ARCI-T: Anchor-Residual Controlled Imputation for Missing Traffic Sensor Data

Anonymous authors
Paper under double-blind review

Abstract

Missing values in traffic sensor records may manifest as isolated points, consecutive blocks, or sensor-level outages, yielding heterogeneous temporal and spatial evidence across different missing positions. Deterministic recovery provides a stable reference, while learned models can capture richer spatiotemporal dynamics. Rather than treating these two solutions as competing alternatives, we formulate traffic imputation as controlled correction around a retained deterministic reference. We formalize **A**nchor-**R**esidual **C**ontrolled **I**mputation for Missing **T**raffic Sensor Data (ARCI-T), which employs the deterministic recovery as an explicit Anchor, learns an anchor-relative correction, and determines the contribution magnitude of this correction at each missing position based on available observational evidence. This design preserves the deterministic reference as a fallback while enabling learned corrections to adapt under heterogeneous missingness conditions. Across four traffic benchmarks under random and structured missingness protocols, ARCI-T achieves state-of-the-art or highly competitive imputation accuracy, demonstrating that controlled correction is an effective approach for traffic sensor imputation.

1 Introduction

Traffic data collected from roadside sensors play a vital role in traffic monitoring, forecasting, and control. Nevertheless, communication breakdowns and sensor malfunctions frequently lead to partial data loss in real-world scenarios. Missing values in time-series sensor data may appear as isolated points, consecutive gaps, or prolonged outages, each scenario offering varying degrees of useful information near the missing location. While an isolated missing value is commonly surrounded by adjacent observations, long gaps or sensor outages suffer from scarce local temporal cues and thus require stronger utilization of inter-sensor correlations. Therefore, different missing positions may require different degrees of reliance on temporal and spatial evidence and should not always be treated uniformly.

Existing traffic imputation methods can be broadly categorized from two perspectives. Deterministic recovery methods infer missing values from available observations through predefined procedures. Representative approaches include interpolation, matrix completion (Candès & Recht, 2009), tensor decomposition (Kolda & Bader, 2009), and multi-view estimation (Yi et al., 2016). These methods can provide stable estimates directly from observed data, although the patterns they can represent are limited by the chosen procedure. In contrast, learned approaches capture richer temporal and spatial relationships from data, including recurrent models (Cao et al., 2018), attention-based methods (Du et al., 2023), graph-based methods (Cini et al., 2022), and generative models (Tashiro et al., 2021; Lopez Alcaraz & Strodthoff, 2023; Liu et al., 2023). These approaches can capture complex data patterns and typically produce missing-value estimates directly from the learned model.

Recent works have developed two closely related directions. First, emerging methods focus on correcting estimated values. DCCN-SPF enhances linear interpolation by incorporating learned deviation terms (Yuan et al., 2024), while RDPI further optimizes imputation results via a diffusion-based refinement approach (Liu et al., 2025). Second, other methods explore how the learned information is used and controlled locally. SAITS combines learned estimates with missing-aware attention weights (Du et al., 2023); OPCR adjusts information propagation according to pointwise confidence (Deng et al., 2024); Meta Guidance and FENCE adapt guidance strength using input-

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.

dependent evidence (You et al., 2025b; Mao et al., 2026). Similar adaptive control strategies are also adopted in input-dependent weighting and structured expert routing mechanisms (You et al., 2025a; Qin et al., 2026).

Overall, these advances demonstrate that imputation performance can be improved by correcting preliminary deterministic estimates and adaptively modulating the utilization of learned information for different missing positions. Nevertheless, correction proposals and their effective magnitude are generally not explicitly decoupled. Furthermore, adaptive control does not necessarily modulate corrections with respect to a retained deterministic reference. When these two components are jointly determined, the final estimate blends the correction content and correction strength together, which makes it difficult to disentangle their respective responsibilities and thus hinders direct control over correction contributions governed by position-dependent evidence. Therefore, we treat deterministic recovery and learned correction as two parts of a controlled correction process, which leads to two design questions: ***What correction proposal can be learned once a deterministic recovery is available?*** and ***What degree of the learned corrections will be accepted at each missing position when the available evidence differs across positions?***

To this end, we propose Anchor-Residual Controlled Imputation for Missing Traffic Sensor Data (ARCI-T), which retains the deterministic recovery in the final output as an explicit Anchor. Specifically, ARP (Anchor-Relative Residual Parameterization) learns an anchor-relative correction proposal, while PBRA (Position-Conditioned Bounded Residual Acceptance) applies evidence about temporal context, spatial support, and the local missingness structure to determine how much of that proposal is accepted at each position. Different from FENCE, which controls correction strength by adjusting diffusion guidance, ARCI-T controls how much an anchor-relative residual contributes to an explicitly retained deterministic output.

Our contributions are summarized as follows:

- **Controlled correction around a deterministic reference.** We formulate traffic imputation as a bounded controlled-correction framework around an explicitly defined deterministic recovery reference. The learned component predicts an anchor-relative residual proposal and uses position-conditioned bounded acceptance to determine how much of that proposal is realized, while retaining the deterministic reference in the output.
- **Separating correction proposal from acceptance.** We design ARP and PBRA to separate what correction is proposed from how much of it is accepted. ARP learns an anchor-relative correction, while PBRA controls its position-specific acceptance using temporal, spatial, and missingness-geometry evidence.
- **Strong performance across missingness conditions.** We achieve state-of-the-art or highly competitive imputation accuracy across random and structured missing scenarios, outperforming strong baselines in most scenarios. Stress tests further support the robustness of controlled correction under heavier missingness and temporal-distance evaluations.

2 Problem Definition

2.1 Traffic Sensor Data Imputation

Let $X \in \mathbb{R}^{B\times T\times N}$ be a batch of traffic sequences, where B, T, and N denote the number of samples, the temporal length, and the number of sensors, respectively. Let $V \in \{0,1\}^{B\times T\times N}$ indicate the validity of the original dataset values before artificial masking, and $M \in \{0,1\}^{B\times T\times N}$ the observations exposed to the model. The model input is

$$X_{\text{obs}} = M \odot X. \tag{1}$$

The goal is to recover the missing entries corresponding to the mask condition $M = 0$ and flag $V = 1$, while strictly forbidding any modification to the observed visible values when $M = 1$. All inherently invalid positions are fully excluded from both the loss supervision and the final evaluation.

2.2 Position-specific support under heterogeneous missingness

Missing positions with the same observation status can nevertheless differ substantially in the evidence available for imputation. Position near the boundary of a short gap may retain nearby obser-

vations on one or both temporal sides, whereas positions deeper inside a long contiguous gap can be farther from visible support; sensor-level outages may remove such local temporal evidence over an extended interval. Hence, imputation difficulty depends not only on whether a value is missing, but also on where the position lies relative to the surrounding observed values.

These position-specific support conditions also determine what observations are available to a deterministic recovery. Thus, even when the same deterministic recovery strategy is used throughout a sequence, its recovery reference is formed under heterogeneous observation support across missing positions. We denote a deterministic description of this support condition derived from the observation mask M by position descriptor. Its exact components and boundary conventions are provided in Appendix Table 9.

3 Controlled Correction for Traffic Imputation

3.1 Motivation of Controlled Correction

Under heterogeneous support conditions, deterministic recovery provides a valuable starting point for controlled correction. It yields a computable recovery reference that is stable, consistent with the observed values, and usable as a fallback in the absence of learned corrections. Nevertheless, its fixed rules restrict the spatiotemporal dynamics that can be captured within a missing segment.

A natural idea is to employ a learned model to supply such dynamics, for instance by constructing richer trend and fluctuation candidates around the recovery reference. However, boosting the representation capacity of the learned candidate does not necessarily reduce the final imputation error. As illustrated in Figure 1, enabling the evaluated trend/fluctuation enhancement under ER60 missing protocol yields a slight MAE increase on both PeMS-BAY and PEMS04. The issue is therefore not only how rich a correction the learned model can propose, but also how the proposed correction should be incorporated into the existing deterministic recovery.

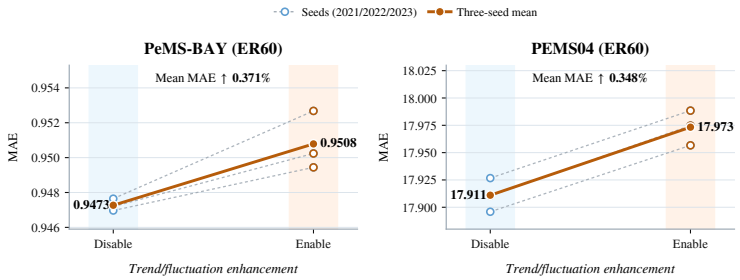


Figure 1: Motivating observation for controlled correction. Three-seed mean MAE under ER60, with the tested trend/fluctuation enhancement disabled and enabled. Enabling this enhancement increases MAE by 0.371% on PeMS-BAY and 0.348% on PEMS04.

ARCI-T addresses these two aspects within the imputation process. It starts from a deterministic recovery reference, learns a residual proposal that describes how the reference should change, and controls the realized contribution of that proposal at each missing position. The proposal specifies the content of the correction, while its realized contribution is determined from the evidence available at the corresponding position.

Figure 2 visualizes the framework of the proposed ARCI-T. Starting from the observed series, it first forms a deterministic recovery reference A. ARP then predicts an anchor-relative residual proposal R, while PBRA uses position-specific evidence to determine the degree to which the proposal is accepted. The resulting controlled estimate is the raw output before the observed-value clamp.

3.2 Anchor-relative Residual Parameterization (ARP)

Direct absolute prediction requires the learned branch to determine the entire missing value. ARP instead specifies the learning task as predicting the correction that should be applied to the determin-

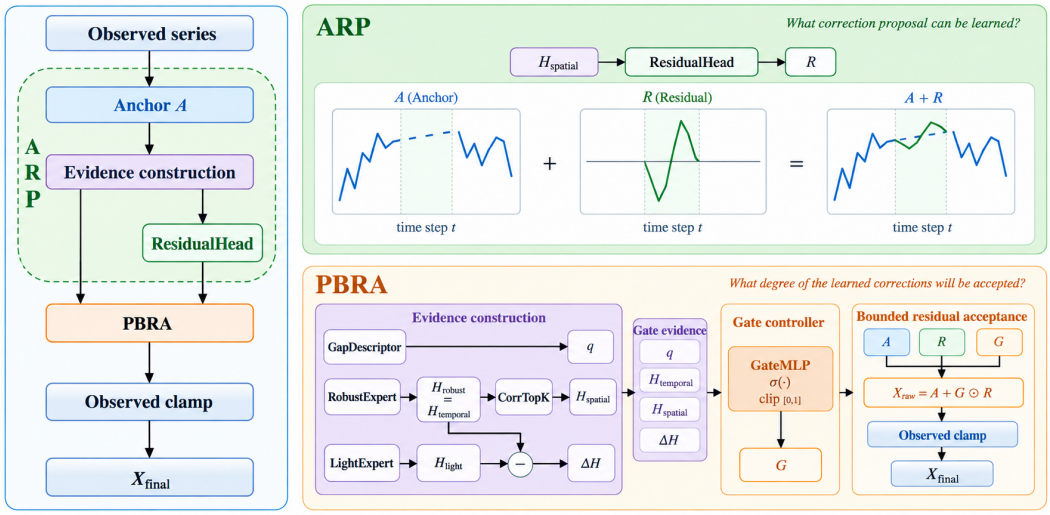


Figure 2: The controlled-correction framework of ARCI-T. The left part shows the end-to-end imputation flow organized as recovery reference $A \rightarrow$ residual proposal ARP $\rightarrow$ controlled acceptance PBRA; the right part details the anchor-relative residual proposal of ARP and the position-conditioned bounded residual acceptance of PBRA. GapDescriptor extracts mask-derived position-specific support descriptor q, RobustExpert forms the main multi-scale temporal state H_{robust}, LightExpert provides a lightweight contrast state H_{light}, and CorrTopK injects frozen sensor relations derived from the training set. Detailed definitions are provided in Appendix B.

istic recovery. The deterministic recovery reference is

$$A = \Phi_{\text{anchor}}(X_{\text{obs}}, M). \tag{2}$$

Here, A denotes the Anchor, *i.e.*, the deterministic recovery reference. Unless otherwise stated, Φ_{anchor} is instantiated with LinearFill, which preserves observed positions and fills missing positions independently along each sample–sensor sequence.

Given the Anchor A and the position-specific support descriptor q, ARCI-T builds temporal representations and integrates cross-sensor information through sparse sensor relations determined from the training data, producing a spatial state H_{spatial} with hidden feature dimension d_h:

$$H_{\text{spatial}} = f_{\text{spatial}}(f_{\text{temporal}}(A, M, q, C); \mathcal{G}_{\text{corr}}) \in \mathbb{R}^{B\times T\times N\times d_h}, \tag{3}$$

where C denotes the four-dimensional timestamp-derived calendar features defined in Appendix B.10, q is the mask-derived position-specific descriptor that summarizes position-specific missingness geometry and observation-support conditions, and $\mathcal{G}_{\text{corr}}$ is a sparse sensor relation graph determined only from training data. f_{temporal}, and f_{spatial} are the temporal encoder and spatial encoder, respectively. Then the anchor-relative correction proposal can be obtained

$$R = f_R(H_{\text{spatial}}) \in \mathbb{R}^{B\times T\times N}. \tag{4}$$

where R is the learned correction proposal, f_R denotes the anchor-relative residual proposal function. Given the recovery reference A and the proposal R, fully accepting the correction would yield $A+R$. In ARCI-T, $A + R$ is therefore used only as the conceptual full-acceptance endpoint, rather than as an intermediate representation consumed by the subsequent controller. When ARP determines the content of the correction, we consider the second question: *how much of the proposal should be accepted at each position?*

3.3 Position-conditioned Bounded Residual Acceptance (PBRA)

Fully accepting the proposal as $A + R$ applies the entire learned correction at every position where a proposal is produced, which can be undesirable under heterogeneous support conditions. PBRA leaves the content of R unchanged and controls its position-specific acceptance through a bounded

coefficient. In addition to the robust temporal state, ARCI-T computes a lightweight temporal state as a contrast:

$$\begin{aligned} H_{\text{light}} &= f_{\text{light}}(A, M, C), \\ \Delta H &= H_{\text{robust}} - H_{\text{light}}. \end{aligned} \tag{5}$$

where f_{light} denotes the lightweight temporal encoder. The acceptance controller reads position-specific acceptance evidence

$$E = [H_{\text{temporal}} \| H_{\text{spatial}} \| q \| \Delta H], \tag{6}$$

The evidence E aggregates the robust temporal state, the spatial refinement state, the position-specific support descriptor q, and the robust–light difference. The controller then produces the position-conditioned acceptance coefficient

$$G = \text{clip}_{[0,1]}(\sigma(f_G(E))) \in [0,1]^{B \times T \times N}, \tag{7}$$

Here, $\text{clip}_{[0,1]}(\cdot)$ denotes elementwise clipping to $[0,1]$, f_G denotes the position-conditioned acceptance scoring function. Since the sigmoid σ already maps to $(0,1)$, the explicit clip serves only as a numerical safeguard. $G_{b,t,n}$ is a position-conditioned acceptance coefficient that controls the actual contribution of the proposed correction to the final output, where b, t, and n denote the sample, time-step, and sensor indices, respectively. It is a bounded control variable, not a calibrated probability or confidence. The proposal and the acceptance are produced from different direct information paths: the acceptance controller does not read R or $A + R$.

