PROST: LEARNING TO PROGRESSIVELY REFINE SPATIO-TEMPORAL FORECASTS IN A MULTI-SCALE FORECAST SPACE WITH ADAPTIVE SCALE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatio-temporal forecasting supports real-world applications such as traffic management and air-quality monitoring. Many existing methods model historical temporal dynamics and spatial dependencies to directly predict the future at the target scale. This approach leaves the relationship between coarse future structure and finer detail implicit, without explicit control over how each is revised during forecasting. We introduce ProST (Progressive Spatio-Temporal Forecasting) to progressively refine predictions of the same future in a joint temporal–spatial scale space. ProST controls coarse-scale revision separately from the addition of finer-scale detail, while adaptive scale selection determines the scales at which refinement proceeds. Each selected scale pair defines an explicit forecast state and a scale-aligned target. Refinement expands the preceding forecast and decomposes a learned correction into a weighted coarse-scale update and zero-aggregate detail. A shared scale-conditioned neural forecaster computes these corrections from historical observations and the preceding forecast. A history-conditioned selector chooses a complete scale-selection sequence before forecasting begins. The refinement satisfies an exact cross-scale identity relating the refined forecast's aggregate to the updated coarse prediction. Experiments show positive mean intermediate refinement gains, and frozen-weight interventions on the temporal parent/detail decomposition increase terminal error when either component is removed across all evaluated seeds and datasets. Our code is available in `https://anonymous.4open.science/r/ProST-1B6C/`

1 INTRODUCTION

Spatio-temporal forecasting predicts future observations across connected locations, supporting traffic management and air-quality monitoring. Graph and attention models learn temporal dynamics and spatial dependencies from history to produce predictions at the target scale (Li et al., 2017; Wu et al., 2019; Zheng et al., 2020; Jiang et al., 2023; Liu et al., 2023a). These methods emphasize how historical information is encoded and propagated. A complementary question concerns the prediction itself: how can an estimate of the future be progressively revised before producing the final forecast?

The same future can be represented at different temporal and spatial scales. Predictions over time intervals and groups of nodes describe coarse structure, while finer predictions specify variation within those aggregates. Making these intermediate predictions explicit provides targets for supervision and estimates on which subsequent predictions can condition. Temporal and cross-temporal hierarchies already coordinate forecasts across aggregation levels (Athanasopoulos et al., 2017; Di Fonzo & Girolimetto, 2023a). Scaleformer refines forecasts along temporal scales, and mr-Diff and SCALER use coarse predictions to guide finer ones (Shabani et al., 2023; Shen et al., 2024; Le et al., 2026). STORM constructs multi-level spatial forecasts, while NeST uses regional futures to guide node-level prediction (Huang et al., 2026; Ai et al., 2026). These forecast-level constructions bring the relationship between coarse predictions and finer detail into the forecasting process.

Refinement must accommodate two distinct changes: revising an inaccurate coarse prediction and adding detail within its temporal and spatial units. A single additive correction can perform both, but

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.

does not distinguish their effects on the coarse aggregate. Conversely, requiring a finer prediction to recover the preceding forecast exactly leaves its coarse errors unchanged. The challenge is to permit coarse-scale revision while separately controlling the addition of finer detail. For example, changing the predicted average over a time interval is different from adjusting fluctuations within that interval without changing its average. This distinction motivates a refinement rule that explicitly separates the two components of a forecast correction.

We introduce ProST (Progressive Spatio-Temporal Forecasting) to learn such refinements in a multi-scale forecast space. Refinement expands the preceding forecast to a finer pair and predicts a correction. Aggregating the correction identifies its coarse-scale update; subtracting the expanded update leaves detail with zero aggregate. Weighting the coarse update controls how much the preceding prediction is revised while retaining the finer detail.

A multi-scale forecast space admits temporal, spatial, and joint refinement, with different computational costs. Whereas Pathformer adapts temporal patch sizes for feature computation (Chen et al., 2024), our selection concerns explicit forecasts of the same future. The formulation permits local decisions; ProST selects a complete legal sequence before forecasting using history-conditioned terminal-loss residual predictions and computational cost.

A shared scale-conditioned forecaster executes the sequence, combining historical evidence and the preceding forecast through temporal processing and hierarchical spatial interaction. After fixed-sequence training, it learns over a mixture of fixed and sampled legal sequences, with supervision at intermediate and terminal scales.

Our contributions are:

- We introduce ProST, a spatio-temporal forecasting framework that progressively refines explicit predictions of the same future in a joint temporal–spatial forecast space. ProST combines forecast-to-forecast refinement with adaptive scale selection, jointly specifying how predictions are revised and which intermediate scales are used.
- We develop a refinement mechanism that separates coarse-scale revision from finer-scale detail. A weighted coarse update corrects the preceding prediction, while a zero-aggregate component adds detail without altering that update. Their combination supports temporal, spatial, and joint refinement with an explicit cross-scale aggregation relation.
- We make the forecasting process adaptable through a shared scale-conditioned forecaster trained over fixed and sampled legal scale-selection sequences. Intermediate supervision trains the forecasts that condition subsequent refinement, while a history-conditioned selector chooses a complete sequence under computational constraints before forecasting.
- Experiments demonstrate the forecasting effectiveness of ProST and the complementary roles of coarse-scale revision and finer-scale detail. Progressive refinement improves intermediate forecasts on average, while training over scale selections improves accuracy across alternative forecasting sequences. Adaptive scale selection provides different accuracy–computation trade-offs.

2 Related Work

Spatio-temporal modeling and forecast hierarchies. Graph forecasters encode spatial dependence through diffusion, learned connectivity, or hierarchical exchange (Li et al., 2017; Wu et al., 2019; Guo et al., 2021); STEP adds pre-trained historical representations (Shao et al., 2022b), and DST-Mamba separates trend and seasonal components (He et al., 2025). Forecast hierarchies instead organize predictions across aggregations of the same future: temporal reconciliation coordinates time scales (Athanasopoulos et al., 2017; Rangapuram et al., 2023; Zhou et al., 2024), cross-temporal reconciliation adds cross-sectional constraints, including geographic aggregation in solar forecasting (Di Fonzo & Girolimetto, 2023a;b), and learned spatial hierarchies coordinate groups of series or locations (Cini et al., 2024; Zhang et al., 2026b). These forecast-level relationships motivate our scale-aligned targets.

Multi-scale and progressive future prediction. N-HiTS combines multi-rate sampling with hierarchical interpolation (Challu et al., 2023). Scaleformer directly interpolates and corrects forecasts

along a fixed temporal schedule (Shabani et al., 2023); mr-Diff and SCALER condition finer predictions on coarse structure through staged diffusion and shape-guided residual prediction, respectively (Shen et al., 2024; Le et al., 2026). DYffusion couples temporal interpolation with forecasting in a dynamics-informed diffusion process (Rühling Cachay et al., 2023). Spatially, STORM combines future predictions from multiple levels (Huang et al., 2026), while NeST uses regional futures to guide progressive node-level prediction (Ai et al., 2026). These methods overlap with our explicit multi-scale predictions; our refinement connects temporal–spatial scale pairs and separates corrections into coarse-scale updates and zero-aggregate detail.

Input-dependent scale computation. Pathformer and TimeMosaic adapt temporal patching (Chen et al., 2024; Ding et al., 2026), STM3 mixes multiscale Mamba experts (Chen et al., 2026), and ARROW selects time intervals for successive weather predictions (Tian et al., 2026). Our selection determines where the same future forecast is refined in temporal–spatial scale space. Section 4.3 defines local and complete-sequence decisions; ProST selects a complete sequence from history before forecasting.

3 Problem Formulation

We consider an undirected graph $\mathcal{G} \triangleq (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} \triangleq \{1, \ldots, N\}$ denotes the set of N nodes, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of E undirected edges. Specifically, there exists an edge $(i, j) \in \mathcal{E}$ if nodes $i \in \mathcal{N}$ and $j \in \mathcal{N}$ are connected. We consider a sequence of T data samples, $\{(\mathbf{C}_\tau, \mathbf{x}_\tau)\}_{\tau \in \mathcal{T}}$, collected from N nodes over T time steps, where $\mathcal{T} \triangleq \{1, ..., T\}$ denotes the set of T time steps, $\mathbf{C}_\tau \in \mathbb{R}^{N \times D}$ represents the features of the N nodes with the (i, d)-th element $C_{\tau,i,d} \in \mathbb{R}$ presenting the d-th feature of node $i \in \mathcal{N}$ at time step $\tau \in \mathcal{T}$, and $\mathbf{x}_\tau \in \mathbb{R}^N$ denotes the labels (parameters of interest) of the N nodes at time step $\tau \in \mathcal{T}$ with the i-th element $x_{\tau,i} \in \mathbb{R}$ representing the label of node $i \in \mathcal{N}$ at time step $\tau \in \mathcal{T}$. The data samples generally possess spatial and temporal correlation, which allows for the prediction of node labels of the future from the node features and labels of the past.

We formally define the spatial and temporal prediction task as follows. Specifically, at each time step t, we predict the labels of the N nodes over the future F time steps from the features and labels of the N nodes over the last H time steps using a learnable function f parameterized by Θ:

$$\{\hat{\mathbf{x}}_\tau\}_{\tau=t+1}^{t+F} = f(\{(\mathbf{C}_\tau, \mathbf{x}_\tau)\}_{\tau=t-H+1}^{t}, \mathbf{A}; \Theta) \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix. Given the forecasting loss $\mathcal{L}\left(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}, \{\hat{\mathbf{x}}_\tau\}_{\tau=t+1}^{t+F}\right)$, we would like to optimize the function parameters Θ to minimize the loss. The spatial and temporal prediction problem is formulated as follows:

$$\min_{\Theta} \quad \mathcal{L}\left(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}, \{\hat{\mathbf{x}}_\tau\}_{\tau=t+1}^{t+F}\right) \tag{2}$$

4 Progressive Multi-scale Forecasting

For prediction origin t, Section 3 provides the future sequence $\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}$ and forecast sequence $\{\hat{\mathbf{x}}_\tau\}_{\tau=t+1}^{t+F}$. We represent these future and forecast sequences at multiple temporal and spatial scales. A forecast network progressively forecasts results over multiple spatio-temporal scales, and a scale selection determines the next scale.

4.1 Multi-Scale Forecast States

Let $\mathcal{R} = \{r_1, \ldots, r_K\}$ and $\mathcal{S} = \{s_1, \ldots, s_L\}$ be ordered temporal and spatial scales, with $r_1 < \cdots < r_K = F$, $s_1 < \cdots < s_L = N$, and $r_k \mid F$ for each k. Successive selected temporal scales $r_k \to r_i$ must satisfy $r_k \mid r_i$.

Definition 4.1 (Multi-Scale Spatio-Temporal Representation)**.** For $i > k$ with $r_k \mid r_i$, let $\eta_{k\to i} = r_i/r_k$. For any spatial width d, define $\mathcal{T}_{i\to k} : \mathbb{R}^{r_i \times d} \to \mathbb{R}^{r_k \times d}$ by

$$[\mathcal{T}_{i\to k}(\mathbf{Z})]_{m,u} = \frac{1}{\eta_{k\to i}} \sum_{a=1}^{\eta_{k\to i}} Z_{(m-1)\eta_{k\to i}+a,u}, \quad m = 1, \ldots, r_k, \quad u = 1, \ldots, d. \tag{3}$$

Let $\mathcal{T}_{k\to k}$ be the identity. For $r_k \mid r_h \mid r_i$, temporal aggregation satisfies $\mathcal{T}_{i\to k} = \mathcal{T}_{h\to k} \circ \mathcal{T}_{i\to h}$.

