial routing state before each input. Chronos-2, TimesFM 2.5, and Kairos use adapter-returned P50 quantiles, while Sundial generates 100 sample paths and takes a per-step median. Table 7 identifies model snapshots and software versions.

Table 7: Forecaster snapshots; revisions show their first 12 characters.

Model	**Model identifier**	**Revision**
Chronos-2	amazon/chronos-2	29ec3766d36d
TimesFM 2.5	google/timesfm-2.5-200m-pytorch	1d952420fba8
Kairos-50M	mldi-lab/Kairos_50m	b788434dd95f
Sundial	thuml/sundial-base-128m	3212e4256449

The main environment uses NumPy 2.2.5, pandas 2.2.3, PyTorch 2.5.1 with CUDA 12.4, Transformers 4.56.0, and statsmodels 0.14.6; the Chronos package version is 2.2.2. Refinement itself makes no forecaster calls and does not use future error to update the current input.

D.5 METRICS AND AGGREGATION

For a target y and point forecast $\widehat{y}$ of length H, we use the metrics below. sMAPE requires an explicit convention at zero denominators (Hyndman & Koehler, 2006); we set these terms to zero:

$$\mathrm{sMAPE} = \frac{200}{H} \sum_{h=1}^{H} \frac{|y_h - \widehat{y}_h|}{|y_h| + |\widehat{y}_h|}. \tag{13}$$

$$\mathrm{MAE} = \frac{1}{H} \sum_{h=1}^{H} |y_h - \widehat{y}_h|, \qquad \mathrm{RMSE} = \sqrt{\frac{1}{H} \sum_{h=1}^{H} (y_h - \widehat{y}_h)^2}. \tag{14}$$

Let $A = \{i : 2 \le i \le L,\ M_{i-1} = M_i = 0\}$ and define historical scale $s = |A|^{-1} \sum_{i \in A} |x_i - x_{i-1}|$. Normalized MAE and RMSE are MAE/s and RMSE/s, respectively. The scale uses only adjacent, originally observed historical positions and is fixed across all imputations and refined outputs within a scenario. All 540 scenarios have positive scales. Hidden historical truth is not used, and raw errors in different units are not averaged together.

Each dataset–forecaster–imputer cell contains 300 scenes: 20 windows, five patterns, and three ratios. We average within each cell and then equally across the $18 \times 4 \times 16$ cells; balanced coverage makes this equal to the direct mean over 345,600 paired errors. Relative improvement is $100(E_0 - E_+)/E_0$. Improvement, deterioration, and equality use strict case-level sMAPE comparisons. Between-imputer standard deviation uses the population denominator of 16; range is the maximum minus minimum of the 16 method means. Both are computed within dataset–forecaster cells and then averaged over 72 cells.

All 5,400 scenes and 86,400 base imputations enter the comparison; each side contains 345,600 logical prediction records, with no missing predictions or technical failures. All outcomes are retained, including 155,424 deteriorations and 3,314 unchanged pairs. The completion audit passes structural coverage, pairing, and finite saved-metric checks; we additionally verify the core means and counts from the full paired CSV. It does not independently recompute all metrics from forecast arrays. Historical windows are dependent, so we report descriptive means and counts rather than treating 345,600 pairs as independent samples for significance testing.

E DETAILED EXPERIMENTAL RESULTS

This appendix reports descriptive results from the complete panel by dataset, forecaster, 16 base imputers, missingness condition, window rank, and source group. Base/refined entries use identical scenes. Among 345,600 pairs, sMAPE decreases in 186,862 cases, increases in 155,424, and is unchanged in 3,314 (54.07%, 44.97%, and 0.96%, respectively); all outcomes remain in the grouped tables. NMAE and NRMSE use the fixed historical scale defined in Appendix D; this panel has zero zero-scale scenes.

E.1 RESULTS BY DATASET

Table 8: sMAPE by dataset. Gain is relative percent; counts are better/worse/equal, with 19,200 pairs per row.