Combining the acceptance coefficient with the correction proposal gives the final controlled correction

$$X_{\text{raw}} = A + G \odot R. \tag{8}$$

When $G_{b,t,n} = 0$, the output returns exactly to the deterministic recovery reference $A_{b,t,n}$; when $G_{b,t,n} = 1$, the correction proposals are fully accepted, yielding $A_{b,t,n} + R_{b,t,n}$; intermediate values accept a corresponding fraction of the proposal. The learned component is therefore a controlled correction built upon the deterministic recovery reference, rather than an independent estimate detached from it. The final output preserves all visible observations:

$$X_{\text{final}} = M \odot X_{\text{obs}} + (1 - M) \odot X_{\text{raw}}. \tag{9}$$

3.4 Implementation and Training

ARCI-T builds the anchor-relative correction proposal from a robust temporal representation and a cross-sensor spatial state. The acceptance evidence combines temporal, spatial, missingness-geometry, and differences between the lightweight and robust temporal states. These two paths correspond to correction proposal and correction acceptance, respectively. Boundary handling of the Anchor, the GapDescriptor, the construction of the sparse correlation graph, network details, and the complete training setup are given in Appendix B.

The training objective combines an imputation loss and an adjacent temporal-difference loss on artificially hidden positions with valid labels:

$$\mathcal{L} = \mathcal{L}_{\text{imp}} + \lambda_{\Delta} \mathcal{L}_{\Delta}. \tag{10}$$

Here, λ_{Δ} weights the adjacent temporal-difference loss; the loss definitions and its value are given in Appendix B.7.

4 Empirical Study

4.1 Experimental Setup

The evaluation covers four traffic datasets—METR-LA, PeMS-BAY, PEMS04, and PEMS08—all sampled at five-minute intervals. Missingness protocols include random entry missingness (ER25/ER60), mixture of random and contiguous block missingness (Mixed25), and sensor-level contiguous missingness (Sensor25, where 25 denotes a 25% missing rate). A visual illustration of these protocols is provided in Appendix D (Figure 8).

Comparison methods span traditional interpolation and representative learned baselines, including BRITS (Cao et al., 2018), SAITS (Du et al., 2023), CSDI (Tashiro et al., 2021), OPCR (Deng et al.,

2024), GiFlow (Zhang et al., 2026), Transformer + MG (You et al., 2025b), ATTST (Li et al., 2026), and others; the complete compared methods list and full numerical results are provided in Appendix D.

4.2 Main Results

Tables 1–4 report MAE under the four missingness protocols, where bold indicates the best result, underline denotes the second-best performance, and complete benchmark results are provided in Appendix D. Tables 1–4 show that ARCI-T maintains strong overall performance under both random and structured missingness and attains the lowest MAE across almost all evaluated dataset–protocol settings. The sole slight inversion occurs on PEMS04 Sensor25, where BRITS (Cao et al., 2018) is marginally better (Tables 1–4). These gains largely persist under contiguous-block and sensor-level missingness.

Table 1: MAE on METR-LA under the four missingness protocols.

Method	ER25	ER60	Mixed25	Sensor25
CSDI	3.4592±0.2497	3.7247±0.3654	4.9260±0.2862	5.1474±0.2272
BRITS	4.2009±0.0161	4.6183±0.0062	4.3530±0.0351	<u>4.4520</u>±0.0112
SAITS	4.4123±0.0652	4.5716±0.0186	4.5260±0.0668	4.4886±0.0140
GiFlow	3.6209±0.1168	4.1328±0.1328	6.5301±0.1808	7.2168±0.1929
OPCR	<u>2.3287</u>±0.0091	<u>2.7142</u>±0.0546	<u>4.2784</u>±0.1214	4.6025±0.0726
Transformer + MG	4.2944±0.0505	4.7638±0.0440	6.3536±0.0516	6.8506±0.0726
ATTST	3.5856±0.0972	4.4109±0.1554	10.6181±0.3847	12.0431±0.4613
ARCI-T	**2.1181**±0.2778	**2.4351**±0.3296	**3.2387**±0.0145	**3.4722**±0.0652

Table 2: MAE on PeMS-BAY under the four missingness protocols.

Method	ER25	ER60	Mixed25	Sensor25
CSDI	1.0453±0.0299	1.1468±0.0243	2.2649±0.0699	2.5579±0.0638
BRITS	1.9574±0.0022	2.0993±0.0072	2.0498±0.0091	2.0791±0.0135
SAITS	2.1544±0.0035	2.2152±0.0033	2.1910±0.0214	2.2126±0.0096
GiFlow	1.1481±0.0378	1.4861±0.0923	2.7913±0.0688	3.1286±0.0721
OPCR	<u>0.7781</u>±0.0094	<u>0.9782</u>±0.0047	<u>1.8081</u>±0.0133	<u>1.9898</u>±0.0299
Transformer + MG	1.8386±0.0107	1.9922±0.0076	2.7720±0.0124	2.9048±0.0127
ATTST	0.9964±0.0291	1.2272±0.0456	3.9115±0.3750	4.4718±0.4480
ARCI-T	**0.6867**±0.0011	**0.8648**±0.0020	**1.4817**±0.0148	**1.6458**±0.0185

Table 3: MAE on PEMS04 under the four missingness protocols.

Method	ER25	ER60	Mixed25	Sensor25
CSDI	18.4684±0.4699	19.7618±0.9237	26.0755±1.0736	27.8359±1.0915
BRITS	21.5783±0.0774	22.6212±0.0711	<u>21.8489</u>±0.3313	**21.8862**±0.0877
SAITS	22.4598±0.1624	23.1621±0.0189	22.9538±0.3743	23.0592±0.1664
GiFlow	23.5271±2.0956	25.4442±1.2988	33.3930±1.2381	35.5746±1.2919
OPCR	17.7324±0.1835	20.0364±0.1167	24.7138±0.1360	25.6527±0.2561
Transformer + MG	23.1509±0.0648	25.1713±0.1195	36.9225±0.3245	39.0646±0.4944
ATTST	<u>16.6890</u>±0.0968	<u>18.9221</u>±0.3474	72.0310±2.8391	83.0503±3.4107
ARCI-T	**15.8302**±0.0258	**16.7568**±0.0402	**20.6271**±0.1860	<u>21.9462</u>±0.1316

Table 4: MAE on PEMS08 under the four missingness protocols.

Method	ER25	ER60	Mixed25	Sensor25
CSDI	13.6322±0.2192	14.7765±0.7140	18.6175±0.5084	20.1170±0.4728
BRITS	17.8121±0.0365	19.7596±0.0445	18.5539±0.4163	18.8274±0.0999
SAITS	19.6267±0.1358	20.7299±0.0728	20.1992±0.6677	20.2777±0.2207
GiFlow	20.9106±1.2183	22.7674±1.8129	35.0715±2.9656	38.3634±3.6857
OPCR	13.8884±0.1652	15.3749±0.3076	20.2636±0.2873	21.1647±0.5851
Transformer + MG	18.6460±0.1815	20.5992±0.1146	31.7632±0.6950	33.3316±0.3092
ATTST	13.0032±0.0789	14.8604±0.2624	61.5602±2.3360	70.8680±2.9895
ARCI-T	**11.4385**±0.0339	**12.4842**±0.0523	**16.2881**±0.1773	**17.2828**±0.0660

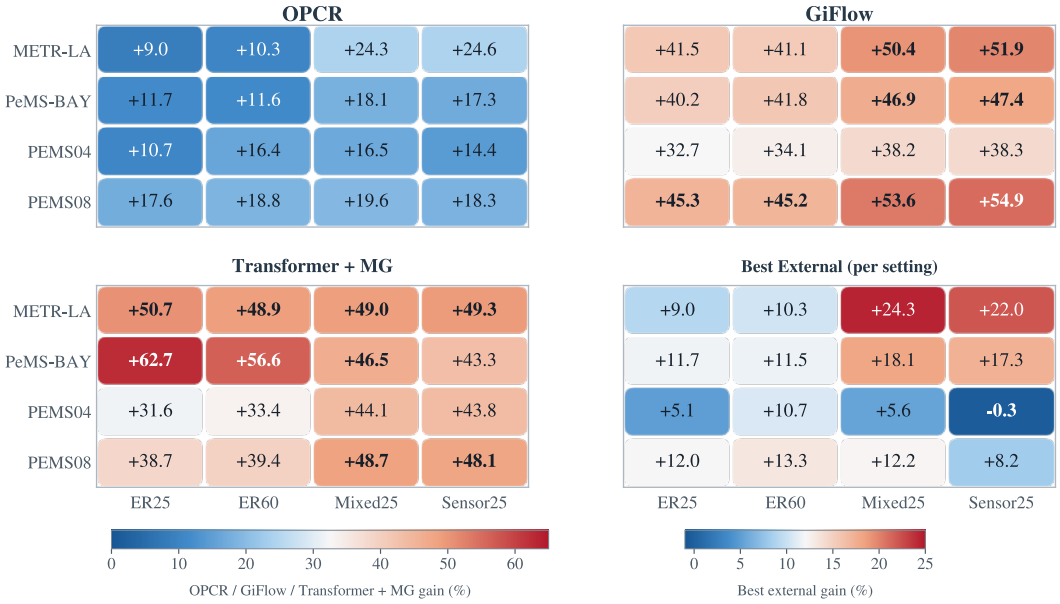


Figure 3: Relative MAE gain of ARCI-T over OPCR, GiFlow, Transformer + MG (Meta Guidance), and the best external baseline in each dataset–protocol setting. Transformer + MG denotes the Transformer baseline equipped with Meta Guidance. Best External is selected separately for each setting as the non-ARCI-T method with the lowest reported MAE among all compared methods available for that setting in Tables 1–4 and Appendix D, and may therefore refer to different methods across settings. Positive values favor ARCI-T. Full results are given in Tables 1–4 and Appendix D.

Figure 3 compares ARCI-T with three representative external methods, OPCR, GiFlow, and Transformer + MG, and the best external result in each dataset–protocol setting. ARCI-T achieves positive relative gains in nearly all settings, with only a slight inversion on PEMS04 Sensor25, where BRITS is marginally better.

4.3 Ablation Studies

Does anchor-relative residual parameterization help? Parameterizing the learned branch in ARCI-T as an anchor-relative residual, rather than directly predicting absolute values, reduces imputation error across all four matched settings. The complete comparison is shown in Table 5.

Table 5: Matched comparison of Direct and ARP, where Δ =ARP−Direct.

Setting	Direct MAE	ARP MAE	Δ
PeMS-BAY Mixed25	1.5961±0.0193	1.5511±0.0089	-0.0450
PeMS-BAY Sensor25	1.7754±0.0129	1.7433±0.0201	-0.0321
PEMS04 Mixed25	21.6503±0.5730	20.7717±0.2206	-0.8785
PEMS04 Sensor25	22.9428±0.5992	21.9530±0.1923	-0.9898

Keeping the deterministic recovery only as an encoder input is also insufficient: removing the explicit output anchor increases macro-average MAE on both analyzed datasets (Appendix C.7).

We additionally evaluate ARCI-T with NearestFill and SeasonalProfile as alternative deterministic recovery references under their corresponding matched training configurations. ARCI-T with the NearestFill anchor improves upon its corresponding deterministic recovery, as does the Seasonal-Profile variant (Figure 4). Across the tested deterministic recovery priors, ARCI-T performs well with several recovery references.

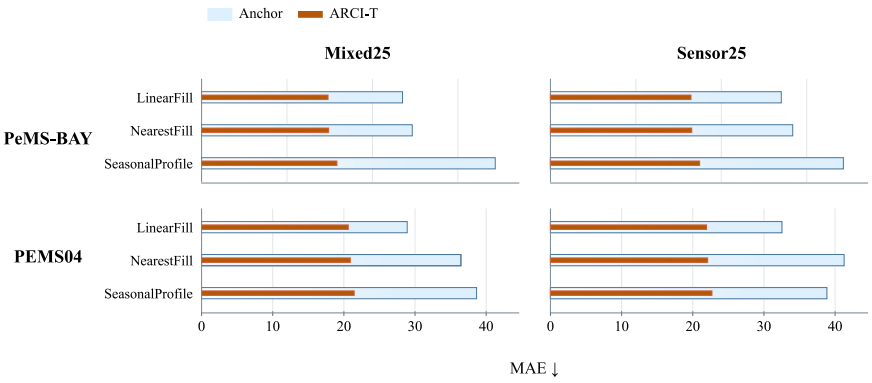


Figure 4: MAE of ARCI-T under different deterministic recovery references. Full experiment results are given in Appendix C.8.

Does position-conditioned acceptance improve imputation? Position-specific evidence matters. Table 6 compares ARCI-T with the two realization endpoints and several acceptance-control variants. Anchor ($G = 0$) and All-R ($G = 1$) correspond to no correction and full acceptance, respectively; Const-G replaces the position-dependent acceptance with a fixed global strength, Shuffle-G preserves the gate-value distribution while breaking its correspondence with specific positions, and Generic-G replaces the proposed evidence-conditioned controller with a generic learned gate. Anchor evaluates the deterministic endpoint. All-R, Const-G, and Shuffle-G intervene on the corresponding frozen Full checkpoint. All-R realizes $A + R_{\theta_{\text{full}}}$. Here, θ_{full} denotes the frozen learned parameters of the matched full ARCI-T model for the corresponding dataset–protocol–seed run, and $R_{\theta_{\text{full}}}$ denotes its residual proposal. Generic-G and the other learned-gate controls are independently trained under matched configurations. ARCI-T achieves the lowest cross-protocol macro-average MAE among these variants on both datasets.