At spatial scale s_l, let $\mathcal{N}^{(s_l)} = \{1, \dots, s_l\}$, with $\mathcal{N}^{(s_L)} = \mathcal{N}$. For $l = 1, \dots, L-1$, a surjective map $\psi_l : \mathcal{N}^{(s_{l+1})} \to \mathcal{N}^{(s_l)}$ clusters finer units into coarser units. For $u \in \mathcal{N}^{(s_l)}$, its children are $\Gamma_u^{(l+1|l)} = \psi_l^{-1}(\{u\}) = \{v \in \mathcal{N}^{(s_{l+1})} : \psi_l(v) = u\}$. For any temporal length q, define $\mathcal{P}_{l+1\to l} : \mathbb{R}^{q\times s_{l+1}} \to \mathbb{R}^{q\times s_l}$ by

$$[\mathcal{P}_{l+1\to l}(\mathbf{Z})]_{m,u} = \frac{1}{|\Gamma_u^{(l+1|l)}|} \sum_{v\in\Gamma_u^{(l+1|l)}} Z_{m,v}, \quad m = 1, \dots, q, \quad u \in \mathcal{N}^{(s_l)}. \tag{4}$$

For $j > l$, let $\mathcal{P}_{j\to l} = \mathcal{P}_{l+1\to l} \circ \cdots \circ \mathcal{P}_{j\to j-1}$, and let $\mathcal{P}_{l\to l}$ be the identity.

The finest-scale target is defined directly from the future sequence in Section 3:

$$\big[\mathrm{Agg}_{F,N}\big(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\big)\big]_{m,n} = x_{t+m,n}, \quad m = 1, \dots, F, \quad n \in \mathcal{N}. \tag{5}$$

The target at (r_k, s_l) is

$$\begin{aligned}\mathrm{Agg}_{r_k,s_l}\big(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\big) &= \mathcal{P}_{L\to l}\big(\mathcal{T}_{K\to k}\big(\mathrm{Agg}_{F,N}\big(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\big)\big)\big) \\ &= \mathcal{T}_{K\to k}\big(\mathcal{P}_{L\to l}\big(\mathrm{Agg}_{F,N}\big(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\big)\big)\big) \in \mathbb{R}^{r_k\times s_l}.\end{aligned} \tag{6}$$

The two forms agree because $\mathcal{T}$ acts on rows and $\mathcal{P}$ on columns. These maps support recursive aggregation along successive selected temporal scales and the nested spatial hierarchy.

The forecast state at (r_k, s_l) is $\hat{\mathbf{f}}_{[t]}^{(r_k,s_l)} \in \mathbb{R}^{r_k\times s_l}$, where $[t]$ marks the prediction origin. At the finest scale, it follows directly from the forecast sequence in Section 3: $[\hat{\mathbf{f}}_{[t]}^{(F,N)}]_{m,n} = [\hat{\mathbf{x}}_{t+m}]_n$.

Definition 4.2 (Scale-Specific Spatio-Temporal Forecasting)**.** Given historical observations $\{(\mathbf{C}_\tau, \mathbf{x}_\tau)\}_{\tau=t-H+1}^{t}$, adjacency matrix $\mathbf{A}$, and a scale pair $(r_k, s_l) \in \mathcal{R} \times \mathcal{S}$, the forecasting problem is to learn a forecast state $\hat{\mathbf{f}}_{[t]}^{(r_k,s_l)} \in \mathbb{R}^{r_k\times s_l}$ that estimates the scale-aligned future target. With model parameters Θ, the objective is

$$\min_{\Theta} \quad \mathcal{L}\Big(\mathrm{Agg}_{r_k,s_l}\big(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\big), \hat{\mathbf{f}}_{[t]}^{(r_k,s_l)}\Big), \tag{7}$$

where the forecast depends on the given observations, $\mathbf{A}$, and scale pair through Θ, and $\mathcal{L}$ compares the target and forecast at that scale. At (F, N), this objective recovers Eq. (2) through the entrywise correspondence above.

4.2 Progressive Forecast Refinement

Progressive forecast refinement first expands the current coarser forecast to a finer scale pair, then separates a correction according to its aggregation at the current scale. The correction is the additive matrix $\mathbf{\Delta}^{(r_i,s_j)} \in \mathbb{R}^{r_i\times s_j}$ that updates the expanded forecast at the finer scale pair. Let $\hat{\mathbf{f}}_{[t]}^{(r_k,s_l)} \in \mathbb{R}^{r_k\times s_l}$ be the current coarser forecast over a scale pair (r_k, s_l) and let (r_i, s_j) be a finer scale pair, where $r_i \geq r_k$, $r_k \mid r_i$, $s_j \geq s_l$, and $(i, j) \neq (k, l)$. With $\eta_{k\to i} = r_i/r_k \in \mathbb{N}$, temporal position m corresponds to

$$\mathcal{B}_{k\to i,m} = \{(m-1)\eta_{k\to i} + 1, \dots, m\eta_{k\to i}\}$$

at scale r_i. For $j > l$ and $u \in \mathcal{N}^{(s_l)}$, its descendants at scale s_j are

$$\Gamma_u^{(j|l)} = \Big\{v \in \mathcal{N}^{(s_j)} : (\psi_l \circ \cdots \circ \psi_{j-1})(v) = u\Big\}, \qquad \Gamma_u^{(l|l)} = \{u\}.$$

Together, $\mathcal{B}_{k\to i,m}$ and $\Gamma_u^{(j|l)}$ identify the entries at (r_i, s_j) associated with entry (m, u) at (r_k, s_l). We copy each current-scale entry to its associated entries at the finer scale pair by an expansion function of the current forecast over (r_k, s_l) to (r_i, s_j) as $\mathbf{J}^{(k,l)\to(i,j)} \in \mathbb{R}^{r_i\times s_j}$, where

$$\mathbf{J}_{a,v}^{(k,l)\to(i,j)} = \Big[\hat{\mathbf{f}}_{[t]}^{(r_k,s_l)}\Big]_{m,u}, \qquad a \in \mathcal{B}_{k\to i,m}, \quad v \in \Gamma_u^{(j|l)}. \tag{8}$$

Starting from this expanded forecast, the following steps decompose the correction at (r_i, s_j) into the forecast preserved after aggregation to (r_k, s_l) and the remaining finer-scale forecast.

To compare a quantity at (r_i, s_j) with the current scale, the operators in Definition 4.1 give the joint aggregation $\mathcal{A}_{(i,j)\to(k,l)} = \mathcal{P}_{j\to l} \circ \mathcal{T}_{i\to k} : \mathbb{R}^{r_i \times s_j} \to \mathbb{R}^{r_k \times s_l}$. Specifically, this aggregation recovers the coarser forecast from its expansion by $\mathcal{A}_{(i,j)\to(k,l)}(\mathbf{J}^{(k,l)\to(i,j)}) = \hat{\mathbf{f}}_{[t]}^{(r_k,s_l)}$. Denote the correction produced by the forecast network at the selected finer scale pair by $\mathbf{\Delta}^{(r_i,s_j)} \in \mathbb{R}^{r_i \times s_j}$. Its aggregation to (r_k, s_l) is $\mathbf{P}^{(k,l)\to(i,j)} = \mathcal{A}_{(i,j)\to(k,l)}\left(\mathbf{\Delta}^{(r_i,s_j)}\right) \in \mathbb{R}^{r_k \times s_l}$. Expanding $\mathbf{P}^{(k,l)\to(i,j)}$ back to (r_i, s_j) gives the correction component $\mathbf{L}^{(k,l)\to(i,j)} \in \mathbb{R}^{r_i \times s_j}$, which is $\mathbf{L}_{a,v}^{(k,l)\to(i,j)} = \mathbf{P}_{m,u}^{(k,l)\to(i,j)}$, where $a \in \mathcal{B}_{k\to i,m}$ and $v \in \Gamma_u^{(j|l)}$. The remaining correction is

$$\mathbf{D}^{(k,l)\to(i,j)} = \mathbf{\Delta}^{(r_i,s_j)} - \mathbf{L}^{(k,l)\to(i,j)} \in \mathbb{R}^{r_i \times s_j}. \tag{9}$$

By construction, $\mathcal{A}_{(i,j)\to(k,l)}(\mathbf{D}^{(k,l)\to(i,j)}) = 0$.

Definition 4.3 (Progressive Forecast Refinement)**.** Combining the expanded forecast with the two correction components gives the candidate forecast at (r_i, s_j):

$$\widetilde{\mathbf{f}}_{[t]}^{(r_i,s_j)|(r_k,s_l)} = \mathbf{J}^{(k,l)\to(i,j)} + \lambda_{k,l\to i,j}\mathbf{L}^{(k,l)\to(i,j)} + \mathbf{D}^{(k,l)\to(i,j)} \in \mathbb{R}^{r_i \times s_j}, \tag{10}$$

where $\lambda_{k,l\to i,j} \in (0, 1)$ weights the correction component preserved by $\mathcal{A}_{(i,j)\to(k,l)}$.

Proposition 4.4 (Cross-Scale Forecast Consistency)**.** *The candidate forecast in Definition 4.3 satisfies*

$$\mathcal{A}_{(i,j)\to(k,l)}\left(\widetilde{\mathbf{f}}_{[t]}^{(r_i,s_j)|(r_k,s_l)}\right) = \hat{\mathbf{f}}_{[t]}^{(r_k,s_l)} + \lambda_{k,l\to i,j}\mathbf{P}^{(k,l)\to(i,j)}. \tag{11}$$

Proof. Equation (8) gives $\mathcal{A}_{(i,j)\to(k,l)}(\mathbf{J}^{(k,l)\to(i,j)}) = \hat{\mathbf{f}}_{[t]}^{(r_k,s_l)}$. The same expansion gives $\mathcal{A}_{(i,j)\to(k,l)}(\mathbf{L}^{(k,l)\to(i,j)}) = \mathbf{P}^{(k,l)\to(i,j)}$, while $\mathcal{A}_{(i,j)\to(k,l)}(\mathbf{D}^{(k,l)\to(i,j)}) = 0$. The result is proved. □

4.3 Adaptive Scale Selection

Scale selection determines the scale pair for next finer forecast. Local selection observes the current coarser forecast. For a nonterminal scale pair $(r_k, s_l) \neq (F, N)$, we define its admissible subsequent scales as

$$\mathcal{C}_{k,l} = \left\{ (r_i, s_j) \in \mathcal{R} \times \mathcal{S} \;\middle|\; r_i \geq r_k,\ s_j \geq s_l,\ (i,j) \neq (k,l),\ \frac{r_i}{r_k} \in \mathbb{N} \right\}. \tag{12}$$