Dataset	**Base**	**Refined**	**Gain %**	**Better/Worse/Equal**
Coastal_T_S_15T	0.1520	0.1518	0.13	9,689/8,315/1,196
Coastal_T_S_20T	0.6829	0.6495	4.89	11,310/7,874/16
Coastal_T_S_H	14.2623	12.9714	9.05	10,865/8,335/0
ETTh1	13.0387	12.8590	1.38	10,349/8,851/0
ETTh2	8.3568	8.2786	0.94	10,146/9,054/0
ETTm1	10.2057	10.0438	1.59	10,606/8,554/40
ETTm2	6.5622	6.4264	2.07	10,903/8,265/32
OpenElectricity_NEM_5T	3.7529	3.6509	2.72	10,224/8,940/36
Water_Quality_Darwin_15T	0.5875	0.5693	3.10	10,573/7,919/708
current_velocity_15T	0.6677	0.6609	1.02	10,358/8,842/0
current_velocity_20T	2.1771	2.1381	1.79	10,760/7,308/1,132
current_velocity_5T	0.0541	0.0632	-17.01	7,789/11,411/0
current_velocity_H	2.5082	2.4590	1.96	10,428/8,772/0
electricity	3.7657	3.5541	5.62	10,990/8,210/0
exchange_rate	1.8878	1.7761	5.91	9,877/9,323/0
national_illness	8.8384	8.7307	1.22	10,104/9,022/74
traffic	16.5170	15.5358	5.94	11,567/7,633/0
weather	0.6511	0.6432	1.21	10,324/8,796/80

Table 9: Fixed-history-scale-normalized errors by dataset.

Dataset	NMAE base	NMAE refined	NRMSE base	NRMSE refined
Coastal_T_S_15T	1.4309	1.4290	1.7448	1.7414
Coastal_T_S_20T	11.1458	10.6190	13.4163	12.8490
Coastal_T_S_H	12.0794	11.1895	15.8021	14.7951
ETTh1	2.6307	2.5927	3.3532	3.3192
ETTh2	3.0222	2.9959	3.8202	3.7923
ETTm1	4.8157	4.7396	5.8311	5.7528
ETTm2	10.4452	10.2311	11.9879	11.7266
OpenElectricity_NEM_5T	4.6491	4.5169	5.5372	5.3822
Water_Quality_Darwin_15T	3.8415	3.7228	4.4719	4.3397
current_velocity_15T	13.4955	13.3536	16.3387	16.1763
current_velocity_20T	1.7663	1.7344	2.1219	2.0865
current_velocity_5T	3.1245	3.6664	3.6939	4.3429
current_velocity_H	1.5077	1.4786	1.8588	1.8243
electricity	1.0574	1.0033	1.3564	1.2879
exchange_rate	3.8013	3.5702	4.3859	4.1293
national_illness	3.0975	3.0633	3.5915	3.5468
traffic	0.7797	0.7255	1.0786	1.0050
weather	3.9554	3.9101	4.6783	4.6264

E.2 Results by forecaster and imputer

Table 10: Mean sMAPE and paired counts by forecaster; 86,400 pairs per row.

Forecaster	Base	Refined	Gain %	Better/Worse/Equal
Chronos-2	5.1801	4.9468	4.50	45,100/40,473/827
TimesFM 2.5	5.1777	4.9948	3.53	47,056/38,511/833
Kairos-50M	5.2243	5.0324	3.67	47,166/38,407/827
Sundial	5.4552	5.2842	3.14	47,540/38,033/827

Table 11: Normalized errors by forecaster.

Forecaster	NMAE base	NMAE refined	NRMSE base	NRMSE refined
Chronos-2	4.8202	4.6557	5.8275	5.6461
TimesFM 2.5	4.5834	4.4695	5.5744	5.4482
Kairos-50M	4.8287	4.7309	5.8943	5.7790
Sundial	5.0223	4.9310	6.0524	5.9543

Table 12: Mean sMAPE and paired counts by base imputer; 21,600 pairs per row.

Base imputer	Base	Refined	Gain %	Better/Worse/Equal
Mean	5.7258	5.2580	8.17	11,619/9,917/64
Forward fill	5.1163	5.0584	1.13	11,272/10,328/0
Backward fill	5.1932	5.0518	2.72	11,506/10,086/8
Linear	5.1043	4.9984	2.07	11,756/9,844/0
PCHIP	5.1095	4.9958	2.23	11,771/9,829/0
Quadratic	5.6006	5.2256	6.70	11,415/8,309/1,876
Cubic spline	5.2619	5.0419	4.18	11,818/9,358/424
Seasonal residual	4.8483	4.7728	1.56	11,766/9,834/0
Time-phase KNN	4.8068	4.7836	0.48	11,092/10,508/0
Structural Kalman	5.5085	5.2988	3.81	12,025/9,571/4
Kalman–ARIMA	5.4669	5.1504	5.79	11,810/9,766/24
GP RBF	5.1286	5.0263	2.00	11,741/9,785/74
MICE	5.4893	5.1829	5.58	11,523/9,969/108
MOMENT-base	5.2767	5.0587	4.13	12,095/9,497/8
SAITS-history	5.1898	5.0450	2.79	11,668/9,208/724
NuwaTS-GPT2-large	5.3227	5.0843	4.48	11,985/9,615/0

Table 13: Fixed-history-scale-normalized errors by base imputer.