The complete matched learned-gate comparison in Appendix C.4 (Table 17) further shows that ARCI-T obtains the lowest mean MAE across all eight dataset–protocol settings when compared with Generic-G and Self-G. Additional controlled-realization comparisons are reported in Appendix C.5 (Table 18), while the contribution of the position descriptor q and other supporting-component ablations is examined in Appendix C.7 (Tables 20, 21, 22, and 23). Supplementary protocol-level geometry analysis is provided in Appendix C.9 (Table 27). Finally, Appendix B.9 (Table 12) reports model-capacity information together with a separate parameter-matched spatial-only gate comparison.

Table 6: Ablation studies.

	Realization variants				Learned acceptance	PBRA
Dataset	Anchor	All-R	Const-G	Shuffle-G	Generic-G	ARCI-T
PeMS-BAY	1.7031	2.3708	1.4281	1.4643	1.2205	**1.2044**
PEMS04	24.5091	33.7752	20.6528	21.3169	19.7553	**19.4937**

4.4 Robustness under Heavier Missingness and Temporal Distance

Beyond the standard baseline protocols, we further examine robustness under heavier missingness pressure and increasing temporal distance from the training period. To evaluate temporal-distance robustness, we partition each chronologically ordered test sequence into four consecutive bins, T1–T4, with increasing temporal distance from the training cutoff. The model is trained on the predefined training set and then frozen; all bins are evaluated without retraining or online adaptation. This evaluation characterizes performance across the predefined temporal-distance bins. These experiments reuse the established evaluation pipeline; detailed protocols and complete results are provided in Appendix Sections D.2 and D.3.

Table 7: Robustness under heavier missingness and temporal distance. (a) Mixed40/Sensor40 pressure tests; (b) frozen-checkpoint MAE across temporal gap bins from T1 to T4.

(a) Missingness pressure

Dataset	Regime	Anchor	All-R	Shuffle-G	ARCI-T
PeMS-BAY	Mixed40	2.6123±0.0239	2.8386±0.1924	2.2081±0.0337	**1.6451 ± 0.0033**
PeMS-BAY	Sensor40	5.3210±0.1414	3.2825±0.4430	3.2764±0.4457	**3.2628 ± 0.4533**
PEMS04	Mixed40	32.6072±0.3014	44.2974±4.5615	27.4743±0.4187	**23.1204 ± 0.2029**
PEMS04	Sensor40	105.2898±3.3993	44.9412±5.8759	40.2989±3.7980	**40.0085 ± 3.6151**

(b) Temporal gap evaluation

Dataset	Method	T1	T2	T3	T4
PeMS-BAY	Linear	1.5487	1.6682	1.8561	1.7402
	ARCI-T	**1.1171**	**1.1532**	**1.2886**	**1.2590**
PEMS04	Linear	22.2608	27.4455	22.1510	26.2156
	ARCI-T	**18.7413**	**20.9433**	**18.1723**	**20.1287**

Under Mixed40, ARCI-T substantially improves upon the deterministic Anchor. Among the realization variants compared in these pressure tests, ARCI-T has the lowest reported mean MAE. Under Sensor40, the Anchor deteriorates sharply and learned correction becomes much more important.

Using frozen checkpoints without retraining or online adaptation, ARCI-T retains lower MAE than Linear in every temporal gap bin on both datasets. Complete uncertainty, degradation statistics, and additional variants are reported in Appendix D.3.

5 Conclusion

ARCI-T approaches traffic imputation by retaining a deterministic recovery as the reference, learning an anchor-relative correction, and controlling how much of that correction is accepted based on position-specific evidence. Across four traffic datasets and random and structured missingness protocols, ARCI-T achieves leading or highly competitive imputation accuracy. The mechanism analyses and stress evaluations further support the effectiveness of controlled correction across the tested settings. Our current validation focuses primarily on deterministic recovery anchors and traffic sensor data. Future work will investigate a broader set of anchoring models and extend controlled correction to more diverse time-series domains.

Reproducibility Statement

To support reproducibility, the submission includes an anonymized reference implementation of the core ARCI-T architecture, representative configurations, and lightweight verification scripts, while detailed implementation and experimental settings are documented in the appendix. The complete training and evaluation pipeline will be released upon publication.

AI Use Statement

Generative AI tools were used to assist with manuscript polishing, related-work discovery, and limited experimental execution. All scientific content, experimental results, and final claims were reviewed and verified by the authors.

References

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. doi: 10.1007/s10208-009-9045-5.

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*, volume 31, 2018.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the gaps: Multivariate time series imputation by graph neural networks. In *International Conference on Learning Representations*, 2022.

Leyan Deng, Defu Lian, Chenwang Wu, and Enhong Chen. Learning from highly sparse spatiotemporal data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 94022–94046, 2024. doi: 10.52202/079017-2982.

Wenjie Du, David Côté, and Yan Liu. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023. doi: 10.1016/j.eswa.2023.119619.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51 (3):455–500, 2009. doi: 10.1137/07070111X.

Mingxi Li, Zhengmin Shi, and Wei Ma. Spatiotemporal traffic prediction with missing data: A self-imputation assisted prediction model. *Transportation Research Interdisciplinary Perspectives*, 37: 101937, 2026. doi: 10.1016/j.trip.2026.101937.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. PriSTI: A conditional diffusion framework for spatiotemporal imputation. In *IEEE 39th International Conference on Data Engineering*, pp. 1927–1939, 2023. doi: 10.1109/ICDE55515.2023.00150.

Zijin Liu, Xiang Zhao, and You Song. RDPI: A refine diffusion probability generation method for spatiotemporal data imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12255–12263, 2025. doi: 10.1609/aaai.v39i12.33335.

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023.

Xiaowei Mao, Huihu Ding, Yan Lin, Tingrui Wu, Shengnan Guo, Dazhuo Qiu, Feiling Fang, Jilin Hu, and Huaiyu Wan. Spatial-temporal feedback diffusion guidance for controlled traffic imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15528–15536, 2026. doi: 10.1609/aaai.v40i18.38581.

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. doi: 10.1145/3637528.3671751.

Cang Qin, Lina Yang, Ling Peng, Wenyue Zhang, and Meng Wei. STMoE: A structured mixture-of-experts framework for robust spatiotemporal series imputation. *Expert Systems with Applications*, 319:132137, 2026. doi: 10.1016/j.eswa.2026.132137.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 914–921, 2020. doi: 10.1609/aaai.v34i01.5438.

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*, volume 34, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yiming Wang, Hao Peng, Senzhang Wang, Haohua Du, Chunyang Liu, Jia Wu, and Guanlin Wu. STAMImputer: Spatio-temporal attention MoE for traffic data imputation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3435–3443, 2025. doi: 10.24963/ijcai.2025/382.

Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. ST-MVL: Filling missing values in geo-sensory time series data. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2704–2710, 2016.

Jiacheng You, Xinyang Chen, Yu Sun, Weili Guan, and Liqiang Nie. Long short-term imputer: Handling consecutive missing values in time series. *Transactions on Machine Learning Research*, 2025a.

Jiacheng You, Xinyang Chen, Yu Sun, Weili Guan, and Liqiang Nie. Meta guidance: Incorporating inductive biases into deep time series imputers. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Peng Yuan, Yiwen Jiao, Jiaxue Li, and Yan Xia. A densely connected causal convolutional network separating past and future data for filling missing PM2.5 time series data. *Heliyon*, 10(2):e24738, 2024. doi: 10.1016/j.heliyon.2024.e24738.

Zepeng Zhang, Aref Einizade, Jhony H. Giraldo, and Olga Fink. Spatiotemporal imputation with graph-informed flow matching. In *International Conference on Machine Learning*, 2026.

A Additional Method and Conceptual Analysis

A.1 Why Three Responsibilities

The controlled-correction form rests on a separation of responsibilities rather than on new universal assumptions. The deterministic recovery reference remains in the output, so the learned component never replaces the reference; the residual proposal determines the content of the correction; and the bounded acceptance determines its realized proportion from position-specific evidence. The acceptance coefficient is a bounded control variable, and the proposal itself is not clipped, so the form does not claim a global magnitude bound on corrections. This separation gives each responsibility a distinct direct information path: the proposal is computed from the spatial state, while the acceptance is computed from temporal, spatial, missingness-geometry, and robust–light evidence without reading the proposal.

A.2 Controlled-Correction Semantics

The final output combines the recovery reference, the residual proposal, and the acceptance coefficient through $X_{\text{raw}} = A + G \odot R$ and $X_{\text{final}} = M \odot X_{\text{obs}} + (1 - M) \odot X_{\text{raw}}$ (Equations (8)–(9)). Three semantic properties follow directly from this form. First, the recovery reference is a retained endpoint: wherever the acceptance coefficient is zero, the output returns exactly to the deterministic reference, so the learned component can never move the output away from the reference without a realized proposal. Second, the learned output has proposal semantics: R describes how the reference should change rather than an absolute estimate, and $A + R$ is the full-correction endpoint that neither the final output nor the acceptance controller reads. Third, the acceptance is bounded and position-conditioned: $G_{b,t,n} \in [0, 1]$ controls the accepted proportion of the proposal at each position from the evidence path $E = [H_{\text{temporal}} \| H_{\text{spatial}} \| q \| \Delta H]$. The acceptance coefficient is therefore a control variable for the extent of the correction, not a confidence, probability, or calibrated uncertainty; the proposal and the acceptance are computed from different direct information paths. The proposal–acceptance decomposition is an architectural factorization rather than an identifiable latent-variable model; we do not claim that R and G are uniquely determined from the training objective alone. Their semantics follow from the imposed information paths, bounded acceptance interface, and matched interventions.

A.3 Comparison on Shared Conceptual Axes

Residual refinement around an initial estimate and adaptive control of a learned quantity are both established mechanisms in time-series imputation. DCCN-SPF uses a linear interpolation S as the reference, learns a deviation T, and outputs $S+T$; RDPI refines an initial imputation through a diffusion process; SAITS combines two learned estimates with missing-aware attention-related weights; OPCR propagates information by pointwise confidence estimated from observation relations; and FENCE adjusts the diffusion guidance strength during generation from feedback and cluster-level correlations. ARCI-T organizes the learned correction between an explicitly retained deterministic reference and a position-conditioned bounded acceptance: the deterministic reference always stays in the output, the correction is proposed relative to it, and a position-conditioned acceptance coefficient decides how much of the correction is added to A, with the acceptance path not reading the proposal itself.

Table 8 compares these forms on shared conceptual axes. The comparison follows interface semantics rather than terminology or shared components: a missing term does not imply a missing mechanism, and a shared component does not imply an identical interface.

Table 8: Technical comparison of representative refinement and adaptive-control forms on shared conceptual axes; sharing basic components such as residual refinement, bounded weighting, or adaptive control does not imply identical forms. The FENCE row reflects its published formulation (Mao et al., 2026).

Method	Reference / initial estimate	Refined object	Adaptive control	Controlled quantity	Imputation form
DCCN-SPF	Linear interpolation S, enters the final output	Learned deviation T relative to S	Fixed addition in the main model; a separately trained comparator exists	Main output $S + T$; no subsequent position-level bounded acceptance	Interpolation plus learned deviation
RDPI	Initial imputation estimate	Deviation relative to the initial estimate, modeled by diffusion	Conditional diffusion refinement process	Refinement residual added to the initial imputation	Residual refinement around an initial estimate
SAITS	Two learned estimates from attention outputs	Position-dependent combination of the two estimates	Missing-aware, attention-related weights	Fusion of learned estimates	Self-attention imputation with adaptive estimate fusion
OPCR	Learned spatial and temporal initial imputation	Iterative message refinement of the initial estimate	Pointwise confidence estimated from observation relations	Propagation strength during refinement	Two-stage imputation with guided propagation
FENCE	Imputation estimate refined through generation	Generated samples conditioned on observations	Feedback and cluster-level correlations adjust the diffusion guidance strength	Diffusion guidance scale during generation	Controlled generative refinement
ARCI-T	LinearFill A, retained in the output as a deterministic endpoint	Anchor-relative correction $R = f_R(H_{\text{spatial}})$	Position evidence (temporal, spatial, q, ΔH) produces $G \in [0, 1]$	Realized correction $G \odot R$ added to A; the acceptance path does not read R or $A + R$	Controlled correction around a retained deterministic reference

B Implementation and Reproducibility

B.1 Forward Computation and Tensor Conventions

The model receives $X_{\text{obs}}, M \in \mathbb{R}^{B\times T\times N}$ and calendar features $C \in \mathbb{R}^{B\times T\times d_c}$. Before entering the modules, M is binarized with threshold 0.5, and the sensor axis of C is obtained by broadcasting. Algorithm 1 summarizes the forward process under robust-only routing, raw correlation, and an adaptive position gate. The acceptance network and the residual head are logically parallel after sharing H_{spatial}; the algorithm evaluates the acceptance network first according to the implementation evaluation order, but there is no output dependency between them.

In the tensor expressions, $\odot$ denotes elementwise multiplication, and $\|$ denotes concatenation along the feature dimension.