Given the current forecast, Definition 4.3 yields candidate $\widetilde{\mathbf{f}}_{[t]}^{(r_i,s_j)|(r_k,s_l)}$ for each $(r_i, s_j) \in \mathcal{C}_{k,l}$. With the future target available for supervision, its immediate gain is

$$g_{k,l}(r_i, s_j) = \mathcal{L}\left(\mathbf{J}^{(k,l)\to(i,j)}, \mathrm{Agg}_{r_i,s_j}\left(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\right)\right) - \mathcal{L}\left(\widetilde{\mathbf{f}}_{[t]}^{(r_i,s_j)|(r_k,s_l)}, \mathrm{Agg}_{r_i,s_j}\left(\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}\right)\right). \tag{13}$$

Both losses use the candidate-scale target, so their difference measures the correction beyond lifting the current forecast. A target-informed local oracle selects

$$(r_{k^\star}, s_{l^\star}) = \underset{(r_i,s_j)\in\mathcal{C}_{k,l}}{\arg\max}\ g_{k,l}(r_i, s_j). \tag{14}$$

At inference, the observable state is$\mathcal{Z}_{k,l} = \left(\{(\mathbf{C}_\tau, \mathbf{x}_\tau)\}_{\tau=t-H+1}^{t}, \hat{\mathbf{f}}_{[t]}^{(r_k,s_l)}, r_k, s_l\right)$. An inference-time local selector would approximate the oracle from this state:

Definition 4.5 (Adaptive Scale Selection)**.** A scale selector Φ_ϕ maps the current forecasting state $\mathcal{Z}_{k,l}$ and its admissible set $\mathcal{C}_{k,l}$ to the subsequent temporal–spatial scale, $(r_{k'}, s_{l'}) = \Phi_\phi\left(\mathcal{Z}_{k,l}, \mathcal{C}_{k,l}\right), (r_{k'}, s_{l'}) \in \mathcal{C}_{k,l}$, where ϕ denotes its parameters. Equation (14) supplies a supervised target for this general decision problem.

The selected candidate becomes $\hat{\mathbf{f}}_{[t]}^{(r_{k'},s_{l'})}$. Every nonterminal pair admits (F, N). Let $\mathfrak{O}$ denote the set of all finite sequences of scale pairs in $\mathcal{R} \times \mathcal{S}$ such that each pair after the first belongs to the admissible set of its predecessor and the last pair is (F, N). A complete legal scale-selection sequence of length $J \geq 1$ is

$$\Omega = \left((r_{k_1}, s_{l_1}), \ldots, (r_{k_J}, s_{l_J})\right) \in \mathfrak{O}. \tag{15}$$

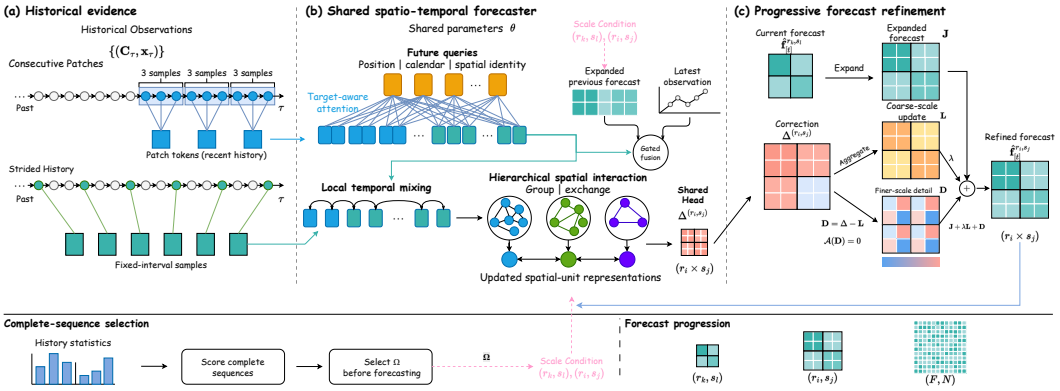


Figure 1: Overview of ProST. (a) Consecutive patches and strided history provide complementary temporal evidence. (b) The shared scale-conditioned forecaster combines target-aware temporal processing with hierarchical spatial interaction to predict a correction. (c) Refinement combines the expanded forecast with a weighted coarse-scale update and zero-aggregate detail.

5 Neural Realization

ProST realizes the refinement in Section 4.2 with a shared spatio-temporal forecaster parameterized by θ, within the task-level function f. A separate history-conditioned predictor selects the complete scale-selection sequence before forecasting. Figure 1 shows how historical evidence and the preceding forecast produce scale-specific corrections for progressive refinement.

5.1 Spatio-Temporal Forecaster

The forecaster encodes $\{(\mathbf{C}_\tau, \mathbf{x}_\tau)\}_{\tau=t-H+1}^{t}$ as consecutive patches and strided histories through learned projections and residual MLPs. At a selected pair (r_i, s_j), queries combine future temporal position, spatial identity, and calendar context from $\mathbf{C}_\tau$. Target-aware attention retrieves evidence from both historical views for the r_i future positions. Learned gates fuse this evidence with the latest observation and an encoding of $\mathbf{J}^{(k,l)\to(i,j)}$. The preceding and selected scale pairs condition feature-wise scaling and shifting, followed by a shared local temporal operator that mixes neighboring future positions.

The resulting features undergo spatial interaction over $\mathcal{N}^{(s_j)}$. Learned source and destination embeddings provide affinities for balanced hierarchical grouping, instantiating ψ_l. Shared low-rank projections and sparse affinity-weighted aggregation exchange information between spatial units; a scale-conditioned gate controls the residual feature update. A shared prediction head and hierarchical readout produce $\mathbf{\Delta}^{(r_i,s_j)} \in \mathbb{R}^{r_i \times s_j}$ for Definition 4.3. Appendices A.2.1–A.2.3 give the module equations.

5.2 Scale Selection

A history-conditioned predictor scores the complete sequences in $\mathfrak{X} \subseteq \mathfrak{O}$ before the shared forecaster runs. The implementation uses an extremely randomized tree ensemble (Geurts et al., 2006) on historical summary statistics, with one loss prediction per candidate sequence. The evaluated selector predicts sequence-centered terminal-loss residuals and combines them with profiled cost to rank feasible sequences. The chosen Ω determines the forecaster's successive scale conditions. Appendix C specifies the residual targets, cost calibration, and fixed-reference settings; Appendix A.2.5 gives the general terminal-loss formulation.

5.3 Training

The shared forecaster is first trained on a fixed legal sequence, then further trained with a mixture of fixed and sampled sequences while holding the discrete spatial hierarchy fixed. Masked MAE

Table 1: Test MAE with $H = 12$. Columns specify F. Bold and underlined entries indicate the lowest and second-lowest means in each column. The ProST entries use fixed-sequence training.

	PEMS04		PEMS07		PEMS08		KnowAir	
Method	12	48	12	48	12	48	12	48
Spatio-Temporal Forecasting Models								
DCRNN	$21.24_{\pm.083}$	$25.91_{\pm.146}$	$25.24_{\pm.097}$	$31.55_{\pm.176}$	$16.89_{\pm.061}$	$21.11_{\pm.119}$	$18.42_{\pm.072}$	$23.39_{\pm.137}$
STGCN	$20.03_{\pm.058}$	$24.04_{\pm.103}$	$21.64_{\pm.071}$	$26.62_{\pm.128}$	$15.64_{\pm.044}$	$19.08_{\pm.089}$	$18.10_{\pm.055}$	$22.63_{\pm.108}$
AGCRN	$19.69_{\pm.047}$	$23.24_{\pm.091}$	$20.83_{\pm.059}$	$25.20_{\pm.115}$	$15.67_{\pm.036}$	$18.80_{\pm.074}$	$17.68_{\pm.049}$	$21.75_{\pm.097}$
MTGNN	$19.98_{\pm.052}$	$23.38_{\pm.102}$	$21.16_{\pm.064}$	$25.82_{\pm.127}$	$15.03_{\pm.039}$	$18.34_{\pm.081}$	$17.20_{\pm.053}$	$21.16_{\pm.105}$
STGODE	$20.84_{\pm.067}$	$24.59_{\pm.126}$	$22.99_{\pm.079}$	$27.82_{\pm.148}$	$16.81_{\pm.051}$	$20.17_{\pm.101}$	$17.05_{\pm.060}$	$20.97_{\pm.116}$
STID	$18.59_{\pm.026}$	$21.56_{\pm.052}$	$19.54_{\pm.033}$	$23.06_{\pm.069}$	$14.23_{\pm.019}$	$\underline{16.79_{\pm.041}}$	$16.62_{\pm.027}$	$19.94_{\pm.057}$
STAEformer	$18.23_{\pm.031}$	$20.67_{\pm.059}$	$19.62_{\pm.038}$	$\underline{22.95_{\pm.078}}$	$13.98_{\pm.021}$	$16.92_{\pm.047}$	$16.18_{\pm.029}$	$19.09_{\pm.061}$
PDFormer	$18.32_{\pm.038}$	$20.88_{\pm.074}$	$19.83_{\pm.046}$	$23.20_{\pm.093}$	$\mathbf{13.58_{\pm.024}}$	$17.11_{\pm.053}$	$16.05_{\pm.034}$	$18.94_{\pm.070}$
HyperD	$\underline{18.20_{\pm.029}}$	$\underline{20.57_{\pm.064}}$	$\underline{19.37_{\pm.037}}$	$23.63_{\pm.086}$	$\underline{13.59_{\pm.020}}$	$16.99_{\pm.046}$	$15.74_{\pm.028}$	$18.58_{\pm.060}$
Multi-Scale and Iterative Forecasting Models								
N-BEATS	$22.80_{\pm.088}$	$27.13_{\pm.164}$	$24.90_{\pm.103}$	$30.88_{\pm.196}$	$17.55_{\pm.067}$	$21.59_{\pm.132}$	$18.30_{\pm.076}$	$22.88_{\pm.151}$
N-HiTS	$21.75_{\pm.069}$	$25.01_{\pm.131}$	$23.60_{\pm.083}$	$27.85_{\pm.162}$	$16.70_{\pm.052}$	$19.71_{\pm.104}$	$17.55_{\pm.061}$	$20.88_{\pm.121}$
TimeMixer	$19.21_{\pm.043}$	$21.90_{\pm.081}$	$20.57_{\pm.049}$	$23.86_{\pm.101}$	$15.22_{\pm.029}$	$17.81_{\pm.061}$	$16.10_{\pm.036}$	$18.84_{\pm.076}$
DiffSTG	$21.90_{\pm.113}$	$24.53_{\pm.207}$	$23.80_{\pm.137}$	$27.13_{\pm.249}$	$17.74_{\pm.086}$	$20.22_{\pm.164}$	$16.55_{\pm.095}$	$19.36_{\pm.183}$
Scaleformer	$22.74_{\pm.080}$	$25.96_{\pm.140}$	$27.58_{\pm.100}$	$31.74_{\pm.180}$	$20.69_{\pm.070}$	$24.16_{\pm.130}$	$16.96_{\pm.060}$	$20.15_{\pm.110}$
Pathformer	$21.63_{\pm.060}$	$24.58_{\pm.110}$	$26.41_{\pm.080}$	$29.88_{\pm.140}$	$18.11_{\pm.050}$	$21.34_{\pm.100}$	$16.43_{\pm.050}$	$19.21_{\pm.090}$
STORM	$18.55_{\pm.040}$	$21.11_{\pm.070}$	$19.86_{\pm.050}$	$23.72_{\pm.090}$	$14.18_{\pm.030}$	$17.08_{\pm.060}$	$\underline{15.09_{\pm.040}}$	$\underline{18.31_{\pm.080}}$
ProST	$\mathbf{18.14_{\pm.049}}$	$\mathbf{20.30_{\pm.030}}$	$\mathbf{19.26_{\pm.066}}$	$\mathbf{22.93_{\pm.042}}$	$13.94_{\pm.062}$	$\mathbf{16.61_{\pm.039}}$	$\mathbf{13.92_{\pm.025}}$	$\mathbf{17.65_{\pm.023}}$

supervises each visited forecast against its scale-aligned target from Definition 4.2, with greater weight on the terminal prediction. Gradients pass through successive forecasts, so later losses also train the states that condition refinement. All sequences share the temporal, spatial, and prediction modules. After forecaster training, its parameters are frozen and cached sequence losses supervise the history-conditioned predictor. Appendix A.2.4 gives the loss combinations, and Section 6.1 gives the training configuration.