Base imputer	NMAE base	NMAE refined	NRMSE base	NRMSE refined
Mean	5.7472	5.2074	6.8733	6.2630
Forward fill	4.5896	4.5717	5.5908	5.5659
Backward fill	4.6456	4.6139	5.6743	5.6329
Linear	4.5470	4.5350	5.5319	5.5195
PCHIP	4.5375	4.5260	5.5201	5.5097
Quadratic	4.7707	4.6519	5.7896	5.6575
Cubic spline	4.6888	4.6091	5.7016	5.6105
Seasonal residual	4.4808	4.4986	5.4636	5.4893
Time-phase KNN	4.5027	4.5128	5.4899	5.5043
Structural Kalman	4.5935	4.5536	5.5852	5.5440
Kalman–ARIMA	5.5137	5.0737	6.6049	6.1159
GP RBF	4.5819	4.5759	5.5670	5.5649
MICE	5.5591	5.2367	6.6505	6.2944
MOMENT-base	4.7846	4.6687	5.8248	5.6942
SAITS-history	4.8677	4.7663	5.9212	5.8075
NuwaTS-GPT2-large	4.6081	4.5471	5.6056	5.5365

Table 14: Mean sMAPE for 16 base imputers and four forecasters (base/refined); 5,400 pairs per cell.

Base imputer	Chronos-2	TimesFM 2.5	Kairos-50M	Sundial
Mean	5.726 / 5.151	5.625 / 5.241	5.769 / 5.202	5.783 / 5.438
Forward fill	4.923 / 4.870	5.102 / 5.056	5.094 / 5.012	5.346 / 5.296
Backward fill	5.035 / 4.872	5.169 / 5.010	5.110 / 5.022	5.458 / 5.303
Linear	4.971 / 4.844	5.050 / 4.943	4.990 / 4.913	5.406 / 5.293
PCHIP	4.963 / 4.833	5.061 / 4.950	4.993 / 4.912	5.420 / 5.289
Quadratic	5.298 / 4.981	5.552 / 5.190	5.804 / 5.262	5.749 / 5.470
Cubic spline	5.199 / 4.943	5.230 / 4.996	5.208 / 5.023	5.410 / 5.206
Seasonal residual	4.819 / 4.774	4.621 / 4.517	4.880 / 4.817	5.074 / 4.984
Time-phase KNN	4.841 / 4.797	4.555 / 4.549	4.867 / 4.849	4.964 / 4.939
Structural Kalman	5.275 / 5.083	5.490 / 5.273	5.475 / 5.236	5.795 / 5.603
Kalman–ARIMA	5.303 / 5.060	5.407 / 5.108	5.448 / 5.053	5.709 / 5.380
GP RBF	5.007 / 4.873	5.134 / 5.027	5.018 / 4.945	5.356 / 5.260
MICE	5.761 / 5.164	5.341 / 5.125	5.273 / 5.094	5.583 / 5.349
MOMENT-base	5.520 / 5.060	5.080 / 4.942	5.187 / 5.066	5.320 / 5.167
SAITS-history	5.045 / 4.929	5.183 / 4.985	5.204 / 5.074	5.327 / 5.192
NuwaTS-GPT2-large	5.193 / 4.915	5.244 / 5.005	5.270 / 5.037	5.584 / 5.380

E.3 Results by missingness pattern and ratio

Table 15: Base/refined errors by missingness pattern; 69,120 pairs per row.

Pattern	sMAPE	NMAE	NRMSE
Random	5.1715 / 5.0150	4.4762 / 4.4056	5.4610 / 5.3772
Extreme	5.4166 / 5.1593	5.1511 / 4.8963	6.1701 / 5.8980
Change	5.4658 / 5.1938	4.9531 / 4.8160	6.0393 / 5.8723
Volatility	5.2342 / 5.0406	5.0858 / 4.9663	6.1619 / 6.0299
Periodic	5.0086 / 4.9140	4.4021 / 4.3996	5.3534 / 5.3570

Table 16: Base/refined errors by nominal missingness ratio; 115,200 pairs per row.

Nominal ratio	sMAPE	NMAE	NRMSE
10%	4.8946 / 4.8134	4.5167 / 4.4698	5.4905 / 5.4428
20%	5.2112 / 5.0308	4.7732 / 4.6623	5.7887 / 5.6637
30%	5.6722 / 5.3494	5.1511 / 4.9582	6.2323 / 6.0141

E.4 Between-imputer dispersion

Table 17: Between-imputer standard deviation and range by forecaster. Each value is computed across 16 imputers within a dataset–forecaster cell, then averaged equally over 18 datasets.