Algorithm 1 ARCI-T forward computation

Require: X_{obs}, M, C; frozen graph $\mathcal{G}_{\text{corr}}$
Ensure: X_{final}
$M \leftarrow \mathbb{I}[M > 0.5]$
$A \leftarrow \text{LinearFill}(X_{\text{obs}}, M)$
$q \leftarrow \text{GapDescriptor}(M)$
$H_{\text{light}} \leftarrow f_{\text{light}}(A, M, C)$
$H_{\text{robust}} \leftarrow f_{\text{robust}}(A, M, q, C)$
$H_{\text{temporal}} \leftarrow H_{\text{robust}}$
$H_{\text{spatial}} \leftarrow f_{\text{spatial}}(H_{\text{temporal}}; \mathcal{G}_{\text{corr}})$
$\Delta H \leftarrow H_{\text{robust}} - H_{\text{light}}$
$\widetilde{G} \leftarrow \text{clip}_{[0,1]}(\sigma(f_G([H_{\text{temporal}} \| H_{\text{spatial}} \| q \| \Delta H])))$
$R \leftarrow \text{squeeze}_{-1}(f_R(H_{\text{spatial}}))$
$G \leftarrow \text{squeeze}_{-1}(\widetilde{G})$
$X_{\text{raw}} \leftarrow A + G \odot R$
$X_{\text{final}} \leftarrow M \odot X_{\text{obs}} + (1 - M) \odot X_{\text{raw}}$
return X_{final}

B.2 LinearFill Boundary Definition

For sample b and sensor n, let

$$\mathcal{O}_{b,n} = \{s \in \{1, \dots, T\} : M_{b,s,n} = 1\}$$

be the set of visible indices in the current window. For a missing position t, let $\ell_{b,t,n}$ and $r_{b,t,n}$ denote the nearest visible indices strictly to the left and right of t (absent when the corresponding set is empty), and write $x_\ell = X_{\text{obs},b,\ell_{b,t,n},n}$ and $x_r = X_{\text{obs},b,r_{b,t,n},n}$. The formal recovery reference is

$$A_{b,t,n} = \begin{cases} X_{\text{obs},b,t,n}, & M_{b,t,n} = 1, \\ x_\ell + \dfrac{t - \ell_{b,t,n}}{r_{b,t,n} - \ell_{b,t,n}}(x_r - x_\ell), & M_{b,t,n} = 0,\ \ell_{b,t,n}, r_{b,t,n} \text{ both exist}, \\ x_\ell, & M_{b,t,n} = 0,\ \ell_{b,t,n} \text{ exists and } r_{b,t,n} \text{ does not}, \\ x_r, & M_{b,t,n} = 0,\ r_{b,t,n} \text{ exists and } \ell_{b,t,n} \text{ does not}, \\ 0, & \mathcal{O}_{b,n} = \varnothing. \end{cases} \tag{11}$$

The implementation first computes a window-visible mean with a bounded denominator as the fallback value; this branch is selected only when both sides lack support, in which case $\mathcal{O}_{b,n} = \varnothing$, the visible count is zero, and the value is exactly 0. All operations are performed independently along the time axis, and no fill value is shared across batches or sensors.

B.3 GapDescriptor Definition

GapDescriptor reads only M and outputs $q \in \mathbb{R}^{B \times T \times N \times 9}$. Left and right support both use strict indices; distances and gap length are normalized by T. Missing support uses distance T as a sentinel value; distances, gap length, and the three support flags are zero at observed positions. The local radius is formally 6, and boundary positions only use the valid neighborhood length inside the window as the denominator. The nine components are given in Table 9.

B.4 Multi-Scale Trend, Fluctuation, and Temporal States

For a requested window w, the implementation first takes the effective width

$$\widetilde{w} = \min\{\max\{1, w\}, T\}, \qquad \ell_{\widetilde{w}} = \left\lfloor \frac{\widetilde{w} - 1}{2} \right\rfloor, \qquad r_{\widetilde{w}} = \widetilde{w} - 1 - \ell_{\widetilde{w}}.$$

Let $\pi_T(s) = \min\{\max\{s, 1\}, T\}$ be the replicated index extension; then AvgPool1D with stride 1 is equivalent to

$$\big[\text{MA}_w(A)\big]_{b,t,n} = \frac{1}{\widetilde{w}} \sum_{u=0}^{\widetilde{w}-1} A_{b,\pi_T(t+u-\ell_{\widetilde{w}}),n}. \tag{12}$$

Table 9: Deterministic definition of the nine-dimensional GapDescriptor. Distances use strict left/right support; the formulas apply to missing positions, with distances, gaps, and support flags zeroed at observed positions.

Component	Meaning	Definition and boundary
q_1	Left distance	$(t-\ell_{b,t,n})/T$ if left support exists, otherwise 1.
q_2	Right distance	$(r_{b,t,n}-t)/T$ if right support exists, otherwise 1.
q_3	Gap length	$(r^\star_{b,t,n}-\ell^\star_{b,t,n}-1)/T$; The starred indices equal $\ell_{b,t,n}$ and $r_{b,t,n}$ when the corresponding support exists; otherwise $\ell^\star_{b,t,n}=0$ and $r^\star_{b,t,n}=T+1$, respectively.
q_4	Local observation rate	$\sum_{s\in\mathcal{J}_t} M_{b,s,n}/\lvert\mathcal{J}_t\rvert$, where $\mathcal{J}_t = [t-6, t+6]\cap\{1,\dots,T\}$.
q_5	Sensor-window observation rate	$T^{-1}\sum_{s=1}^{T} M_{b,s,n}$.
q_6	Left support flag	1 if left support exists and the position is missing, otherwise 0.
q_7	Right support flag	1 if right support exists and the position is missing, otherwise 0.
q_8	Bilateral support flag	1 if both left and right support exist and the position is missing, otherwise 0.
q_9	Nearest support distance	$\min\{d^{\text{raw}}_{\text{L}}, d^{\text{raw}}_{\text{R}}\}/T$; Here, $d^{\text{raw}}_{\text{L}} = t-\ell_{b,t,n}$ if left support exists, and $d^{\text{raw}}_{\text{R}} = r_{b,t,n}-t$ if right support exists; each absent-side distance is set to T.

This definition covers both odd and even windows; even widths pad one more position on the right than on the left. The formal set $\mathcal{W}=\{3,7,13,25\}$ contains only odd values and $T=200$, so window trimming is never triggered.

We use $\mathcal{W}$ for the set of candidate window widths and w for an individual width; the fixed-window ablation uses $w=3$.

Projected calendar features are broadcast along the sensor axis as $C'_{b,t,n,:}$, and the scale network reads $[q_{b,t,n,:}\|C'_{b,t,n,:}]$ at each position. If $z^\alpha = f_\alpha([q\|C'])$, then

$$\alpha_{b,t,n,w} = \frac{\exp(z^\alpha_{b,t,n,w}/\tau)}{\sum_{v\in\mathcal{W}}\exp(z^\alpha_{b,t,n,v}/\tau)}, \qquad \sum_{w\in\mathcal{W}}\alpha_{b,t,n,w}=1, \tag{13}$$

$$\text{Trend}_{b,t,n} = \sum_{w\in\mathcal{W}}\alpha_{b,t,n,w}\big[\text{MA}_w(A)\big]_{b,t,n}, \tag{14}$$

$$\text{Fluctuation}_{b,t,n} = A_{b,t,n} - \text{Trend}_{b,t,n}\,. \tag{15}$$

Here $\alpha\in\mathbb{R}^{B\times T\times N\times|\mathcal{W}|}$, the softmax acts only on the final scale axis, and the formal temperature is $\tau=1$. The trend branch encodes [Trend, M] with a Transformer, and the fluctuation branch encodes [Fluctuation, M] with a per-sensor temporal convolution; both add the projected calendar features. Writing the corresponding states as H_{trend} and H_{fluct}, the two-component softmax weights γ give

$$H_{\text{robust}} = \gamma_1\odot H_{\text{trend}} + \gamma_2\odot H_{\text{fluct}}, \qquad \gamma_1+\gamma_2=1. \tag{16}$$

At each (b,t,n) position, γ is generated by applying a LayerNorm–Linear–GELU–Linear map to $[H_{\text{trend}}\|H_{\text{fluct}}]$, followed by a two-way softmax over the component axis. Thus $\gamma\in[0,1]^{B\times T\times N\times 2}$ is broadcast over the hidden channels when the two component states are combined at the same position. LightExpert applies LayerNorm and a per-sensor temporal convolution to the sum of the Anchor, mask, and calendar embeddings. The formal model uses robust-only temporal routing, so $H_{\text{temporal}}=H_{\text{robust}}$; H_{light} is retained only as the contrast state of the acceptance controller.

B.5 Training-Set Frozen CorrTopK

Let $\mathcal{I}_{\text{tr}}$ be the set of temporal positions of the flattened normalized training windows, $z_{s,n}$ the value of sensor n at position s, and $\varepsilon > 0$ a numeric lower bound. The graph builder first centers

$$u_{s,n} = z_{s,n} - \frac{1}{|\mathcal{I}_{\text{tr}}|} \sum_{r \in \mathcal{I}_{\text{tr}}} z_{r,n}, \qquad \eta_n = \max\left\{ \sqrt{\sum_{s \in \mathcal{I}_{\text{tr}}} u_{s,n}^2}, \varepsilon \right\},$$

and then computes

$$\rho_{n,j} = \frac{\sum_{s \in \mathcal{I}_{\text{tr}}} u_{s,n} u_{s,j}}{\eta_n \eta_j}. \tag{17}$$

Inputs are first required to be finite; zero variance is handled by the ε lower bound, and any remaining non-finite correlations are set to zero. In the formal absolute mode, for $1 \le K \le N$, let $\mathcal{N}_K(n)$ denote the K source-node indices selected for target node n by the absolute-correlation scores $|\rho_{n,j}|$, with the diagonal score set to positive infinity to include n itself. The selected unnormalized weights are

$$\widetilde{a}_{n,j} = \begin{cases} 1, & j = n, \\ |\rho_{n,j}|, & j \ne n, \end{cases} \qquad a_{n,j} = \frac{\widetilde{a}_{n,j}}{\max\{\sum_{k \in \mathcal{N}_K(n)} |\widetilde{a}_{n,k}|, \varepsilon\}}.$$

The formal weights are therefore nonnegative and each row sums to 1. Row n is the target node receiving messages, and $j \in \mathcal{N}_K(n)$ is the source node; the aggregation at the same temporal position is

$$m_{b,t,n,:} = \sum_{j \in \mathcal{N}_K(n)} a_{n,j} H_{\text{temporal},b,t,j,:}. \tag{18}$$

Let $p_{b,t,n,:} = [H_{\text{temporal},b,t,n,:} \| m_{b,t,n,:}]$, with f_{sg} and f_{msg} the spatial acceptance gating MLP and message MLP, and LN LayerNorm; the direct spatial refinement is

$$H_{\text{spatial},b,t,n,:} = \text{LN}(H_{\text{temporal},b,t,n,:} + \sigma(f_{\text{sg}}(p_{b,t,n,:})) \odot f_{\text{msg}}(p_{b,t,n,:})). \tag{19}$$

The graph is constructed once from the complete training labels; validation and test data are not involved, and the indices and weights are then saved as persistent buffers and frozen during validation and testing. With flattened training length L_{tr}, the correlation matrix costs $O(L_{\text{tr}} N^2)$, online sparse aggregation costs $O(BTNKd_h)$, and graph storage is $O(NK)$.

B.6 Residual Head, Acceptance Controller, and Output Dimensions

The residual head applies LayerNorm, a linear map, GELU, Dropout, and a final linear map whose output has one channel to H_{spatial}. In the implementation this first produces $B \times T \times N \times 1$; the final singleton channel is then squeezed to obtain $R \in \mathbb{R}^{B \times T \times N}$ in the main-text notation.

The acceptance controller input is strictly

$$Z_G = [H_{\text{temporal}} \| H_{\text{spatial}} \| q \| (H_{\text{robust}} - H_{\text{light}})] \in \mathbb{R}^{B \times T \times N \times (3d_h + d_q)}.$$

The formal dimensions are $d_h = 64$ and $d_q = 9$, giving input width 201. Here, $Z_G \equiv E$ denotes the acceptance evidence defined in the main text. GateMLP sequentially applies LayerNorm, a linear map to 64 dimensions, GELU, Dropout, and a linear map to 1 dimension, producing

$$\widetilde{G} = \text{clip}_{[0,1]}(\sigma(f_G(Z_G))) \in [0,1]^{B \times T \times N \times 1}, \qquad G = \text{squeeze}_{-1}(\widetilde{G}).$$

The sigmoid already bounds the output in the open interval $(0, 1)$, and the explicit clip is a numerical safeguard. Then A, G, and R are all $B \times T \times N$, and Equation (8) does not rely on implicit broadcasting. The residual head's direct input is only H_{spatial}; the acceptance controller does not read R or $A + R$, and LightExpert does not enter the spatial refinement or the residual head.

B.7 Training Masks and Loss Definition

Let $U \in \{0,1\}^{B \times T \times N}$ be the artificially hidden supervision mask of the training interface (1 means supervision is required) and V the valid-label mask. The supervised positions and adjacent temporal pairs are

$$\mathcal{S} = \{(b,t,n) : U_{b,t,n} = 1,\ V_{b,t,n} = 1\}, \tag{20}$$

$$\mathcal{P} = \{(b,t,n) : 2 \le t \le T,\ V_{b,t-1,n} = V_{b,t,n} = 1,\ U_{b,t-1,n} = 1 \ \vee\ U_{b,t,n} = 1\}. \tag{21}$$

For an error e, the PyTorch Huber definition corresponds to

$$\ell_\delta(e) = \begin{cases} \frac{1}{2}e^2, & |e| \le \delta, \\ \delta(|e| - \frac{1}{2}\delta), & |e| > \delta. \end{cases}$$

Writing $\widehat{X} = X_{\text{final}}$, the two implemented means follow these empty-set conventions:

$$\mathcal{L}_{\text{imp}} = \begin{cases} \dfrac{1}{|\mathcal{S}|} \sum_{(b,t,n)\in\mathcal{S}} \ell_\delta(\widehat{X}_{b,t,n} - X_{b,t,n}), & |\mathcal{S}| > 0, \\ 0, & |\mathcal{S}| = 0, \end{cases} \tag{22}$$

$$\mathcal{L}_{\Delta} = \begin{cases} \dfrac{1}{|\mathcal{P}|} \sum_{(b,t,n)\in\mathcal{P}} \left| (\widehat{X}_{b,t,n} - \widehat{X}_{b,t-1,n}) - (X_{b,t,n} - X_{b,t-1,n}) \right|, & |\mathcal{P}| > 0, \\ 0, & |\mathcal{P}| = 0. \end{cases} \tag{23}$$

Empty sets return a zero term connected to the computation graph, with no division by zero. The 80 main-benchmark runs and the controlled-realization, Direct/ARP, multi-anchor, and learned-gate experiments all use $\delta = 1$, $\lambda_\Delta = 0.1$, and the temporal-difference term enabled. Validation and checkpoint selection for the Main configuration use missing-position-only normalized MAE; this choice is a validation-selection detail and does not mean the training objective is MAE-only. Matched configurations select models by the lowest validation loss; the two training configurations are given in Table 11.