6 Experiments

The experiments examine whether explicit intermediate forecasts improve prediction, which parts of temporal–spatial refinement account for that improvement, and whether history-conditioned scale selection allocates computation better than a fixed sequence. The main comparison evaluates ProST with a fixed scale-selection sequence; the selection study evaluates the complete-sequence realization in Section 5.2.

6.1 Experimental Setup

Data. PEMS04, PEMS07, and PEMS08 record traffic flow at 5-minute intervals; KnowAir records $PM_{2.5}$ at 3-hour intervals. Table 2 summarizes their sizes. Forecasting samples are split chronologically in a 6:2:2 ratio. Z-score statistics are fitted on training data, and MAE is computed after inverse transformation. The training-procedure and refinement studies use $H = 12$ on all four datasets; selection is evaluated on PEMS04 and KnowAir.

Table 2: Dataset Description.

Dataset	Steps	Nodes	Interval
PEMS04	16,992	307	5 min
PEMS07	28,224	883	5 min
PEMS08	17,856	170	5 min
KnowAir	11,688	184	3 h

Training and hardware. Experiments use Python 3.9, PyTorch 2.8, CUDA 12.8, an Intel Core i9-14900K CPU, and two RTX 4090 GPUs (24 GB each). The audited $H = F = 12$ runs use seeds 1–3, with sample standard deviation ($\mathrm{ddof} = 1$).

Spatio-temporal forecaster. The shared forecaster in Section 5.1 is first trained on a fixed legal sequence Ω. Adam uses learning rate 10^{-3}, weight decay 10^{-4}, and 100 epochs, halving the learning rate at epochs 1, 35, 60, 80, and 95. The best validation-MAE checkpoint is retained.

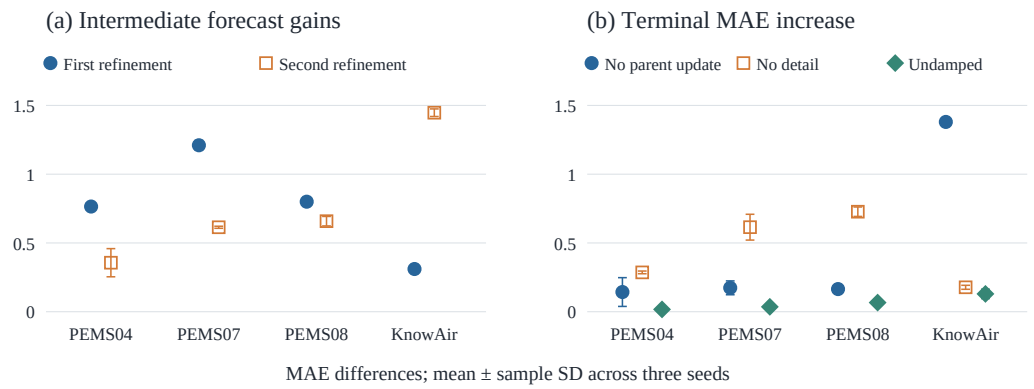


Figure 2: Refinement evidence ($H = F = 12$, three seeds). (a) Window-averaged MAE reduction. (b) Paired terminal-MAE increase under frozen-weight recursive interventions, using the batch-averaged benchmark metric. Positive values indicate refinement gains in (a) and harm from interventions in (b). Bars show sample standard deviations.

Training over scale selections. Following Section 5.3, training continues from this checkpoint with the hierarchy fixed: 60% of batches retain the fixed sequence and 40% sample legal sequences. Sampled-sequence losses use $\eta = 0.25$ (Appendix A.2.4). A fresh Adam optimizer uses learning rate 5×10^{-5}, weight decay 10^{-4}, 100 epochs, halving at epochs 50, 75, and 90, and gradient clipping at 5.

Scale selection. The predictor uses 59 historical features and 300 extremely randomized trees (Geurts et al., 2006), with feature fraction 0.7 and minimum leaf size in $\{10, 25, 50, 100\}$. For $F = 12$, we use $\mathcal{R} = \{2, 3, 4, 6, 12\}$, $\mathcal{S} = \{\lceil N/20 \rceil, \lceil N/10 \rceil, \lceil N/4 \rceil, N\}$, and the fixed sequence $\Omega = ((3, s_1), (6, s_2), (12, N))$. The evaluated set $\mathfrak{X}$ contains 22 legal sequences ending at $(12, N)$: one direct prediction, 15 two-pair sequences, and six three-pair sequences.

Baselines. The baselines in Table 1 cover graph-based forecasting (DCRNN, STGCN, AGCRN, MTGNN, and STGODE) (Li et al., 2017; Yu et al., 2017; Bai et al., 2020; Wu et al., 2020; Fang et al., 2021), identity- and attention-based forecasting (STID, STAEformer, and PDFormer) (Shao et al., 2022a; Liu et al., 2023a; Jiang et al., 2023), and periodicity decomposition (HyperD) (Shao et al., 2026). We also compare with N-BEATS, N-HiTS, TimeMixer, and DiffSTG (Oreshkin et al., 2019; Challu et al., 2023; Wang et al., 2024; Wen et al., 2023). The closest multi-scale comparisons include Scaleformer for iterative forecast refinement (Shabani et al., 2023), Pathformer for adaptive multi-scale computation (Chen et al., 2024), and STORM for cross-scale spatio-temporal modeling with multi-scale future predictions (Huang et al., 2026).

6.2 Forecasting Accuracy

Table 1 reports ProST with a fixed scale-selection sequence. At $F = 12$, it has the lowest mean MAE on PEMS04, PEMS07, and KnowAir, while PDFormer obtains the lowest mean on PEMS08. On KnowAir, MAE decreases from STORM's 15.09 to 13.9228 (7.73%).

Training across sequences. Training over scale selections changes fixed-sequence test MAE by -0.41% to $+0.20\%$ while reducing mean MAE across the same 22 legal sequences by 16.71%, 15.28%, 15.63%, and 2.57% on PEMS04, PEMS07, PEMS08, and KnowAir. Both checkpoints are evaluated on identical test samples under every candidate sequence. We compute the mean and maximum terminal MAE across sequences within each seed before averaging across seeds.

Mean and worst-sequence errors decrease in every evaluated cost interval at unchanged per-sequence FLOPs. On PEMS04, worst-sequence MAE falls from 39.5092 to 18.4788; on KnowAir, it falls from 15.9103 to 14.8891. These reductions show that training over scale selections improves forecasts for sequences poorly served by fixed-sequence training. PEMS07 and PEMS08 have substantial seed variability, with one run on each retaining the initial checkpoint. Appendix A.3 reports fixed-sequence accuracy, worst-sequence errors, and cost-grouped results.

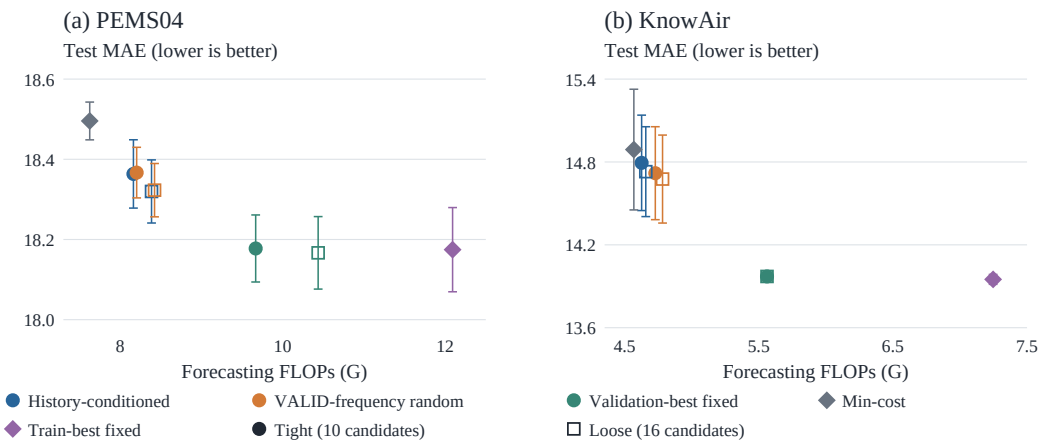


Figure 3: Selection accuracy versus forecasting cost. Circles/squares mark Tight/Loose; diamonds mark fixed references. Points show means and bars show MAE sample standard deviations across three seeds. Random selection uses validation frequencies, with potentially different test cost. Lower MAE and fewer FLOPs are preferable.

6.3 Mechanism of Progressive Refinement

Intermediate forecast refinement. At $H = F = 12$, we evaluate frozen checkpoints after training over scale selections. Each gain compares the expanded preceding forecast with the refined forecast against the same aligned target, with equal weight per test window. Figure 2(a) shows positive mean gains at both steps on all four datasets; 84.11–96.22% of windows improve. Appendix B provides seed-wise distributions, including negative gains, and forecast heatmaps.

Parent/detail interventions. With parameters frozen, each intervened forecast conditions the next prediction. Removing the parent update, removing detail, or using an undamped correction increases terminal MAE in every seed and dataset (Figure 2(b)). Parent removal affects KnowAir most; detail removal affects PEMS08 most. The undamped update also increases error, although its effect is smaller than removing either component on every dataset. These comparisons isolate the executed correction within the trained model, with each modified forecast propagated through the remaining refinements. Local gains need not predict terminal accuracy because subsequent conditions change. For example, removing the parent update on PEMS04 raises the first-step gain from 0.7648 to 0.7926, yet worsens terminal MAE. Appendix B.3 reports absolute errors, paired differences, and per-horizon results.

6.4 Adaptive Scale Selection

We evaluate PEMS04 and KnowAir at $H = F = 12$ with three seeds. Adaptive–Tight and Adaptive–Loose select among 10 and 16 feasible sequences, respectively. Min-cost predicts directly at $(12, N)$; Train-best fixed uses the sequence with lowest mean training MAE. Figure 3 shows that both adaptive settings improve on Min-cost while using more FLOPs. Relative to Train-best fixed, Tight saves 32.5–36.2% and Loose saves 30.7–35.7% of forecasting FLOPs, with higher MAE. Costs exclude selector overhead; Appendix C defines the cost ceilings and reports exact values.