Forecaster	Std. base	Std. refined	Range base	Range refined
Chronos-2	0.4527	0.2169	1.6310	0.8092
TimesFM 2.5	0.4553	0.3224	1.7916	1.2695
Kairos-50M	0.4837	0.2358	1.8426	0.9544
Sundial	0.4366	0.3127	1.6819	1.2074

Table 18: Between-imputer standard deviation and range by dataset, computed across 16 methods within each forecaster and then averaged across four models.

Dataset	Std. base	Std. refined	Range base	Range refined
Coastal_T_S_15T	0.0064	0.0065	0.0270	0.0268
Coastal_T_S_20T	0.2446	0.1980	0.7357	0.6220
Coastal_T_S_H	2.5048	0.7516	9.1185	2.8360
ETTh1	0.3375	0.1552	1.4575	0.6634
ETTh2	0.1841	0.1129	0.6228	0.3901
ETTm1	0.3019	0.2569	1.0848	0.8589
ETTm2	0.2988	0.2716	0.9117	0.7857
OpenElectricity_NEM_5T	0.3337	0.2349	1.1825	0.7787
Water_Quality_Darwin_15T	0.1100	0.0906	0.4898	0.4033
current_velocity_15T	0.0162	0.0098	0.0658	0.0385
current_velocity_20T	0.2131	0.2075	0.8451	0.8337
current_velocity_5T	0.0081	0.0052	0.0261	0.0183
current_velocity_H	0.1051	0.0441	0.3938	0.1810
electricity	0.3886	0.0831	1.2900	0.2969
exchange_rate	0.2385	0.0264	0.7689	0.1011
national_illness	0.4333	0.3946	1.3797	1.3351
traffic	2.4878	2.0369	10.8137	8.8785
weather	0.0149	0.0092	0.0487	0.0345

E.5 Window ranks and source groups

Window ranks and source groups show coverage and heterogeneity in the complete results; they are not separately sampled test sets. Source groups combine datasets sharing a source.

Table 19: Base/refined errors by window rank 1–20; 17,280 pairs per row.

Window	sMAPE	NMAE	NRMSE
1	4.7649 / 4.6543	4.4884 / 4.3080	5.4046 / 5.1912
2	7.8018 / 7.6275	6.3759 / 6.1726	7.4638 / 7.2293
3	4.7608 / 4.5747	3.9416 / 3.8171	4.6574 / 4.5112
4	4.7599 / 4.5032	4.5350 / 4.2484	5.3774 / 5.0444
5	3.4865 / 3.3583	2.9762 / 2.8856	3.5789 / 3.4741
6	5.2823 / 5.1718	5.4363 / 5.3516	6.7077 / 6.6237
7	3.8418 / 3.6450	2.8663 / 2.7608	3.4489 / 3.3234
8	3.8428 / 3.6609	3.9846 / 3.7990	4.7273 / 4.5152
9	3.4243 / 3.3038	3.9092 / 3.8452	4.6783 / 4.6101
10	3.4130 / 3.2046	3.3800 / 3.3492	4.1901 / 4.1403
11	3.7100 / 3.5132	3.5280 / 3.4460	4.2874 / 4.2015
12	7.0125 / 7.0699	4.9846 / 5.0252	5.8495 / 5.9013
13	6.9904 / 6.8644	9.3879 / 9.3521	12.7712 / 12.7439
14	9.2081 / 8.7554	8.5308 / 8.3523	10.0824 / 9.9044
15	5.0373 / 4.8996	4.2902 / 4.1785	5.1268 / 5.0018
16	4.7167 / 4.5374	4.5218 / 4.4408	5.3797 / 5.2899
17	7.0371 / 6.8510	4.1345 / 4.0611	4.9918 / 4.9191
18	6.6194 / 6.3180	5.9530 / 5.7152	7.1588 / 6.8969
19	3.9335 / 3.7163	3.7798 / 3.7198	4.4544 / 4.3819
20	5.5434 / 5.0617	5.2688 / 5.1070	6.4067 / 6.2339

Table 20: Mean sMAPE by data-source group; groups contain different numbers of datasets and therefore different pair counts.

Source group	**Pairs**	**Base/refined sMAPE**	**Gain %**
Coastal_T_S	57600	5.0324 / 4.5909	8.77
ETT	76800	9.5409 / 9.4020	1.46
OpenElectricity_NEM_5T	19200	3.7529 / 3.6509	2.72
Water_Quality_Darwin_15T	19200	0.5875 / 0.5693	3.10
current_velocity	76800	1.3518 / 1.3303	1.59
electricity	19200	3.7657 / 3.5541	5.62
exchange_rate	19200	1.8878 / 1.7761	5.91
national_illness	19200	8.8384 / 8.7307	1.22
traffic	19200	16.5170 / 15.5358	5.94
weather	19200	0.6511 / 0.6432	1.21