Temporal splitting precedes non-overlapping windowing with length 200, and training statistics are used for per-sensor normalization; training and validation masks use seed offsets $1{,}000{,}003$ and $2{,}000{,}003$, respectively, and test masks are loaded from fixed paths and hashes. The formal training seeds are 2021–2025. CorrTopK reads only the normalized complete training labels. The external evaluator evaluates only artificially hidden and originally valid positions after inverse scaling; originally invalid values and visible values do not enter the error, and visible values are preserved exactly by the observation clamp.

The canonical mask generator used by the released data adapter samples an entry-random mask by drawing each position independently with NumPy's `default_rng(seed)` and hiding 25% (ER25) or 60% (ER60) of positions. For Mixed25, it first hides 5% of positions independently and then samples batch–sensor pairs and temporal blocks until the hidden count reaches the integer target round$(0.25BTN)$; each block length is an integer drawn uniformly from 12 through min$(48, T)$ and its start is uniform over valid in-window starts. Sensor25 uses the same block sampler without the initial 5% entry-random step. Blocks are placed whole and never truncated. Sampling stops when the number of unique hidden positions first reaches or exceeds round$(0.25BTN)$; any overshoot therefore comes only from the final whole block and is fewer than 48 positions, while overlapping draws add no new hidden positions. These rules operate on the window tensor and do not encode originally invalid entries; those entries are excluded later by the label-validity mask. Training and validation use the same protocol family with the stated seed offsets, whereas the benchmark test masks are frozen artifacts referenced by the mask manifest and verified by SHA-256 hashes. Mixed40 uses `np.random.default_rng(training_seed)` and first hides 8% of positions independently, then samples batch–sensor pairs and whole contiguous temporal blocks with integer lengths drawn uniformly and inclusively from 12 through 48. Blocks are added until the hidden count reaches or exceeds round$(0.40BTN)$; the final block is never truncated, so the realized hidden fraction can slightly exceed the 0.40 target. The resulting mask is deterministic conditional on the seed and therefore reproducible seed-by-seed. Mixed40 is evaluated with frozen Mixed25 mechanism-analysis checkpoints, without retraining or online adaptation. Entry40 and Sensor40 are also evaluation-only stress masks applied to frozen mechanism-analysis checkpoints without retraining. Entry40 hides each test position independently with probability 0.40. Sensor40 randomly selects exactly round$(0.40N)$ sensors and hides them for the entire test horizon. Mask seeds are deterministically derived from the dataset, training protocol, training seed, and stress-mask type; each run records the mask definition, realized fraction, and SHA-256, and all compared realization schemes share the same stress mask.

B.8 Training Configurations and Result Correspondence

The main benchmark and the mechanism analyses share the training objective, optimizer, batch size, and epoch budget, but differ in stopping and checkpoint-selection rules; they therefore cor-

respond to two independently trained and frozen configuration families. The main-benchmark configuration early-stops by the missing-position-only normalized MAE on the validation set, while the mechanism-analysis configuration uses a fixed budget and the lowest validation loss. ARCI-T in Tables 1–4 and Table 6 comes from these two families, respectively. Generic-G, Self-G, Global-G, and Shared-E use the matched mechanism-analysis configuration described below. The mapping between reported positions and training configurations is given in Table 10, and the two configurations are given in Table 11; all formal ARCI-T training uses dropout 0.1 and no scheduler. The main benchmark uses gradient clipping with norm 5; the mechanism-analysis configuration uses no gradient clipping.

B.9 Model Capacity and Parameter-Matched Comparison

We characterize model capacity using trainable parameter counts computed from the formal model source without a forward pass. Table 12 reports both acceptance-network and full-model parameter counts. The wider evidence-conditioned input accounts for ARCI-T's additional acceptance-network capacity relative to Generic-G.

Matched-capacity control. To separate evidence conditioning from controller capacity, we also compare ARCI-T with a spatial-only acceptance gate widened to the same total model capacity (123,055 trainable parameters). In this separate matched campaign, ARCI-T achieves lower macro-average MAE on PeMS-BAY (1.1697 vs. 1.1831) and PEMS04 (18.7901 vs. 18.9953), with 5/5 paired seed-level macro wins on each dataset. Because the campaign uses its own matched Full reference, its values are kept separate from Table 6.

B.10 Centralized Implementation Configuration

Data versions: METR-LA and PeMS-BAY use the speed benchmark released with DCRNN (Li et al., 2018) and the standard temporal split, and PEMS04 and PEMS08 use the community-standard traffic flow version used by STSGCN (Song et al., 2020); all data are at five-minute resolution. The complete configuration is given in Table 13. The common audited evaluation contract for the in-scope ARCI-T and matched mechanism analyses uses strict chronological splits, train-only per-sensor scalers, canonical masks, an original-scale missing-only evaluator on originally valid and artificially hidden positions, and an exact observed- value clamp. Test hidden values are used only for metric computation, not for training, tuning, or checkpoint selection; the detailed objective and mask definitions are in Appendix Sections B.7–B.8 and Tables 10–11.

Baseline provenance. The deterministic Mean, KNN, and Linear baselines use frozen train-only statistics and each report 80 result cells over four datasets, four protocols, and five training seeds (2021–2025). CSDI uses the official source snapshot (commit `7f24a436`) without core modifications and likewise reports 80 result cells over four datasets, four protocols, and five training seeds (2021–2025), with validation-only diffusion-loss selection and a raw-scale missing-only evaluator. OPCR (commit `48ce4f83`) and Transformer + MG (commit `76b4820f`) use their official snapshots with matched adapters; each covers 80 dataset–protocol–training-seed evaluations over four datasets, four protocols, and five training seeds (2021–2025), with shared splits, masks, scalers, and evaluator, validation-only checkpoint selection, and no test tuning.

ATTST. The ATTST campaign contains 80 formal dataset–protocol–training-seed runs ($4\times4\times5 = 80$, training seeds 2021–2025). Each run has one formal checkpoint and one canonical test mask, so it has no additional five-mask evaluation axis. Each run is trained for a fixed budget of 3,500 optimizer steps, and the checkpoint with the lowest validation-corruption loss is selected for test evaluation. Its task adapter changes the original missing-data forecasting task to imputation and evaluates the input-aligned self-imputation output rather than the future-prediction head. The campaign is based on ATTST commit `3409066c7807e1cc069829ef904505270a183bc3` with a recorded compatibility patch and frozen source/adapter/evaluator identities.

GiFlow. The GiFlow campaign contains 20 formal training checkpoints (four datasets times five independent training seeds, 2021–2025). Each checkpoint is evaluated under four missingness protocols and five independent canonical evaluation-mask seeds (2021–2025), yielding 80 dataset–protocol–training- seed identities and 400 evaluation records. For each identity, the five mask results

Table 10: Correspondence between reported results and training configurations. The same method name corresponds to independently trained configurations in different analyses.

Analysis	Reported location	Training/evaluation identity	Relation to other analyses
Main benchmark configuration	ARCI-T in the main benchmark tables and the corresponding appendix MSE	Four datasets, four protocols, five training seeds	Trained independently of the mechanism analyses
Matched comparison with external methods	OPCR, Transformer + Meta Guidance, and ARCI-T in the main benchmark tables	Shared splits, masks, seeds, and evaluator; each method keeps its own training configuration	Matched-comparison contract; introduces no new aggregated configuration
Mechanism-analysis configuration	Tables 6 and 20 and the appendix mechanism interventions	Independently trained ARCI-T and component variants on PeMS-BAY/PEMS04	Generic-G and Self-G are paired with ARCI-T by seed
Direct vs. ARP	Table 5 and the complete appendix values	Independent matched pairs with the same backbone, masks, seeds, budget, and evaluator	Its values are not mixed with ARCI-T values of other analyses
Multi-anchor	Figure 4 and appendix Table 26	LinearFill reference corresponds to the main-benchmark ARCI-T; alternative anchors each train matched models	Compares only the tested deterministic priors
Learned-gate alternatives	Macro averages in Table 6 and appendix Table 17	ARCI-T, Generic-G, and Self-G use matched training and five seeds	ARCI-T is the mechanism-analysis configuration
Supplementary controlled-realization comparisons (Global-G, Shared-E)	Appendix Table 18	Matched training under the same contract	Reported as paired-win summaries
Missingness pressure / temporal-distance evaluation	Table 34 and the complete appendix results	Re-evaluates frozen mechanism-analysis checkpoints directly; no retraining or online adaptation	Scalers and CorrTopK remain frozen as at training time

Table 11: Training configurations for the formal ARCI-T results. The two families share the objective and optimizer but differ in stopping and checkpoint rules.

Configuration scope	Training objective	Optimizer / batch size	Training duration	Stopping and checkpoints
Main-benchmark ARCI-T	Huber ($\delta = 1$) on missing positions only, $+0.1\mathcal{L}_\Delta$	Adam, 10^{-3}, weight decay 0; batch size 8	Up to 100 epochs; clipping norm 5; no scheduler	Early stopping with patience 20; lowest validation missing-position-only normalized MAE
Matched mechanism, Direct/ARP, alternative-anchor, and learned-gate runs	Huber ($\delta = 1$) on missing positions only, $+0.1\mathcal{L}_\Delta$	Adam, 10^{-3}, weight decay 0; batch size 8	Fixed 100 epochs; no clipping / scheduler	No early stopping (patience 0); lowest validation loss

Table 12: Trainable parameter counts of the matched learned-gate variants, computed from the same formal source and hidden widths.

Variant	Gate input (width)	Gate params	Total params	vs. Generic-G
Self-G	R (1)	193	109,853	0.9635×
Generic-G	H_{spatial} (64)	4,353	114,013	1.0000×
ARCI-T	Temporal, spatial, q, ΔH (201)	13,395	123,055	1.0793×

are first averaged; protocol summaries then aggregate the five training seeds. For the macro summary, the four protocol values are averaged with equal weight within each training seed before the five seed macros are aggregated. Checkpoints use the unweighted mean of validation-only protocol losses, with test data excluded from selection and scheduling. The campaign records shared strict splits, canonical masks, train-only scalers, a missing-only raw-scale evaluator, and observed-value clamping. Historical training seeds 2022–2025 use source SHA-256 prefix `90b3b23a`, which is not a Git commit; seed 2021 uses the upstream commit `4e7d0375155750f6c3d9387ad68e7c78d99dd0ac`. These runs use the two recorded source/adapter stages specified above.

Other baselines. BRITS, SAITS, Vanilla Transformer, STAMImputer, and ImputeFormer use the benchmark adapters and manifests adopted in this work. For PeMS-BAY, PEMS04, and PEMS08, each method reports 60 result cells over four protocols and five training seeds (2021–2025). For METR-LA, each method reports an additional 20 strict dataset–protocol–seed identities (four protocols × five training seeds, 2021–2025), with sample standard deviation computed using `ddof=1`. Accordingly, each method reports 80 result cells in total across the four datasets. The reported aggregates correspond to the values used in the paper tables.

Input-validity and calendar semantics. In the formal loader, V is constructed as `isfinite(raw)` before any artificial missingness mask is applied. NaN and Inf are therefore invalid; natural invalid entries are excluded from training statistics and filled only to form finite window tensors, while zero remains a valid value. The strict dataset assets have zero natural-missing entries for all four datasets, so $V = 1$ for those assets, but the same validity rule is applied uniformly. With $M = 1$ denoting an exposed observation, the ARCI-T formal runner's training supervision and test scoring use $V \odot (1 - M)$ for point-value imputation. The adjacent-difference term instead uses the valid adjacent pairs defined in Appendix B.7, retaining pairs with at least one artificially hidden endpoint. Observed positions and naturally invalid positions are excluded from point-value scoring, and observed outputs are clamped exactly. The four timestamp-only calendar channels are minute-within-hour divided by 55, hour-of-day divided by 23, weekday divided by 6 (Monday 0 through Sunday 6), and a weekend indicator $\mathbf{1}[\text{weekday} \geq 5]$. They lie in $[0, 1]$, have shape $[\text{window}, 200, 4]$, are broadcast across sensors, and use no target values or target masks.

Configuration and evaluation mapping. The main benchmark and matched mechanism family use the separate selection rules in Tables 10–11; the main family uses validation missing-only normalized MAE, whereas the mechanism family uses a fixed 100-epoch budget and lowest validation loss. Pressure evaluations are inference-only. Each Mixed40 run evaluates one frozen Mixed25 checkpoint; each reported Mixed40 cell aggregates the five corresponding training-seed runs. Each Entry40 or Sensor40 cell aggregates the ten frozen Mixed25/Sensor25 checkpoints for that dataset, with no retraining or online adaptation. Temporal-distance evaluation reuses the same protocol–seed checkpoint, scaler, graph, and canonical mask across T1–T4, without refitting or adaptation. These mappings are detailed with the pressure and temporal-distance protocols in Appendix Sections D.2 and D.3; evaluator implementations remain campaign-scoped.

Table 13: Centralized implementation configuration of ARCI-T. The forward configuration is identical across the 80 main-benchmark runs; differences between the two training families are given in Table 11.

Configuration item	Formal value
Input window length	$T = 200$
Input / calendar dimensions	$d_{\text{in}} = 1$, $d_c = 4$
Hidden and feed-forward dimensions	$d_h = 64$, $d_{\text{ff}} = 128$
Multi-scale temporal windows	$\mathcal{W} = \{3, 7, 13, 25\}$
GapDescriptor	$d_q = 9$, local radius 6
Formal temporal routing	robust-only routing, i.e., $H_{\text{temporal}} = H_{\text{robust}}$
CorrTopK	raw Pearson correlation, with $K = 8$ including self-loops
Acceptance mapping	adaptive position gate; input width $3d_h + d_q = 201$
Formal training objective	Huber ($\delta = 1$) on missing positions $+ 0.1\,\mathcal{L}_\Delta$
Optimizer and regularization	Adam, lr 10^{-3}, weight decay 0, dropout 0.1, no scheduler
Batch size / epochs	Batch size 8; up to 100 epochs
Formal main-benchmark seeds	2021–2025

C Supplementary Mechanism Evidence

The mechanism analyses below use the same data protocol, normalization, objective, and matched evaluation contract as ARCI-T unless an inference-only or historical comparator is stated explicitly.