Validation-best fixed achieves lower MAE than the selector, generally at higher cost. Against input-independent random selection using validation frequencies, Tight/Loose improves MAE by only $0.0034/0.0032$ on PEMS04 and worsens it by $0.0749/0.0536$ on KnowAir. History conditioning therefore yields no consistent accuracy gain at these operating points. The two budgets change both the feasible candidates and the cost penalty. Direct prediction remains the most frequent choice in the displayed selection distributions, especially on KnowAir; the looser setting shifts some choices toward longer or finer intermediate sequences. Appendix C reports paired uncertainty, oracle regret, selection frequencies, and the separate post-hoc cost-matched comparison.

AI USE STATEMENT

Generative AI assisted with manuscript drafting and editing, literature discovery and source checking, and refinement of mathematical definitions, notation, and exposition. It also provided feedback on experimental design and interpretation of reported results, assisted with figure design and preparation, and supported preparation and anonymity checks of the code supplement. The authors reviewed and verified the AI-assisted material and take responsibility for the final text, mathematical claims, experimental results, citations, and released artifacts.

REPRODUCIBILITY STATEMENT

Section 4 defines the forecast states, refinement, and selection scopes; Section 5 gives the shared neural realization. Appendix A describes the temporal and spatial modules, forecast initialization, and training objectives. The proof of Proposition 4.4 accompanies its statement. Our code is available in `https://anonymous.4open.science/r/ProST-1B6C/`

REFERENCES

Yinghao Ai, Yukai Zhou, Ruoxi Jiang, Junyi An, Chao Qu, Zhijian Zhou, Shiyu Wang, Fenglei Cao, Zenglin Xu, Furao Shen, and Yuan Qi. Nested spatio-temporal time series forecasting. In *International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2605.16447`.

George Athanasopoulos, Rob J. Hyndman, Nikolaos Kourentzes, and Fotios Petropoulos. Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1):60–74, 2017. doi: 10.1016/j.ejor.2017.02.046.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/hash/75c45fca2aa416ada062b26cc4fb7641-Abstract.html`.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Lingxiao Cao, Bin Wang, Guiyuan Jiang, Yanwei Yu, and Junyu Dong. Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11463–11471, 2025.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Haolong Chen, Liang Zhang, Zhengyuan Xin, and Guangxu Zhu. STM3: Mixture of multiscale Mamba for long-term spatio-temporal time-series prediction. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. URL `https://arxiv.org/abs/2508.12247`.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=8KYeilT3Ow`.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lJkOCMP2aW.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are AllSet: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hpBTIv2uy_E.

Andrea Cini, Danilo Mandic, and Cesare Alippi. Graph-based time series clustering for end-to-end hierarchical forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8985–8999. PMLR, 2024. URL https://proceedings.mlr.press/v235/cini24a.html.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Tommaso Di Fonzo and Daniele Girolimetto. Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives. *International Journal of Forecasting*, 39(1):39–57, 2023a. doi: 10.1016/j.ijforecast.2021.08.004.

Tommaso Di Fonzo and Daniele Girolimetto. Spatio-temporal reconciliation of solar forecasts. *Solar Energy*, 251:13–29, 2023b. doi: 10.1016/j.solener.2023.01.003.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. TimeMosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20790–20798, 2026. doi: 10.1609/aaai.v40i25.39218.

Yuchen Fang, Yanjun Qin, Haiyong Luo, Fang Zhao, Bingbing Xu, Liang Zeng, and Chenxing Wang. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pp. 517–529. IEEE, 2023.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 364–373, 2021.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1. URL https://doi.org/10.1007/s10994-006-6226-1.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. Hierarchical graph convolution network for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 151–159, 2021. doi: 10.1609/aaai.v35i1.16088. URL https://ojs.aaai.org/index.php/AAAI/article/view/16088.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proceedings of the VLDB Endowment*, 17(5):1081–1090, 2024.

Sicheng He, Junzhong Ji, and Minglong Lei. Decomposed spatio-temporal Mamba for long-term traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11772–11780, 2025. doi: 10.1609/aaai.v39i11.33281. URL https://ojs.aaai.org/index.php/AAAI/article/view/33281.

Qihe Huang, Zhengyang Zhou, Yangze Li, Jiaming Ma, Kuo Yang, Binwu Wang, Xu Wang, and Yang Wang. STORM: Synergistic cross-scale spatio-temporal modeling for weather forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9ad996b5c45130de2bc00b60d8607904-Abstract-Conference.html.

Yuting Huang, Ziquan Fang, Zhihao Zeng, Lu Chen, and Yunjun Gao. Causal spatio-temporal prediction: An effective and efficient multi-modal approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b71169b4db5448c494c91db848f060be-Abstract-Conference.html.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4365–4373, 2023.

Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/5d84236751fe6d25dc06db055a3180b0-Abstract.html.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, pp. 11906–11917. PMLR, 2022.

Xuan-May Le, Minh-Tuan Tran, Ling Luo, Uwe Aickelin, Dinh Phung, and Trung Le. Efficient test-time scaling for LLM-based time series forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3818201. URL https://arxiv.org/abs/2608.08675.

Mengzhang Li and Zhanxing Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4189–4196, 2021.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. Open spatio-temporal foundation models for traffic prediction. *ACM Transactions on Intelligent Systems and Technology*, 2025. doi: 10.1145/3773912. URL https://doi.org/10.1145/3773912.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 4125–4129, 2023a.

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

Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/28d38c036365420f61ce03300418e44a-Abstract-Conference.html.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv preprint arXiv:2201.02849*, 2022.

Syama Sundar Rangapuram, Shubham Kapoor, Rajbir Singh Nirwan, Pedro Mercado, Tim Januschowski, Yuyang Wang, and Michael Bohlke-Schneider. Coherent probabilistic forecasting of temporal hierarchies. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 9362–9376. PMLR, 2023. URL https://proceedings.mlr.press/v206/rangapuram23a.html.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. DYffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/8df90a1440ce782d1f5607b7a38f2531-Abstract-Conference.html.

Mohammad Amin Shabani, Amir Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sCrnllCtjoE.

Minlan Shao, Zijian Zhang, Yili Wang, Yiwei Dai, Xu Shen, and Xin Wang. HyperD: Hybrid periodicity decoupling framework for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15689–15697, 2026. doi: 10.1609/aaai.v40i18.38599. URL https://ojs.aaai.org/index.php/AAAI/article/view/38599.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 4454–4458, 2022a.

Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1567–1577, 2022b. doi: 10.1145/3534678.3539396. URL https://doi.org/10.1145/3534678.3539396.

Lifeng Shen, Weiyu Chen, and James T. Kwok. Multi-resolution diffusion models for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d64740dd69bcc90ba225a182984b81ba-Abstract-Conference.html.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31613–31632, 2023. URL https://proceedings.mlr.press/v202/shirzad23a.html.

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.

Jindong Tian, Yifei Ding, Ronghui Xu, Hao Miao, Chenjuan Guo, and Bin Yang. ARROW: An adaptive rollout and routing method for global weather forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/59fb5cf4683eaec4c56918a69637c9d2-Abstract-Conference.html.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *Proceedings of the 31st ACM international conference on advances in geographic information systems*, pp. 1–12, 2023.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/1efa39bcaec6f3900149160693694536-Abstract.html.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in neural information processing systems*, 36:76656–76679, 2023.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. Unist: A prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4095–4106, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Haoyu Zhang, Wentao Zhang, Hao Miao, Xinke Jiang, Yuchen Fang, and Yifan Zhang. STRAP: Spatio-temporal pattern retrieval for out-of-distribution generalization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/ab062c6281873221743383f9619e07d2-Abstract-Conference.html.

Hong Zhang, Fangzheng Qi, Yu Zhang, Yijie Qin, and Yayong Li. A traffic flow forecasting model based on dynamic graph learning and temporally adaptive attention. *Safety Science*, 195:107063, 2026a. doi: 10.1016/j.ssci.2025.107063. URL https://doi.org/10.1016/j.ssci.2025.107063.

Zichi Zhang, Phi Hung Nguyen, Ngoc Phu Doan, Viet-Hung Tran, Xuan Hoang Nguyen, Hui Wang, Hans Vandierendonck, and Son Thai Mai. HierarNet: Independent interactive hierarchical disease outbreak forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 39628–39636, 2026b. doi: 10.1609/aaai.v40i46.41315.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1234–1241, 2020.

Chuanpan Zheng, Xiaoliang Fan, Shirui Pan, Haibing Jin, Zhaopeng Peng, Zonghan Wu, Cheng Wang, and Philip S Yu. Spatio-temporal joint graph convolutional networks for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):372–385, 2023.

Fan Zhou, Chen Pan, Lintao Ma, Yu Liu, Siqiao Xue, James Zhang, Jun Zhou, Hongyuan Mei, Weitao Lin, Zi Zhuang, Wenxin Ning, and Yunhua Hu. GMP-AR: Granularity message passing and adaptive reconciliation for temporal hierarchy forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9414–9421, 2024. doi: 10.1609/aaai.v38i8.28795.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *arXiv preprint arXiv:2012.07436*, 2020.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

APPENDIX TABLE OF CONTENTS

A Method Details **16**

A.1 Notation . . . 16

A.2 Multi-Scale Spatio-Temporal Forecaster . . . 17

A.2.1 Temporal Module . . . 17

A.2.2 Spatial Module . . . 19

A.2.3 Progressive Multi-Scale Forecasting . . . 21

A.2.4 Training Objective . . . 21

A.2.5 Complete-Sequence Selector . . . 22

A.3 Training-procedure comparison . . . 22

B Progressive Refinement Analysis **23**

B.1 Full-test refinement gains . . . 23

B.2 Evaluation and sample selection . . . 23

B.3 Recursive correction interventions . . . 24

B.4 Scale-selection sequence error and cost . . . 29

C Adaptive Scale Selection Analysis **30**

C.1 Cost and selection protocol . . . 30

C.2 Operating points and paired comparisons . . . 30

C.3 Selection frequencies and matched-cost comparison . . . 30

A METHOD DETAILS

A.1 NOTATION

Tables 3–4 summarize the notation established in Sections 3–4 and reused in Section 5. The prediction origin $[t]$, scale pair (r_k, s_l), and element indices (m, u) have distinct roles.

Table 3: Problem and scale notation from Sections 3–4.1.

Symbol	Meaning
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	Graph, node set, and edge set; N is the number of nodes.
t, τ, H, F	Prediction origin, observation time, history length in Eq. (1), and future horizon.
$\mathbf{C}_\tau \in \mathbb{R}^{N \times D}$	Node features at time τ, with D feature channels.
$\mathbf{x}_\tau, \hat{\mathbf{x}}_\tau \in \mathbb{R}^N$	Target and forecast at one time step. Historical observations are pairs $(\mathbf{C}_\tau, \mathbf{x}_\tau)$.
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Adjacency matrix in the task-level forecasting function.
$f, \Theta, \mathcal{L}$	Task-level forecasting function, its parameters, and forecasting loss.
$\mathcal{R} = \{r_k\}_{k=1}^K, \mathcal{S} = \{s_l\}_{l=1}^L$	Ordered temporal and spatial scales; $r_K = F$ and $s_L = N$.
$\mathcal{N}^{(s_l)}, \psi_l$	Spatial units at scale s_l and the map from immediately finer units to their parents.
$\Gamma_u^{(j\|l)}$	Descendants of unit u at finer spatial scale s_j; $\Gamma_u^{(l\|l)} = \{u\}$.
$\eta_{k\to i}, \mathcal{B}_{k\to i,m}$	Temporal scale ratio r_i/r_k and fine positions associated with current position m.

Symbol	Meaning (continued)
$\mathcal{T}_{i\to k}$	Temporal averaging, $\mathbb{R}^{r_i\times d}\to\mathbb{R}^{r_k\times d}$, for arbitrary spatial width d.
$\mathcal{P}_{j\to l}$	Recursive spatial averaging, $\mathbb{R}^{q\times s_j}\to\mathbb{R}^{q\times s_l}$, for arbitrary temporal length q.
Agg_{r_k,s_l}	Maps $\{\mathbf{x}_\tau\}_{\tau=t+1}^{t+F}$ to its scale-aligned target in $\mathbb{R}^{r_k\times s_l}$.
$\hat{\mathbf{f}}_{[t]}^{(r_k,s_l)}$	Explicit forecast state in $\mathbb{R}^{r_k\times s_l}$ at prediction origin t.

Table 4: Refinement and selection notation from Sections 4.2–4.3.

Symbol	Meaning
$(r_k,s_l)\to(r_i,s_j)$	Transition from the current scale pair to a selected finer pair.
$\mathbf{J}^{(k,l)\to(i,j)}$	Expanded current forecast in $\mathbb{R}^{r_i\times s_j}$.
$\mathcal{A}_{(i,j)\to(k,l)}$	Joint aggregation $\mathcal{P}_{j\to l}\circ\mathcal{T}_{i\to k}$ back to the current pair.
$\mathbf{\Delta}^{(r_i,s_j)}$	Forecast-network correction in $\mathbb{R}^{r_i\times s_j}$.
$\mathbf{P}^{(k,l)\to(i,j)}$	Aggregated correction in $\mathbb{R}^{r_k\times s_l}$; distinct from the spatial operator $\mathcal{P}$.
$\mathbf{L}^{(k,l)\to(i,j)}$	Expansion of $\mathbf{P}^{(k,l)\to(i,j)}$ to $\mathbb{R}^{r_i\times s_j}$.
$\mathbf{D}^{(k,l)\to(i,j)}$	Remaining correction $\mathbf{\Delta}^{(r_i,s_j)}-\mathbf{L}^{(k,l)\to(i,j)}$, whose joint aggregation is zero.
$\lambda_{k,l\to i,j}\in(0,1)$	Weight on the expanded aggregated correction.
$\widetilde{\mathbf{f}}_{[t]}^{(r_i,s_j)\mid(r_k,s_l)}$	Candidate forecast $\mathbf{J}+\lambda\mathbf{L}+\mathbf{D}$ at the finer pair, with the transition superscripts shown above.
$\mathcal{C}_{k,l}, g_{k,l}(r_i,s_j)$	Admissible subsequent scale pairs and target-informed immediate refinement gain.
$\mathcal{Z}_{k,l}, \Phi_\phi$	Observable forecasting state and the general local selector with parameters ϕ.
$\Omega\in\mathfrak{O}, J$	Legal complete scale-selection sequence and its length; the final pair is (F,N).

A.2 Multi-Scale Spatio-Temporal Forecaster

A.2.1 Temporal Module

Temporal Encoding. We model the historical sequence from two complementary temporal views. The first view groups consecutive observations to capture local temporal dependencies, while the second view samples observations at fixed intervals to capture dependencies across different time steps.

We partition $\{(\mathbf{C}_\tau,\mathbf{x}_\tau)\}_{\tau=t-H+1}^{t}$ into consecutive patches of length $p\mid H$, giving $M=H/p$ patches. Patch m contains times $t-H+(m-1)p+q$ for $q=1,\dots,p$. At each node, contextual features and the observed target are concatenated along the feature dimension.

The strided view samples the same observations at stride d. Let $H^+=d\lceil H/d\rceil$, $Q=H^+/d$, and $\delta=H^+-H$. Prepending δ copies of the latest observation gives the observation-time index

$$\iota(h)=\begin{cases}t, & 1\le h\le\delta,\\ t-H+h-\delta, & \delta<h\le H^+.\end{cases}\tag{16}$$

Strided sequence a uses times $\iota(a+(j-1)d)$ for $j=1,\dots,Q$ and $a=1,\dots,d$.

Let D_e denote the hidden feature width. The learnable mappings $f^{\mathrm{patch}}:\mathbb{R}^{p(D+1)}\to\mathbb{R}^{D_e}$ and $f^{\mathrm{str}}:\mathbb{R}^{Q(D+1)}\to\mathbb{R}^{D_e}$ encode the two views, where D is the contextual feature dimension in Section 3. The encodings $\mathbf{E}^{\mathrm{patch}}\in\mathbb{R}^{M\times N\times D_e}$ and $\mathbf{E}^{\mathrm{str}}\in\mathbb{R}^{d\times N\times D_e}$ are

$$\mathbf{E}^{\mathrm{patch}}_{m,n,:}=f^{\mathrm{patch}}\left(\mathrm{vec}\left(\left[C_{t-H+(m-1)p+q,n,:}\|x_{t-H+(m-1)p+q,n}\right]_{q=1}^{p}\right)\right),\qquad m=1,\dots,M,\tag{17}$$

and

$$\mathbf{E}^{\mathrm{str}}_{a,n,:} = f^{\mathrm{str}}\left(\mathrm{vec}\left(\left[C_{\iota(a+(j-1)d),n,:}\|x_{\iota(a+(j-1)d),n}\right]_{j=1}^{Q}\right)\right), \qquad a = 1,\ldots,d, \tag{18}$$

where $\|$ concatenates features, the brackets stack observations in temporal order, and vec flattens each resulting matrix.

Time-of-day and day-of-week are calendar features in $\mathbf{C}_\tau$ when available. We encode them in both temporal views. Let $\mathcal{W}^{\mathrm{tod}} = \{\mathbf{w}^{\mathrm{tod}}_k\}_{k=1}^{K_{\mathrm{tod}}}$ and $\mathcal{W}^{\mathrm{dow}} = \{\mathbf{w}^{\mathrm{dow}}_k\}_{k=1}^{K_{\mathrm{dow}}}$ denote two trainable codebooks. Each entry represents one time-of-day or day-of-week category. For a time-of-day attribute u and a day-of-week attribute w, let $q^{\mathrm{tod}}(u)$ and $q^{\mathrm{dow}}(w)$ denote their category indices. Their embeddings are $\mathbf{e}^{\mathrm{tod}}(u) = \mathbf{w}^{\mathrm{tod}}_{q^{\mathrm{tod}}(u)}$ and $\mathbf{e}^{\mathrm{dow}}(w) = \mathbf{w}^{\mathrm{dow}}_{q^{\mathrm{dow}}(w)}$. We represent the temporal information of an observation by $\boldsymbol{\tau}(u,w) = \mathbf{e}^{\mathrm{tod}}(u)\|\mathbf{e}^{\mathrm{dow}}(w)$, where $\|$ denotes concatenation along the feature dimension.

For each patch, let $\boldsymbol{\tau}^{\mathrm{s,patch}}_{m,n}$ and $\boldsymbol{\tau}^{\mathrm{e,patch}}_{m,n}$ denote the temporal representations of its first and last observations at node n. We define $\boldsymbol{\tau}^{\mathrm{s,str}}_{a,n}$ and $\boldsymbol{\tau}^{\mathrm{e,str}}_{a,n}$ in the same way for the a-th strided sequence. We also use a shared node embedding layer $\mathcal{E}^{\mathrm{node}} : \{1,\ldots,N\} \to \mathbb{R}^{D_n}$ and write $\mathbf{s}_n = \mathcal{E}^{\mathrm{node}}(n)$.

The feature transformations in the two views use residual MLPs. For a hidden feature vector $\mathbf{z}$, a residual MLP is

$$\rho(\mathbf{z}) = \mathrm{LN}\left(\mathbf{z} + \mathbf{W}_2\sigma\left(\mathbf{W}_1\mathbf{z}\right)\right), \tag{19}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable transformations, $\sigma(\cdot)$ is a nonlinear activation, and $\mathrm{LN}(\cdot)$ denotes layer normalization. The resulting temporal tokens are

$$\mathbf{Z}^{\mathrm{patch}}_{m,n,:} = \rho^{\mathrm{patch,temp}}\left(\boldsymbol{\tau}^{\mathrm{s,patch}}_{m,n}\|\rho^{\mathrm{patch,node}}\left(\mathbf{E}^{\mathrm{patch}}_{m,n,:}\|\mathbf{s}_n\right)\|\boldsymbol{\tau}^{\mathrm{e,patch}}_{m,n}\right), \tag{20}$$

and

$$\mathbf{Z}^{\mathrm{str}}_{a,n,:} = \rho^{\mathrm{str,temp}}\left(\boldsymbol{\tau}^{\mathrm{s,str}}_{a,n}\|\rho^{\mathrm{str,node}}\left(\mathbf{E}^{\mathrm{str}}_{a,n,:}\|\mathbf{s}_n\right)\|\boldsymbol{\tau}^{\mathrm{e,str}}_{a,n}\right), \tag{21}$$

where $\mathbf{Z}^{\mathrm{patch}} \in \mathbb{R}^{M\times N\times D_e}$ and $\mathbf{Z}^{\mathrm{str}} \in \mathbb{R}^{d\times N\times D_e}$.

Temporal Aggregation. We next aggregate the temporal tokens for each target position. Throughout the module description, (r_k, s_l) denotes the selected pair. At this pair, let $\mathbf{p}^{(k)}_\ell$ denote the positional representation of target position ℓ, let $u^{(k)}_{\ell,n}$ denote its time-of-day attribute at node n, and let $w_{t,n}$ denote the day-of-week attribute of the latest historical observation. Time-of-day is extrapolated to the selected temporal interval midpoint, while $w_{t,n}$ is retained from the latest observation. The target query is

$$\mathbf{q}^{(k)}_{\ell,n} = \mathbf{W}_p\mathbf{p}^{(k)}_\ell + \mathbf{W}_t\boldsymbol{\tau}\left(u^{(k)}_{\ell,n}, w_{t,n}\right) + \mathbf{W}_s\mathbf{s}_n, \tag{22}$$

where $\mathbf{W}_p$, $\mathbf{W}_t$, and $\mathbf{W}_s$ are learnable projections. Collecting the queries gives $\mathbf{Q}^{(k)} \in \mathbb{R}^{r_k\times N\times D_e}$.