C.1 Motivation for Controlled Realization

The motivational diagnostic is presented in the main text as Figure 1. Under ER60, enabling the tested trend/fluctuation enhancement increases MAE by 0.371% on PeMS-BAY and 0.348% on PEMS04. This diagnostic is used as motivation for controlled realization rather than as a claim that richer candidate construction is uniformly detrimental.

Table 14: Experimental variants used in the supplementary mechanism analyses. For each dataset–protocol–seed run, θ_{full} denotes the frozen parameters of the matched full ARCI-T model; $R_{\theta_{\text{full}}}$ and $G_{\theta_{\text{full}}}$ denote its residual proposal and acceptance output before intervention.

Variant	Construction / difference from ARCI-T
ARCI-T	Matched Full model with $X_{\text{raw}} = A + G_{\theta_{\text{full}}} \odot R_{\theta_{\text{full}}}$.
Anchor	Deterministic endpoint control with $G = 0$; returns A without loading model weights.
All-R	Full-checkpoint inference intervention with $G = 1$; realizes $A + R_{\theta_{\text{full}}}$.
Const-G	Full-checkpoint inference intervention using fixed $\bar{g}$ equal to the validation acceptance mean.
Shuffle-G	Full-checkpoint inference intervention that permutes gate positions within each sample, preserving values but breaking correspondence.
Generic-G	Independently trained matched alternative whose gate is conditioned only on H_{spatial}.
Self-G	Independently trained matched alternative whose gate is conditioned only on proposal R.
Global-G	Independently trained matched alternative with one learned scalar acceptance coefficient shared across positions.
Shared-E	Independently trained matched alternative whose residual proposal additionally receives the full acceptance evidence $E = [H_{\text{temporal}} \| H_{\text{spatial}} \| q \| \Delta H]$.
w/o q	Removes q only from the final acceptance input; its other verified uses are retained.
w/ Fixed $w = 3$	Replaces the adaptive multiscale trend with the fixed $w = 3$ construction.
w/o Output Anchor	Retains A as encoder input but outputs $G \odot R$ with a direct-value target.
w/o CorrTopK	Sets $H_{\text{spatial}} = H_{\text{temporal}}$; CorrTopK is bypassed and its refiner is inactive.
Predecessor	Separately trained historical legacy adapter with a single $w = 25$ decomposition and predecessor architecture/configuration.
Gate$+R$	Original acceptance evidence plus stop-gradient R as an additional gate input; distinct from Self-G.
Direct	Matched absolute-value prediction.
ARP	Matched anchor-relative residual prediction.

Anchor evaluates the deterministic endpoint, while All-R, Const-G, and Shuffle-G intervene on the corresponding frozen Full checkpoint. Generic-G, Self-G, Global-G, and Shared-E are independently trained under the matched mechanism configuration. Direct and ARP are separately trained from scratch with the same prior-aware backbone and differ in whether the final head predicts an absolute value or an anchor-relative residual.

C.2 Residual Structure Diagnostics

Residual structure analysis shows that the ideal correction still contains periodic and spatial structure, providing empirical support for modeling an anchor-relative residual: the correction is not noise around the anchor but carries structured signal (Table 15).

Definition and aggregation. For each dataset, protocol, and training seed, the analysis uses normalized training windows and the deterministic training mask. Let X denote the normalized ground truth and A the LinearFill reference; at artificial-missing positions it forms the ideal residual $R^* = X - A$ and $\rho = \sum (R^*)^2 / \sum X^2$, with the descriptive anchor-energy proxy $1 - \rho$ (not R^2). Daily and weekly values are residual correlations on the chronologically concatenated training-window stream at lags 288 and 2016 (five-minute sampling), using only pairs that are both hidden; concatenation does not add a separate window-boundary exclusion. Neighbor correlations use the frozen CorrTopK non-self edges at the same time positions, while Random uses an equal number of non-self pairs sampled with `default_rng(20260808)`. Pearson correlations require at least three valid pairs, and the four displayed rows average the five training seeds.

Table 15: Training-period residual structure analysis and anchor energy proxy.

Setting	ρ	Anchor energy proxy	Daily	Weekly	Neighbor	Random
PeMS-BAY Mixed25	0.379	0.621	0.170	0.203	0.273	0.093
PeMS-BAY Sensor25	0.457	0.543	0.207	0.239	0.331	0.116
PEMS04 Mixed25	0.148	0.852	0.151	0.206	0.213	0.113
PEMS04 Sensor25	0.171	0.829	0.202	0.250	0.271	0.147

C.3 Complete Direct vs. ARP Comparison

The main-text Table 5 retains the core matched cells; relative magnitudes are reported below. Here, Δ = ARP − Direct; Relative % is computed relative to Direct, with negative values favoring ARP.

Table 16: Complete matched comparison of Direct and ARP.

Setting	Direct MAE	ARP MAE	Δ	Relative %
PeMS-BAY Mixed25	1.5961±0.0193	**1.5511±0.0089**	-0.0450	-2.82%
PeMS-BAY Sensor25	1.7754±0.0129	**1.7433±0.0201**	-0.0321	-1.81%
PEMS04 Mixed25	21.6503±0.5730	**20.7717±0.2206**	-0.8785	-4.06%
PEMS04 Sensor25	22.9428±0.5992	**21.9530±0.1923**	-0.9898	-4.31%

C.4 Matched Learned-Gate Alternatives

Using the variants defined above, Table 17 reports two matched learned-gate controls against ARCI-T.

ARCI-T achieves the lowest average MAE in all eight matched settings; in per-seed comparisons, ARCI-T wins 36/40 against Generic-G and 40/40 against Self-G. Local fluctuations exist between the two alternative controls, so the results do not constitute a strict per-seed hierarchy among Self-G, Generic-G, and ARCI-T.

C.5 Additional Controlled-Realization Comparisons

Using the variants defined above, Table 18 summarizes the supplementary controlled-realization comparisons. Paired and aggregate results favor ARCI-T over Global-G. Shared-E yields small aggregate differences with mixed protocol-level effects. For the identity-matched Global-G records, let C_s be the equal-weight mean of its four protocol MAEs for training seed s, let F_s be the corresponding Full mean, and let $\Delta_s = C_s - F_s$. Across the five paired seed macros, Δ is 0.01609 ± 0.00348 on

Table 17: MAE of matched learned-gate alternatives (mean±sample std, $n = 5$; lower is better).

Dataset	Protocol	ARCI-T	Generic-G	Self-G
PeMS-BAY	ER25	**0.7033±0.0018**	0.7092±0.0044	0.7131±0.0026
PeMS-BAY	ER60	**0.8887±0.0031**	0.8957±0.0022	0.8978±0.0021
PeMS-BAY	Mixed25	**1.5324±0.0137**	1.5465±0.0101	1.5526±0.0110
PeMS-BAY	Sensor25	**1.6930±0.0201**	1.7309±0.0126	1.7367±0.0187
PEMS04	ER25	**15.9793±0.0253**	15.9966±0.0390	16.0016±0.0360
PEMS04	ER60	**16.9862±0.0269**	17.0447±0.0167	17.0372±0.0095
PEMS04	Mixed25	**21.7311±0.1445**	22.2347±0.2593	22.2150±0.1416
PEMS04	Sensor25	**23.2781±0.2959**	23.7451±0.1898	23.7315±0.2945

PeMS-BAY and 0.19453 ± 0.05388 on PEMS04 (mean $\pm$ sample SD); positive values favor ARCI-T because lower MAE is better. Statistics are computed separately for each dataset from five paired training-seed macro averages in the matched mechanism configuration.

Table 18: Supplementary controlled-realization comparisons: paired wins and aggregate observations.

Variant	Interface	Paired wins vs. ARCI-T	Aggregate observation
Global-G	Learned global acceptance coefficient shared across positions	ARCI-T 40/40	Paired and aggregate summaries favor ARCI-T.
Shared-E	The proposal receives the full acceptance evidence (temporal, spatial, q, ΔH)	ARCI-T 27/40; Shared-E 13/40	Aggregate differences are small with mixed protocol-level effects; no consistent gain is observed.

C.6 Complete Controller Interventions

Table 19 gives the complete MAE of Anchor, All-R, Const-G, Shuffle-G, and ARCI-T under the four protocols, and Figure 5 shows the protocol-level effects. The comparison tests whether fixed acceptance strength or disconnected position correspondence is sufficient for controlled realization.

Table 19: Complete matched controlled-realization intervention MAE.

Dataset	Setting	ER25	ER60	Mixed25	Sensor25
PeMS-BAY	Anchor	0.7799±0.0010	0.9773±0.0021	2.3524±0.0308	2.7029±0.0164
PeMS-BAY	All-R	1.7575±0.0685	2.1506±0.0933	2.8488±0.1881	2.7263±0.2004
PeMS-BAY	Const-G	0.7176±0.0014	0.9153±0.0032	1.9676±0.0415	2.1121±0.0269
PeMS-BAY	Shuffle-G	0.7254±0.0009	0.9270±0.0030	2.0423±0.0415	2.1623±0.0330
PeMS-BAY	ARCI-T	**0.7033±0.0018**	**0.8887±0.0031**	**1.5324±0.0137**	**1.6930±0.0201**
PEMS04	Anchor	17.8356±0.0316	18.7669±0.0216	28.9025±0.3603	32.5315±0.3504
PEMS04	All-R	22.6161±1.4072	24.3734±1.6062	44.5042±3.8702	43.6069±5.2159
PEMS04	Const-G	16.0529±0.0455	17.1477±0.0894	24.1079±0.2622	25.3027±0.6216
PEMS04	Shuffle-G	16.1333±0.0578	17.2835±0.1214	25.4802±0.3818	26.3706±0.5113
PEMS04	ARCI-T	**15.9793±0.0253**	**16.9862±0.0269**	**21.7311±0.1445**	**23.2781±0.2959**

C.7 Position Description and Component Ablations

The position description q provides mask-derived support evidence. The contribution of explicitly including q in the final acceptance evidence is evaluated in Table 20; its other verified uses in representation construction are retained. Using the variants defined above, the complete protocol-level

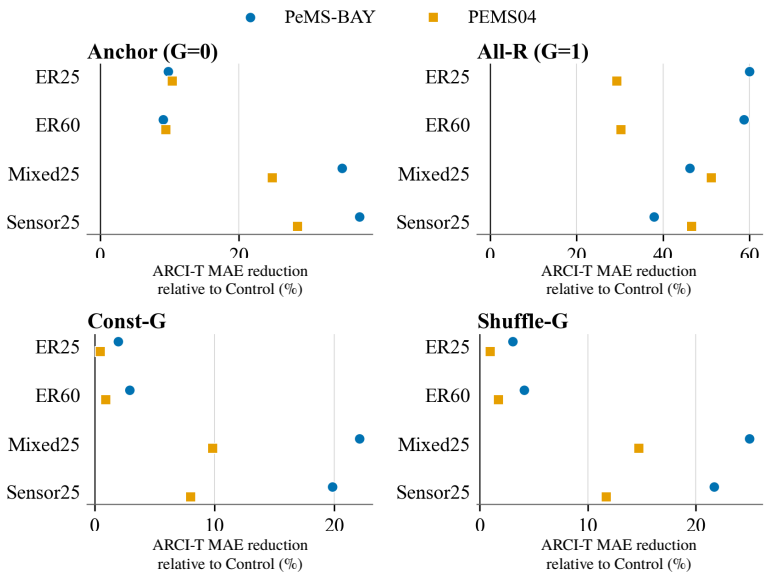


Figure 5: Protocol-level effects of the four matched controlled-realization interventions. The horizontal axis of each facet is independent, and protocols are categorical rather than a continuous trend. The x-axis reports ARCI-T MAE reduction relative to Control (%); positive values favor ARCI-T.

MAE of supporting components, paired macro-average effects, and the Gate+R supplementary analysis are given in Tables 21, 22, 23, and 24, with the Direct–ARP and Gate+R effects in Figures 6 and 7.

Table 20: Ablation of the position description q: ARCI-T vs. w/o q.

Dataset	Protocol	ARCI-T	w/o q
PeMS-BAY	ER25	**0.7033±0.0018**	0.7053±0.0027
PeMS-BAY	ER60	**0.8887±0.0031**	0.8920±0.0040
PeMS-BAY	Mixed25	**1.5324±0.0137**	1.5488±0.0129
PeMS-BAY	Sensor25	**1.6930±0.0201**	1.7370±0.0184
PEMS04	ER25	**15.9793±0.0253**	15.9947±0.0325
PEMS04	ER60	**16.9862±0.0269**	17.0287±0.0228
PEMS04	Mixed25	**21.7311±0.1445**	22.3206±0.2369
PEMS04	Sensor25	**23.2781±0.2959**	23.7801±0.1737

For Table 23, each paired observation first averages the four protocols within one training seed. The reported direction is Δ = Control − Full, so positive values favor ARCI-T; the relative gain is $100\,\overline{\Delta}/\overline{\text{Control}}$. Each row uses five paired seed-level observations. The displayed 95% CI is the two-sided paired t interval $\overline{\Delta} \pm t_{0.975,4} s_\Delta/\sqrt{5}$, computed from the unrounded member metrics; the four protocols are not treated as 20 independent training repetitions. Here, s_Δ is the sample standard deviation of the five paired training-seed macro-average differences.

C.8 Multi-Anchor Supplementary Results

Table 26 reports the complete Anchor and ARCI-T MAE cells under four dataset–protocol settings, and Table 25 reports the relative effects of the alternative anchors. The four LinearFill Anchor means and sample standard deviations are aggregated over five training seeds from the corresponding matched recomputation summaries. Values are taken only from this analysis and are not mixed with the summaries of other analyses.

Table 21: Complete protocol-level MAE of supporting components.

Dataset	Variant	ER25	ER60	Mixed25	Sensor25
PeMS-BAY	ARCI-T	**0.7033±0.0018**	**0.8887±0.0031**	**1.5324±0.0137**	**1.6930±0.0201**
PeMS-BAY	w/o q	0.7053±0.0027	0.8920±0.0040	1.5488±0.0129	1.7370±0.0184
PeMS-BAY	w/ Fixed $w = 3$	0.7102±0.0014	0.8982±0.0038	1.5363±0.0125	1.7126±0.0134
PeMS-BAY	w/o Output Anchor	0.7395±0.0060	0.9280±0.0194	1.5822±0.0259	1.7551±0.0182
PeMS-BAY	w/o CorrTopK	0.7571±0.0022	0.9519±0.0035	2.2214±0.0363	2.5435±0.0326
PEMS04	ARCI-T	**15.9793±0.0253**	**16.9862±0.0269**	**21.7311±0.1445**	**23.2781±0.2959**
PEMS04	w/o q	15.9947±0.0325	17.0287±0.0228	22.3206±0.2369	23.7801±0.1737
PEMS04	w/ Fixed $w = 3$	16.4876±0.0352	17.5740±0.0472	21.8833±0.0594	**23.2068±0.2152**
PEMS04	w/o Output Anchor	16.5495±0.0580	17.6073±0.0794	22.3007±0.1107	23.6872±0.2548
PEMS04	w/o CorrTopK	16.5626±0.0331	17.7120±0.0211	29.0663±0.5588	32.2963±0.6236

Table 22: Matched four-protocol macro-average MAE of supporting components.

Variant	PeMS-BAY	PEMS04
Predecessor	1.6757±0.0092	23.6065±0.1261
ARCI-T	**1.2044±0.0053**	**19.4937±0.0684**
w/o q	1.2208±0.0060	19.7810±0.0834
w/ Fixed $w = 3$	1.2143±0.0043	19.7879±0.0411
w/o Output Anchor	1.2512±0.0074	20.0362±0.0909
w/o CorrTopK	1.6185±0.0088	23.9093±0.2652

Table 23: Paired macro-average effects of w/o Output Anchor and w/o CorrTopK.

Dataset	Control	ΔMAE	95% CI	Relative gain	ARCI-T wins
PeMS-BAY	w/o Output Anchor	0.04681	[0.03733,0.05630]	3.74%	5/5
PEMS04	w/o Output Anchor	0.54249	[0.41412,0.67086]	2.71%	5/5
PeMS-BAY	w/o CorrTopK	0.41412	[0.40625,0.42198]	25.59%	5/5
PEMS04	w/o CorrTopK	4.41559	[4.13336,4.69781]	18.47%	5/5

Table 24: Gate+R (stop-gradient) effects relative to ARCI-T; negative values favor Gate+R.

Setting	ΔMAE	ΔRMSE	Gate+R MAE wins
PeMS-BAY Mixed25	-0.10%	-0.14%	4/5
PeMS-BAY Sensor25	+0.09%	+0.18%	2/5
PEMS04 Mixed25	+0.68%	+0.57%	1/5
PEMS04 Sensor25	-0.27%	-0.31%	3/5

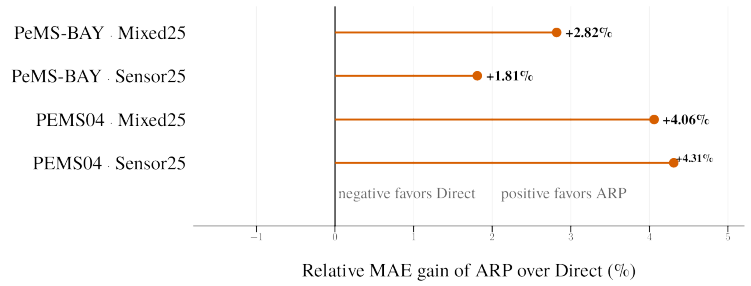


Figure 6: Zero-centered relative MAE effects of Direct and ARP; positive values indicate lower ARP MAE.

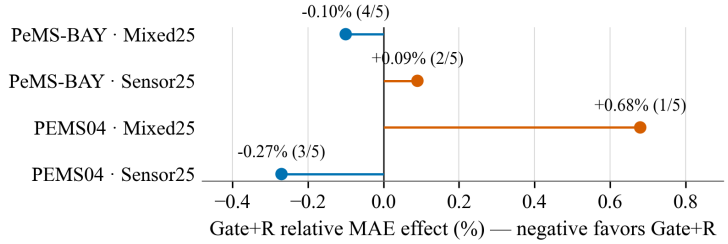


Figure 7: MAE effects of the Gate+R intervention relative to ARCI-T; negative values favor Gate+R. The effects are small and mixed in direction across the four settings. Corresponding RMSE effects are reported in Table 24.

Table 25: Supplementary MAE effects of alternative anchors (Anchor). Relative improvement: (Anchor − ARCI-T)/Anchor; Δ LinearFill: current ARCI-T minus the LinearFill ARCI-T of the same dataset and protocol.

Dataset	Protocol	Anchor	Relative improvement	Δ LinearFill
PeMS-BAY	Mixed25	NearestFill	39.69%	+0.0060
PeMS-BAY	Mixed25	SeasonalProfile	53.91%	+0.1027
PeMS-BAY	Sensor25	NearestFill	41.68%	+0.0086
PeMS-BAY	Sensor25	SeasonalProfile	49.04%	+0.1020
PEMS04	Mixed25	NearestFill	42.63%	+0.2816
PEMS04	Mixed25	SeasonalProfile	44.47%	+0.8309
PEMS04	Sensor25	NearestFill	46.45%	+0.1547
PEMS04	Sensor25	SeasonalProfile	41.56%	+0.7619

Table 26: ARCI-T MAE under different deterministic anchors.

Dataset	Protocol	Recovery reference	Anchor	ARCI-T
PeMS-BAY	Mixed25	LinearFill	**2.3524±0.0308**	**1.4817±0.0148**
		NearestFill	2.4666±0.0306	1.4877±0.0186
		SeasonalProfile	3.4378±0.0126	1.5844±0.0147
	Sensor25	LinearFill	**2.7029±0.0164**	**1.6458±0.0185**
		NearestFill	2.8366±0.0174	1.6544±0.0219
		SeasonalProfile	3.4297±0.0103	1.7478±0.0165
PEMS04	Mixed25	LinearFill	**28.9025±0.3603**	**20.6271±0.1860**
		NearestFill	36.4442±0.4121	20.9087±0.1771
		SeasonalProfile	38.6411±0.4348	21.4580±0.2647
	Sensor25	LinearFill	**32.5315±0.3504**	**21.9462±0.1316**
		NearestFill	41.2707±0.3240	22.1009±0.1033
		SeasonalProfile	38.8554±0.2895	22.7081±0.4565

NearestFill performs deterministic filling with the nearest visible support along the time axis and contains no learnable parameters. SeasonalProfile is built only from training statistics and forms a deterministic cache from node×period-of-day medians; these statistics are not updated during validation or testing. The formal multi-anchor results use MAE as the evaluation metric.

C.9 Supplementary Protocol-Level Geometry Analysis

This subsection relates the learned-gate advantage to coarse missingness geometry encoded by the evaluated protocols. ER25 and ER60 form the entry-random group, whereas Mixed25 and Sensor25 form the structured block/sensor group. For each dataset and protocol setting, the five training-seed MAEs are first averaged to form that setting's protocol mean. Let $\Delta_{d,p} = \mathrm{MAE}_{\mathrm{Generic-G},d,p} - \mathrm{MAE}_{\mathrm{ARCI-T},d,p}$ and define the corresponding relative advantage as $100\Delta_{d,p}/\mathrm{MAE}_{\mathrm{ARCI-T},d,p}$, using ARCI-T MAE as the denominator. Each group value in the table is then the equal-weight mean over its two valid protocol settings; positive values favor ARCI-T.

Table 27: Quantitative protocol-level geometry summary from matched learned-gate comparisons. $\Delta_{\mathrm{MAE}} = \mathrm{MAE}_{\mathrm{Generic-G}} - \mathrm{MAE}_{\mathrm{ARCI-T}}$; mean relative advantage is $100\Delta_{\mathrm{MAE}}/\mathrm{MAE}_{\mathrm{ARCI-T}}$, computed after five-seed protocol means and then averaged equally over the two protocols in each group. Positive values indicate lower ARCI-T MAE.

Dataset	Coarse geometry group	Valid settings	Mean Δ_{MAE}	Mean relative advantage	ARCI-T lower
PeMS-BAY	Entry-random (ER25/ER60)	2	0.0064	0.808%	2/2
PeMS-BAY	Structured block/sensor (Mixed25/Sensor25)	2	0.0259	1.576%	2/2
PEMS04	Entry-random (ER25/ER60)	2	0.0379	0.226%	2/2
PEMS04	Structured block/sensor (Mixed25/Sensor25)	2	0.4853	2.162%	2/2

The structured group has the larger mean advantage on both datasets: the increase is from 0.0064 to 0.0259 MAE on PeMS-BAY and from 0.0379 to 0.4853 MAE on PEMS04. The corresponding relative advantages increase from 0.808% to 1.576% and from 0.226% to 2.162%, respectively. This supports a protocol-level association between structured missingness and a larger ARCI-T advantage.

D Full Benchmark and Secondary Analyses

This appendix supplements the main-result summary, the MSE of ARCI-T, and the complete numeric values of each matched comparison. Matched experiments use five training seeds and report uncertainty as the sample standard deviation.

D.1 Complete Numerical Results

Table 28 summarizes the main results with the best external baseline per dataset–protocol setting; Table 29 reports the per-seed aggregated MSE of ARCI-T; and Tables 30– 33 contain the complete benchmark tables—Mean, KNN, Linear, CSDI, Vanilla Transformer (Vaswani et al., 2017), BRITS, SAITS, STAMImputer (Wang et al., 2025), ImputeFormer (Nie et al., 2024), GiFlow, OPCR, Transformer + MG, ATTST, ARCI-T, and the retained representative methods—used to derive the compact main-text tables.

D.2 Hard-Missingness Robustness

Entry40, Mixed40, and Sensor40 create different correction demands. Mixed40 combines random missing entries with substantially increased contiguous temporal blocking relative to Mixed25. Among the realization variants compared in these pressure tests, ARCI-T has the lowest reported mean MAE. Under Entry40, the deterministic Anchor remains competitive, whereas under Sensor40

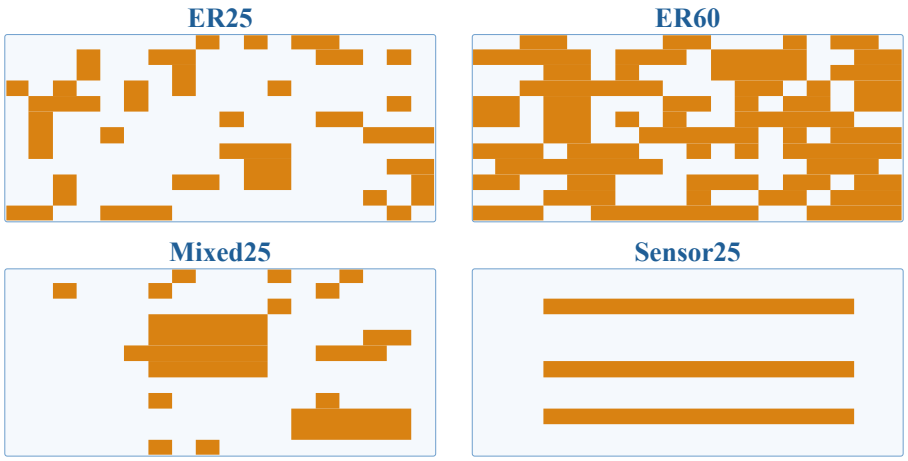


Figure 8: The four missingness protocols used in the evaluation: ER25, ER60, Mixed25, and Sensor25. Orange marks artificially hidden positions and light blue marks observed positions.

Table 28: Main-result summary; negative relative difference indicates lower ARCI-T MAE.

Dataset	Protocol	Best external	External MAE	ARCI-T MAE	Relative Δ	Winner
METR-LA	ER25	OPCR	2.3287	2.1181	-9.04%	ARCI-T
METR-LA	ER60	OPCR	2.7142	2.4351	-10.28%	ARCI-T
METR-LA	Mixed25	OPCR	4.2784	3.2387	-24.30%	ARCI-T
METR-LA	Sensor25	BRITS	4.4520	3.4722	-22.01%	ARCI-T
PeMS-BAY	ER25	OPCR	0.7781	0.6867	-11.75%	ARCI-T
PeMS-BAY	ER60	Linear	0.9771	0.8648	-11.49%	ARCI-T
PeMS-BAY	Mixed25	OPCR	1.8081	1.4817	-18.05%	ARCI-T
PeMS-BAY	Sensor25	OPCR	1.9898	1.6458	-17.29%	ARCI-T
PEMS04	ER25	ATTST	16.6890	15.8302	-5.15%	ARCI-T
PEMS04	ER60	Linear	18.7619	16.7568	-10.69%	ARCI-T
PEMS04	Mixed25	BRITS	21.8489	20.6271	-5.59%	ARCI-T
PEMS04	Sensor25	BRITS	21.8862	21.9462	+0.27%	BRITS
PEMS08	ER25	ATTST	13.0032	11.4385	-12.03%	ARCI-T
PEMS08	ER60	Linear	14.3977	12.4842	-13.29%	ARCI-T
PEMS08	Mixed25	BRITS	18.5539	16.2881	-12.21%	ARCI-T
PEMS08	Sensor25	BRITS	18.8274	17.2828	-8.20%	ARCI-T

Table 29: Per-seed and aggregated MSE of ARCI-T.

Protocol	METR-LA	PeMS-BAY	PEMS04	PEMS08
ER25	18.9723±7.2705	1.8164±0.0248	678.0570±5.5222	354.7077±2.0420
ER60	27.1056±9.5940	3.0889±0.0494	751.2475±5.2334	404.1057±3.5266
Mixed25	49.4899±0.6022	10.6852±0.2460	1069.3093±24.9837	677.7251±30.5171
Sensor25	54.8948±2.7331	12.6665±0.5342	1199.2053±24.5951	750.6318±11.3049
Macro avg	37.6157±2.6834	7.0642±0.1318	924.4548±1.9510	546.7926±9.4764

Table 30: Complete protocol-level MAE on METR-LA.