We use target-aware temporal aggregation to summarize the two views separately. The patch and strided contexts are $\mathbf{C}^{\mathrm{patch}}_k = \mathrm{Attn}(\mathbf{Q}^{(k)}, \mathbf{Z}^{\mathrm{patch}})$ and $\mathbf{C}^{\mathrm{str}}_k = \mathrm{Attn}(\mathbf{Q}^{(k)}, \mathbf{Z}^{\mathrm{str}})$, where $\mathrm{Attn}(\cdot,\cdot)$ denotes node-wise scaled dot-product attention over the temporal-token dimension. Both contexts lie in $\mathbb{R}^{r_k\times N\times D_e}$, as do the previous-forecast and latest-observation contexts used below.

Scale-Conditioned Temporal Refinement. The condition $\mathbf{c}_{k,l}$ encodes the preceding and selected scale pairs. The two temporal views and the preceding forecast are fused as

$$\mathbf{H}^{(0)}_{k,l} = \mathbf{Q}^{(k)} + \mathbf{H}^{\mathrm{prev}}_{k,l} + \lambda^{\mathrm{patch}}_{k,l}\mathbf{C}^{\mathrm{patch}}_k + \lambda^{\mathrm{str}}_{k,l}\mathbf{C}^{\mathrm{str}}_k + \lambda^{\mathrm{per}}_{k,l}\mathbf{C}^{\mathrm{per}}_k, \tag{23}$$

where $\mathbf{H}^{\mathrm{prev}}_{k,l}$ encodes the expanded preceding forecast at the corresponding spatial unit, $\mathbf{C}^{\mathrm{per}}_k$ is derived from the latest historical observation, and $\lambda^{\mathrm{patch}}_{k,l}$, $\lambda^{\mathrm{str}}_{k,l}$, and $\lambda^{\mathrm{per}}_{k,l}$ are determined by $\mathbf{c}_{k,l}$.

The forecast-scale condition further modulates the fused representation through feature-wise scale and shift parameters:

$$\overline{\mathbf{H}}_{k,l} = \mathrm{LN}\left(\boldsymbol{\gamma}_{k,l}\odot\mathbf{H}^{(0)}_{k,l} + \boldsymbol{\beta}_{k,l}\right), \tag{24}$$

where $\boldsymbol{\gamma}_{k,l}$ and $\boldsymbol{\beta}_{k,l}$ are generated from $\mathbf{c}_{k,l}$.

We then model local interactions between neighboring future positions. Let $\mathcal{K}$ denote a shared local temporal operator applied independently at each node. For target position ℓ,

$$\left[\mathcal{K}\left(\overline{\mathbf{H}}_{k,l}\right)\right]_{\ell,n,:} = \sum_{j\in\mathcal{K}_\ell^{\text{temp}}} \mathbf{a}_{\ell-j} \odot \left[\overline{\mathbf{H}}_{k,l}\right]_{j,n,:}, \tag{25}$$

where $\mathcal{K}_\ell^{\text{temp}}$ denotes the neighboring positions in $\{1,\dots,r_k\}$ and $\mathbf{a}_{\ell-j}$ is the learnable temporal kernel for relative offset $\ell-j$. The temporal representation is then refined as

$$\mathbf{H}_{k,l}^{\text{temp}} = \rho^{\text{temp}}\left(\overline{\mathbf{H}}_{k,l} + \mathcal{K}\left(\overline{\mathbf{H}}_{k,l}\right)\right) \in \mathbb{R}^{r_k\times N\times D_e}, \tag{26}$$

where ρ^{temp} follows the residual transformation defined above.

A.2.2 Spatial Module

Nested Spatial Hierarchy. We learn two global node embedding matrices $\mathbf{E}^{\text{src}}, \mathbf{E}^{\text{dst}} \in \mathbb{R}^{N\times d_a}$ to describe source and destination roles in spatial relations. At the finest scale $s_L = N$, let $\mathbf{e}_{L,n}^{\text{src}}$ and $\mathbf{e}_{L,n}^{\text{dst}}$ denote the corresponding embeddings of node n. For spatial units u and v at scale s_l, we first normalize their embeddings and compute

$$b_{l,uv} = \frac{\left\langle \widehat{\mathbf{e}}_{l,u}^{\text{src}}, \widehat{\mathbf{e}}_{l,v}^{\text{dst}} \right\rangle}{\sqrt{d_a}}, \qquad q_{l,uv} = \frac{\exp(b_{l,uv})}{\sum_{v'=1}^{s_l}\exp(b_{l,uv'})}, \tag{27}$$

where $\widehat{\mathbf{e}} = \mathbf{e}/\|\mathbf{e}\|_2$. The symmetric affinity used to construct the spatial hierarchy is

$$A_{l,uv}^{\text{hier}} = \frac{q_{l,uv} + q_{l,vu}}{2}. \tag{28}$$

The hierarchy instantiates the surjections $\psi_l : \mathcal{N}^{(s_{l+1})} \to \mathcal{N}^{(s_l)}$ from Section 4.1. The initial centers are selected to cover units with low affinity to the centers already chosen. The first center has the largest total affinity, and each following center favors a unit with small maximum affinity to the selected centers.

After selecting s_l centers, the s_{l+1} child units are assigned by their affinities to these centers under balanced capacities. The capacity of the u-th coarse unit is

$$\text{cap}_{l,u} = \left\lfloor \frac{s_{l+1}}{s_l} \right\rfloor + \mathbb{1}\left[u \le s_{l+1} \bmod s_l\right]. \tag{29}$$

Each child unit prefers centers with larger affinity and is assigned to the highest-ranked center with remaining capacity. The center of each resulting group is then refined by selecting the member with the largest total affinity to the other members, followed by balanced reassignment.

The hierarchy remains nested because each scale is formed from the units of the preceding finer scale. Let $\mu_{l,u}$ denote the number of original nodes represented by unit u at scale s_l, with $\mu_{L,n} = 1$. After applying ψ_l, its node count and source embedding are updated as

$$\mu_{l,u} = \sum_{v\in\Gamma_u^{(l+1|l)}} \mu_{l+1,v}, \qquad \mathbf{e}_{l,u}^{\text{src}} = \frac{\sum_{v\in\Gamma_u^{(l+1|l)}} \mu_{l+1,v}\mathbf{e}_{l+1,v}^{\text{src}}}{\mu_{l,u}}, \tag{30}$$

with the destination embedding $\mathbf{e}_{l,u}^{\text{dst}}$ computed in the same way. The affinity matrix is then recomputed at scale s_l before constructing the next coarser level.

For later spatial propagation, we compose the adjacent assignments into an original-node assignment. We set $\pi_L(n) = n$ and define

$$\pi_l(n) = \psi_l\left(\pi_{l+1}(n)\right), \qquad l = L-1,\dots,1. \tag{31}$$

The original-node descendants from Section 4.2 satisfy

$$\Gamma_u^{(L|l)} = \{n \in \{1,\dots,N\} \mid \pi_l(n) = u\}. \tag{32}$$

Scale-Specific Spatial Propagation. For an active spatial scale $s_l < N$, we pool the current node-level representation within each spatial unit. For target position ℓ, the pooled representation is

$$\left[\widetilde{\mathbf{H}}_{k,l}\right]_{\ell,u,:} = \frac{1}{|\Gamma_u^{(L|l)}|} \sum_{n \in \Gamma_u^{(L|l)}} \left[\mathbf{H}_{k,l}^{\text{in}}\right]_{\ell,n,:}, \tag{33}$$

where $\mathbf{H}_{k,l}^{\text{in}} \in \mathbb{R}^{r_k \times N \times D_e}$ denotes the hidden features entering spatial interaction, initialized by $\mathbf{H}_{k,l}^{\text{temp}}$. Pooling is applied independently at each future position and gives $\widetilde{\mathbf{H}}_{k,l} \in \mathbb{R}^{r_k \times s_l \times D_e}$.

The graph used for spatial propagation is constructed from the current adaptive node embeddings. At a coarse scale, the source and destination embeddings of a spatial unit are obtained by averaging the embeddings of its member nodes:

$$\mathbf{g}_{l,u}^{\text{src}} = \frac{1}{|\Gamma_u^{(L|l)}|} \sum_{n \in \Gamma_u^{(L|l)}} \mathbf{E}_{n,:}^{\text{src}}, \qquad \mathbf{g}_{l,u}^{\text{dst}} = \frac{1}{|\Gamma_u^{(L|l)}|} \sum_{n \in \Gamma_u^{(L|l)}} \mathbf{E}_{n,:}^{\text{dst}}. \tag{34}$$

Their normalized similarity is

$$a_{l,uv} = \frac{\left\langle \widehat{\mathbf{g}}_{l,u}^{\text{src}}, \widehat{\mathbf{g}}_{l,v}^{\text{dst}} \right\rangle}{\sqrt{d_a}}. \tag{35}$$

Let $\mathcal{K}_{l,u}^{\text{sp}}$ contain the K_l^{sp} units with the largest values of $a_{l,uv}$. The directed propagation weights are

$$w_{l,uv} = \frac{\exp(a_{l,uv})}{\sum_{v' \in \mathcal{K}_{l,u}^{\text{sp}}} \exp(a_{l,uv'})}, \qquad v \in \mathcal{K}_{l,u}^{\text{sp}}. \tag{36}$$

We perform spatial propagation in a lower-dimensional feature space. Let $P_{\text{in}} : \mathbb{R}^{D_e} \to \mathbb{R}^{d_g}$ and $P_{\text{out}} : \mathbb{R}^{d_g} \to \mathbb{R}^{D_e}$ denote shared feature mappings. The projected features and their weighted spatial average are

$$[\mathbf{V}_{k,l}]_{\ell,u,:} = P_{\text{in}}\left(\left[\widetilde{\mathbf{H}}_{k,l}\right]_{\ell,u,:}\right), \qquad [\mathbf{U}_{k,l}]_{\ell,u,:} = \sum_{v \in \mathcal{K}_{l,u}^{\text{sp}}} w_{l,uv} \left[\mathbf{V}_{k,l}\right]_{\ell,v,:}. \tag{37}$$

The hidden-feature residual for unit u is then

$$\left[\mathbf{R}_{k,l}^{\text{unit}}\right]_{\ell,u,:} = P_{\text{out}}\left([\mathbf{U}_{k,l}]_{\ell,u,:} - [\mathbf{V}_{k,l}]_{\ell,u,:}\right). \tag{38}$$

Spatial Feature Update. The hidden-feature residual is mapped back to the original nodes using the nested assignment:

$$[\mathbf{R}_{k,l}]_{\ell,n,:} = \left[\mathbf{R}_{k,l}^{\text{unit}}\right]_{\ell,\pi_l(n),:}. \tag{39}$$

Thus, all nodes within the same spatial unit receive the hidden-feature residual from spatial propagation at that scale.

Let $g_{k,l}^{\text{sp}} = \sigma(\mathbf{w}_g^\top \mathbf{c}_{k,l} + b_g)$ denote a forecast-scale gate derived from the scale condition $\mathbf{c}_{k,l}$, and let α_l denote the learned strength of spatial scale s_l. The node-level representation is refined as

$$\mathbf{H}_{k,l}^{\text{sp}} = \text{LN}\left(\mathbf{H}_{k,l}^{\text{in}} + g_{k,l}^{\text{sp}} \alpha_l \mathbf{R}_{k,l}\right), \qquad s_l < N. \tag{40}$$

Within one forecasting stage, the hidden-feature update may be applied at successive active spatial partitions. Forecast states are constructed by the scale-specific prediction step below.