Method	ER25	ER60	Mixed25	Sensor25
Mean	15.2474±0.0399	15.2298±0.0063	15.3634±0.0731	15.4454±0.1085
KNN	8.9533±0.0099	8.9516±0.0181	9.0603±0.0588	9.1630±0.0684
Linear	2.6392±0.0114	3.0546±0.0035	6.2913±0.1002	7.0923±0.0930
CSDI	3.4592±0.2497	3.7247±0.3654	4.9260±0.2862	5.1474±0.2272
Transformer	5.2688±0.0397	5.4213±0.0186	5.3515±0.0204	5.5008±0.0262
BRITS	4.2009±0.0161	4.6183±0.0062	4.3530±0.0351	4.4520±0.0112
SAITS	4.4123±0.0652	4.5716±0.0186	4.5260±0.0668	4.4886±0.0140
STAMImputer	3.2895±0.1061	4.1034±0.1570	5.1125±0.1482	6.3277±1.1234
ImputeFormer	3.4515±0.0901	3.8115±0.0316	5.2002±0.1282	5.4317±0.1778
GiFlow	3.6209±0.1168	4.1328±0.1328	6.5301±0.1808	7.2168±0.1929
OPCR	2.3287±0.0091	2.7142±0.0546	4.2784±0.1214	4.6025±0.0726
Transformer + MG	4.2944±0.0505	4.7638±0.0440	6.3536±0.0516	6.8506±0.0726
ATTST	3.5856±0.0972	4.4109±0.1554	10.6181±0.3847	12.0431±0.4613
ARCI-T	**2.1181±0.2778**	**2.4351±0.3296**	**3.2387±0.0145**	**3.4722±0.0652**

Table 31: Complete protocol-level MAE on PeMS-BAY.

Method	ER25	ER60	Mixed25	Sensor25
Mean	5.4090±0.0097	5.4129±0.0025	5.4445±0.0226	5.4697±0.0189
KNN	2.6969±0.0074	2.6979±0.0062	2.7027±0.0114	2.7213±0.0178
Linear	0.7793±0.0010	0.9771±0.0004	2.3318±0.0055	2.7106±0.0153
CSDI	1.0453±0.0299	1.1468±0.0243	2.2649±0.0699	2.5579±0.0638
Transformer	2.4043±0.0019	2.4520±0.0134	2.4756±0.0150	2.4983±0.0374
BRITS	1.9574±0.0022	2.0993±0.0072	2.0498±0.0091	2.0791±0.0135
SAITS	2.1544±0.0035	2.2152±0.0033	2.1910±0.0214	2.2126±0.0096
STAMImputer	1.3541±0.0948	1.6074±0.0048	2.2323±0.0431	2.5713±0.2386
ImputeFormer	1.5937±0.0109	1.7508±0.0157	2.4917±0.0267	2.5287±0.0257
GiFlow	1.1481±0.0378	1.4861±0.0923	2.7913±0.0688	3.1286±0.0721
OPCR	0.7781±0.0094	0.9782±0.0047	1.8081±0.0133	1.9898±0.0299
Transformer + MG	1.8386±0.0107	1.9922±0.0076	2.7720±0.0124	2.9048±0.0127
ATTST	0.9964±0.0291	1.2272±0.0456	3.9115±0.3750	4.4718±0.4480
ARCI-T	**0.6867±0.0011**	**0.8648±0.0020**	**1.4817±0.0148**	**1.6458±0.0185**

Table 32: Complete protocol-level MAE on PEMS04.

Method	ER25	ER60	Mixed25	Sensor25
Mean	105.1054±0.1384	105.0301±0.0285	104.9515±0.9634	104.8767±1.3138
KNN	31.6316±0.0780	31.6078±0.0218	31.6196±0.3228	31.7336±0.0830
Linear	17.8422±0.0341	18.7619±0.0278	28.7938±0.4622	32.3551±0.3379
CSDI	18.4684±0.4699	19.7618±0.9237	26.0755±1.0736	27.8359±1.0915
Transformer	26.4993±0.0107	27.6620±0.0793	27.6099±0.3237	28.0019±0.2182
BRITS	21.5783±0.0774	22.6212±0.0711	21.8489±0.3313	**21.8862±0.0877**
SAITS	22.4598±0.1624	23.1621±0.0189	22.9538±0.3743	23.0592±0.1664
STAMImputer	24.4377±0.4087	22.9790±0.1005	33.9869±0.8822	35.9572±0.4661
ImputeFormer	21.0585±0.1569	23.2714±0.7371	32.7313±0.6403	34.5750±1.0135
GiFlow	23.5271±2.0956	25.4442±1.2988	33.3930±1.2381	35.5746±1.2919
OPCR	17.7324±0.1835	20.0364±0.1167	24.7138±0.1360	25.6527±0.2561
Transformer + MG	23.1509±0.0648	25.1713±0.1195	36.9225±0.3245	39.0646±0.4944
ATTST	16.6890±0.0968	18.9221±0.3474	72.0310±2.8391	83.0503±3.4107
ARCI-T	**15.8302±0.0258**	**16.7568±0.0402**	**20.6271±0.1860**	21.9462±0.1316

Table 33: Complete protocol-level MAE on PEMS08.

Method	ER25	ER60	Mixed25	Sensor25
Mean	89.9686±0.1379	90.1336±0.0507	90.6923±0.5180	90.6138±0.8574
KNN	27.4497±0.0501	27.4413±0.0175	27.4038±0.3501	27.2159±0.2726
Linear	13.4581±0.0304	14.3977±0.0454	23.7914±0.3179	26.5041±0.3636
CSDI	13.6322±0.2192	14.7765±0.7140	18.6175±0.5084	20.1170±0.4728
Transformer	24.3575±0.0083	26.1088±0.1873	25.3964±0.3592	25.6868±0.5571
BRITS	17.8121±0.0365	19.7596±0.0445	18.5539±0.4163	18.8274±0.0999
SAITS	19.6267±0.1358	20.7299±0.0728	20.1992±0.6677	20.2777±0.2207
STAMImputer	18.7189±0.0710	18.3922±0.1935	25.6622±0.5576	27.7046±0.3078
ImputeFormer	15.9497±0.2822	18.0933±0.3547	23.6201±0.3695	25.6015±1.1270
GiFlow	20.9106±1.2183	22.7674±1.8129	35.0715±2.9656	38.3634±3.6857
OPCR	13.8884±0.1652	15.3749±0.3076	20.2636±0.2873	21.1647±0.5851
Transformer + MG	18.6460±0.1815	20.5992±0.1146	31.7632±0.6950	33.3316±0.3092
ATTST	13.0032±0.0789	14.8604±0.2624	61.5602±2.3360	70.8680±2.9895
ARCI-T	**11.4385±0.0339**	**12.4842±0.0523**	**16.2881±0.1773**	**17.2828±0.0660**

the Anchor error rises substantially and learned correction becomes more important. Table 34 reports the complete pressure-test results. Each Entry40/Sensor40 cell pools ten frozen checkpoints per dataset (five seeds from each of the Mixed25- and Sensor25-trained mechanism configurations), while each Mixed40 cell pools the five frozen Mixed25 checkpoints. All three regimes use no retraining or online adaptation.

Table 34: Missingness pressure tests under Entry40, Mixed40, and Sensor40.

Dataset	Regime	Anchor	All-R	Shuffle-G	ARCI-T
PeMS-BAY	Entry40	0.8420±0.0017	3.0608±0.3955	0.8666±0.0172	**0.8248±0.0031**
PeMS-BAY	Mixed40	2.6123±0.0239	2.8386±0.1924	2.2081±0.0337	**1.6451±0.0033**
PeMS-BAY	Sensor40	5.3210±0.1414	3.2825±0.4430	3.2764±0.4457	**3.2628±0.4533**
PEMS04	Entry40	18.1377±0.0347	45.9579±6.4319	18.0656±0.4026	**17.4195±0.3951**
PEMS04	Mixed40	32.6072±0.3014	44.2974±4.5615	27.4743±0.4187	**23.1204±0.2029**
PEMS04	Sensor40	105.2898±3.3993	44.9412±5.8759	40.2989±3.7980	**40.0085±3.6151**

D.3 Temporal-Distance Analysis

All temporal-distance evaluations use frozen checkpoints without retraining or online adaptation; scalers and CorrTopK are not refitted per bin. Test sequences are generated as non-overlapping consecutive windows of length 200 with step 200 and are grouped into T1–T4 by increasing distance from the training cutoff. The PeMS-BAY test interval is $[41692, 52116)$, giving 52 complete windows (13 per bin); the PEMS04 test interval is $[13593, 16992)$, giving 16 complete windows (4 per bin). PEMS04 is reported by sample index and sequential distance.

This subsection keeps the sample-standard-deviation convention (ddof=1). Absolute change is defined as T4−T1, relative degradation as $(\text{T4} - \text{T1})/\text{T1}$, and the late/early ratio as T4/T1; the normalized slope and Spearman ρ describe the direction of change across the four bins. For ARCI-T, the temporal-distance summary is based on four missingness protocols (ER25, ER60, Mixed25, and Sensor25) and five training seeds (2021–2025). Within each protocol–seed run, the corresponding frozen checkpoint is evaluated across T1–T4 without retraining or online adaptation. MAE is read from the original-scale evaluator for each protocol–seed run. For each seed and temporal bin, the four protocol MAEs are first macro-averaged; for each seed, an ordinary-least-squares slope is then fit to those four macro MAEs at normalized bin positions $x = (0, 1/3, 2/3, 1)$, with the slope reported as Norm. slope. The reported five-seed Norm. slope mean and sample standard deviation are finally computed from those five seed-level slopes.

Complete bin-level and summary statistics are given in Tables 36–39. On both datasets, the absolute T4−T1 change of ARCI-T is lower than that of Linear and Predecessor; on PEMS04 its relative

Table 35: MAE under temporal-distance bins.

Dataset	Method	T1	T2	T3	T4
PeMS-BAY	Linear	1.5487	1.6682	1.8561	1.7402
	ARCI-T	**1.1171**	**1.1532**	**1.2886**	**1.2590**
PEMS04	Linear	22.2608	27.4455	22.1510	26.2156
	ARCI-T	**18.7413**	**20.9433**	**18.1723**	**20.1287**

Table 36: Complete T1–T4 MAE on PeMS-BAY.

Method	T1	T2	T3	T4
Linear	1.5487±0.0055	1.6682±0.0198	1.8561±0.0250	1.7402±0.0092
Predecessor	1.5164±0.0086	1.6351±0.0222	1.8319±0.0259	1.7202±0.0103
ARCI-T	**1.1171±0.0078**	**1.1532±0.0118**	**1.2886±0.0058**	**1.2590±0.0052**
w/o q	1.1309±0.0076	1.1680±0.0116	1.3072±0.0067	1.2775±0.0049
w/ Fixed $w = 3$	1.1264±0.0069	1.1634±0.0090	1.2988±0.0049	1.2692±0.0051
w/o Output Anchor	1.1577±0.0073	1.1944±0.0118	1.3441±0.0114	1.3090±0.0083
w/o CorrTopK	1.4733±0.0150	1.5708±0.0188	1.7655±0.0164	1.6651±0.0060

Table 37: PeMS-BAY temporal-distance summary statistics.

Method	Absolute Δ	Relative Δ	Late/Early	Norm. slope	Spearman ρ
Linear	0.1915±0.0132	12.37%±0.89	1.1237±0.0089	0.2287±0.0201	0.80
Predecessor	0.2038±0.0164	13.44%±1.15	1.1344±0.0115	0.2424±0.0232	0.80
ARCI-T	0.1419±0.0093	12.71%±0.91	1.1271±0.0091	0.1683±0.0114	0.80
w/o q	0.1467±0.0079	12.97%±0.77	1.1297±0.0077	0.1738±0.0103	0.80
w/ Fixed $w = 3$	0.1428±0.0079	12.68%±0.77	1.1268±0.0077	0.1691±0.0099	0.80
w/o Output Anchor	0.1512±0.0093	13.07%±0.85	1.1307±0.0085	0.1810±0.0126	0.80
w/o CorrTopK	0.1919±0.0151	13.03%±1.15	1.1303±0.0115	0.2311±0.0189	0.80

Table 38: Complete T1–T4 MAE on PEMS04.

Method	T1	T2	T3	T4
Linear	22.2608±0.0943	27.4455±0.3647	22.1510±0.1775	26.2156±0.2045
Predecessor	21.0601±0.1218	26.7169±0.4058	21.3618±0.1698	25.3246±0.2356
ARCI-T	**18.7413±0.0640**	**20.9433±0.1976**	**18.1723±0.1002**	**20.1287±0.1417**
w/o q	18.9953±0.0355	21.2185±0.2006	18.4617±0.0744	20.4590±0.1536
w/ Fixed $w = 3$	19.0471±0.1008	21.1952±0.1614	18.5150±0.1214	20.4048±0.1028
w/o Output Anchor	19.3373±0.1217	21.3682±0.1241	18.8146±0.1698	20.6336±0.1675
w/o CorrTopK	21.5380±0.2548	26.8716±0.4015	21.6748±0.3286	25.5901±0.2606

Table 39: PEMS04 temporal-distance summary statistics.

Method	Absolute Δ	Relative Δ	Late/Early	Norm. slope	Spearman ρ
Linear	3.9548±0.2256	17.77%±1.05	1.1777±0.0105	1.9710±0.2547	0.16
Predecessor	4.2644±0.3381	20.26%±1.71	1.2026±0.0171	2.2315±0.3364	0.40
ARCI-T	1.3874±0.1604	7.40%±0.87	1.0740±0.0087	0.4174±0.1430	0.00
w/o q	1.4637±0.1364	7.71%±0.71	1.0771±0.0071	0.4903±0.1296	0.00
w/ Fixed $w = 3$	1.3577±0.1440	7.13%±0.79	1.0713±0.0079	0.4178±0.1264	0.00
w/o Output Anchor	1.2962±0.0608	6.70%±0.29	1.0670±0.0029	0.4005±0.1001	0.00
w/o CorrTopK	4.0522±0.3307	18.83%±1.68	1.1883±0.0168	2.0879±0.3510	0.24

change is 7.40%, clearly lower than their 17.77% and 20.26%. Together, these summaries show that the frozen model retains lower absolute MAE than Linear in every displayed bin on both datasets.

E Conceptual Comparison with FENCE

FENCE (Mao et al., 2026) is the closest published formulation to the controlled-correction concern: it studies controlled traffic imputation and adjusts the diffusion guidance strength during generation from feedback and cluster-level correlations. The conceptual relationship is that both methods control the strength of a learned correction instead of applying it unconditionally. The difference lies in the controlled quantity: FENCE modulates the guidance scale of an iterative generative process, whereas ARCI-T controls the realized contribution of an anchor-relative residual to an explicitly retained deterministic recovery reference, through a bounded position-conditioned acceptance coefficient whose evidence path does not read the proposal itself. This conceptual comparison follows the published FENCE formulation.