At the finest scale $s_L = N$, the assignment becomes $\pi_L(n) = n$. Pooling and broadcast are no longer required. We directly construct the sparse adaptive graph from $\mathbf{E}^{\text{src}}$ and $\mathbf{E}^{\text{dst}}$ and apply the same low-rank propagation to the original nodes:

$$[\mathbf{V}_{k,L}]_{\ell,n,:} = P_{\text{in}}\left(\left[\mathbf{H}_{k,L}^{\text{in}}\right]_{\ell,n,:}\right), \qquad [\mathbf{U}_{k,L}]_{\ell,n,:} = \sum_{v \in \mathcal{K}_{L,n}^{\text{sp}}} w_{L,nv} \left[\mathbf{V}_{k,L}\right]_{\ell,v,:}, \tag{41}$$

$$[\mathbf{R}_{k,L}]_{\ell,n,:} = P_{\text{out}}\left([\mathbf{U}_{k,L}]_{\ell,n,:} - [\mathbf{V}_{k,L}]_{\ell,n,:}\right). \tag{42}$$

The spatial output at forecasting scale r_k is

$$\mathbf{H}_{k,L}^{\text{sp}} = \text{LN}\left(\mathbf{H}_{k,L}^{\text{in}} + g_{k,L}^{\text{sp}} \mathbf{R}_{k,L}\right) \in \mathbb{R}^{r_k \times N \times D_e}. \tag{43}$$

A.2.3 PROGRESSIVE MULTI-SCALE FORECASTING

A shared position-wise head h_θ maps the spatial features to scalar corrections. The recursive spatial aggregation in Section 4.1 gives the correction at the selected pair:

$$\mathbf{\Delta}^{(r_k,s_l)} = \mathcal{P}_{L\to l}\Big(h_\theta(\mathbf{H}^{\mathrm{sp}}_{k,l})\Big) \in \mathbb{R}^{r_k\times s_l}. \tag{44}$$

The hidden-feature residuals $\mathbf{R}$ above supply spatial evidence; $\mathbf{\Delta}^{(r_k,s_l)}$ is the additive forecast correction used by Section 4.2. The head is shared across scale pairs.

Forecast Generation at Each Selected Pair. Consider a transition $(r_k, s_l) \to (r_i, s_j)$ in Ω. The forecast anchor is the expanded forecast $\mathbf{J}^{(k,l)\to(i,j)}$ from Eq. (8). Its entries are

$$\mathbf{J}^{(k,l)\to(i,j)}_{a,v} = \Big[\hat{\mathbf{f}}^{(r_k,s_l)}_{[t]}\Big]_{m,u}, \qquad a \in \mathcal{B}_{k\to i,m}, \quad v \in \Gamma^{(j|l)}_u. \tag{45}$$

Encoding this expansion provides the preceding-forecast context $\mathbf{H}^{\mathrm{prev}}_{i,j}$ in the temporal module. Each original node uses the entry associated with its spatial unit under π_j.

The correction is decomposed using the joint aggregation from Section 4.2:

$$\mathbf{P}^{(k,l)\to(i,j)} = \mathcal{A}_{(i,j)\to(k,l)}\Big(\mathbf{\Delta}^{(r_i,s_j)}\Big) \in \mathbb{R}^{r_k\times s_l}. \tag{46}$$

Expanding this aggregated correction gives

$$\mathbf{L}^{(k,l)\to(i,j)}_{a,v} = \mathbf{P}^{(k,l)\to(i,j)}_{m,u}, \qquad a \in \mathcal{B}_{k\to i,m}, \quad v \in \Gamma^{(j|l)}_u, \tag{47}$$

and the remaining detail is

$$\mathbf{D}^{(k,l)\to(i,j)} = \mathbf{\Delta}^{(r_i,s_j)} - \mathbf{L}^{(k,l)\to(i,j)}. \tag{48}$$

Its joint aggregation is zero. The selected candidate becomes

$$\hat{\mathbf{f}}^{(r_i,s_j)}_{[t]} = \mathbf{J}^{(k,l)\to(i,j)} + \lambda_{k,l\to i,j}\mathbf{L}^{(k,l)\to(i,j)} + \mathbf{D}^{(k,l)\to(i,j)} \in \mathbb{R}^{r_i\times s_j}, \tag{49}$$

where $\lambda_{k,l\to i,j} \in (0, 1)$ is determined by the transition's scale condition.

At the first pair (r_{k_1}, s_{l_1}) of Ω, the latest observed target is repeated over the future horizon and aggregated to that pair. The first forecast adds its correction directly:

$$\hat{\mathbf{f}}^{(r_{k_1},s_{l_1})}_{[t]} = \mathrm{Agg}_{r_{k_1},s_{l_1}}\big(\{\mathbf{x}_t\}^{t+F}_{\tau=t+1}\big) + \mathbf{\Delta}^{(r_{k_1},s_{l_1})}. \tag{50}$$

Here every entry of the sequence passed to Agg is the same observed vector $\mathbf{x}_t$; no future target is used for initialization.

Progressive Temporal–Spatial Refinement. Successive pairs in Ω refine the temporal partition, the spatial partition, or both. Each explicit forecast conditions the next refinement. Temporal and spatial hidden features are recomputed within each stage. A complete sequence ends at (F, N), where

$$\Big[\hat{\mathbf{f}}^{(F,N)}_{[t]}\Big]_{m,n} = [\hat{\mathbf{x}}_{t+m}]_n, \qquad m = 1,\dots,F, \quad n \in \mathcal{N}. \tag{51}$$

A.2.4 TRAINING OBJECTIVE

At a visited pair (r_k, s_l), we instantiate the scale-specific loss in Eq. (7) with masked MAE. The stage loss is

$$\mathcal{L}_{k,l} = \mathrm{MAE}_{\mathrm{mask}}\Big(\hat{\mathbf{f}}^{(r_k,s_l)}_{[t]}, \mathrm{Agg}_{r_k,s_l}\big(\{\mathbf{x}_\tau\}^{t+F}_{\tau=t+1}\big)\Big), \tag{52}$$

where $\mathrm{MAE}_{\mathrm{mask}}$ averages absolute error over valid entries of the scale-aligned target. Both arguments lie in $\mathbb{R}^{r_k\times s_l}$.

For a fixed scale-selection sequence $\Omega = ((r_{k_1}, s_{l_1}), \dots, (r_{k_J}, s_{l_J})) \in \mathfrak{O}$, the objective combines all visited pairs:

$$\mathcal{L}_{\mathrm{fixed}} = \sum_{p=1}^{J} \omega_p \mathcal{L}_{k_p,l_p}, \tag{53}$$

where $\omega_p \geq 0$ is the supervision weight at position p, with greater weight on the terminal forecast.

For a sampled legal sequence $\Omega \in \mathfrak{O}$, training combines terminal and intermediate losses:

$$\mathcal{L}_\Omega = \mathcal{L}_{k_J,l_J} + \frac{\eta}{J-1}\sum_{p=1}^{J-1}\mathcal{L}_{k_p,l_p}, \qquad J > 1. \tag{54}$$

For $J = 1$, $\mathcal{L}_\Omega = \mathcal{L}_{k_1,l_1}$. The weight $\eta \geq 0$ controls intermediate supervision. Training over scale selections alternates between the fixed sequence and sampled legal sequences, sharing the temporal, spatial, and prediction modules.

A.2.5 Complete-Sequence Selector

The selector in Section 5.2 evaluates a finite subset $\mathfrak{X} \subseteq \mathfrak{O}$ of the legal sequences defined in Section 4.3. Let $\widehat{\ell}_\varphi$ denote its terminal-loss predictor, with parameters φ. It is trained using historical observations and the terminal losses produced by the shared forecaster. Selection is

$$\widehat{\Omega} = \arg\min_{\Omega\in\mathfrak{X}} \widehat{\ell}_\varphi\big(\{(\mathbf{C}_\tau, \mathbf{x}_\tau)\}_{\tau=t-H+1}^{t}, \Omega\big)\,. \tag{55}$$

For a feasible budget b and profiled cost $c(\Omega)$, minimization is restricted to the nonempty subset $\{\Omega \in \mathfrak{X} : c(\Omega) \leq b\}$. The selected $\widehat{\Omega}$ is a complete sequence fixed before forecasting. The local selector Φ_ϕ in Section 4.3 retains its distinct current-forecast-conditioned meaning.

A.3 Training-procedure comparison

Table 5: Fixed-sequence test MAE ($H = F = 12$) after the two training procedures. Subscripts give standard deviations.

Dataset	Fixed sequence	Sampled sequences
PEMS04	$18.1369_{\pm.0488}$	$18.1588_{\pm.1041}$
PEMS07	$19.2646_{\pm.0655}$	$19.1952_{\pm.0270}$
PEMS08	$13.9444_{\pm.0618}$	$13.8876_{\pm.0563}$
KnowAir	$13.9228_{\pm.0250}$	$13.9504_{\pm.0352}$

Training across sequences. Table 5 evaluates both training procedures on the same fixed sequence. Training over scale selections lowers mean MAE on PEMS07 and PEMS08 by 0.36% and 0.41%, respectively, while increasing it on PEMS04 and KnowAir by 0.12% and 0.20%. All 12 dataset–seed runs satisfy the validation-set constraint of at most 0.01 MAE degradation; the test-set changes quantify the resulting trade-off separately.

To evaluate performance across scale selections, we execute the same 22 legal sequences on identical test samples for both training procedures. For each seed, we compute the equal-weight mean of their terminal MAEs and the maximum sequence MAE, then summarize these statistics over three seeds. Training over scale selections reduces mean MAE from 21.8343 to 18.1861 on PEMS04, from 25.3879 to 21.5082 on PEMS07, from 18.6462 to 15.7327 on PEMS08, and from 14.4211 to 14.0503 on KnowAir, corresponding to reductions of 16.71%, 15.28%, 15.63%, and 2.57%. Worst-sequence MAE decreases from 39.5092 to 18.4788, from 46.4192 to 27.8700, from 41.4659 to 26.0933, and from 15.9103 to 14.8891, respectively. These results support improved accuracy across the evaluated sequences, including those poorly served by fixed-sequence training.

The improvement extends across four common normalized-cost intervals, $[0, 0.25]$, $(0.25, 0.50]$, $(0.50, 0.75]$, and $(0.75, 1]$, containing 1, 9, 6, and 6 sequences. On all four datasets, both mean and worst-sequence MAE decrease in every interval when averaged over seeds. Each sequence has identical FLOPs under both training procedures, so these comparisons measure accuracy gains at unchanged sequence cost. The fractions of sequences improved are $90.9 \pm 7.9\%$, $66.7 \pm 57.7\%$, $66.7 \pm 57.7\%$, and $86.4 \pm 0.0\%$ for PEMS04, PEMS07, PEMS08, and KnowAir, respectively. PEMS07 and PEMS08 show substantial seed variability: one run on each retains the initial checkpoint, and their worst-sequence MAEs after further training are 27.8700 ± 14.0151 and 26.0933 ± 19.5940, respectively.

B PROGRESSIVE REFINEMENT ANALYSIS

B.1 FULL-TEST REFINEMENT GAINS

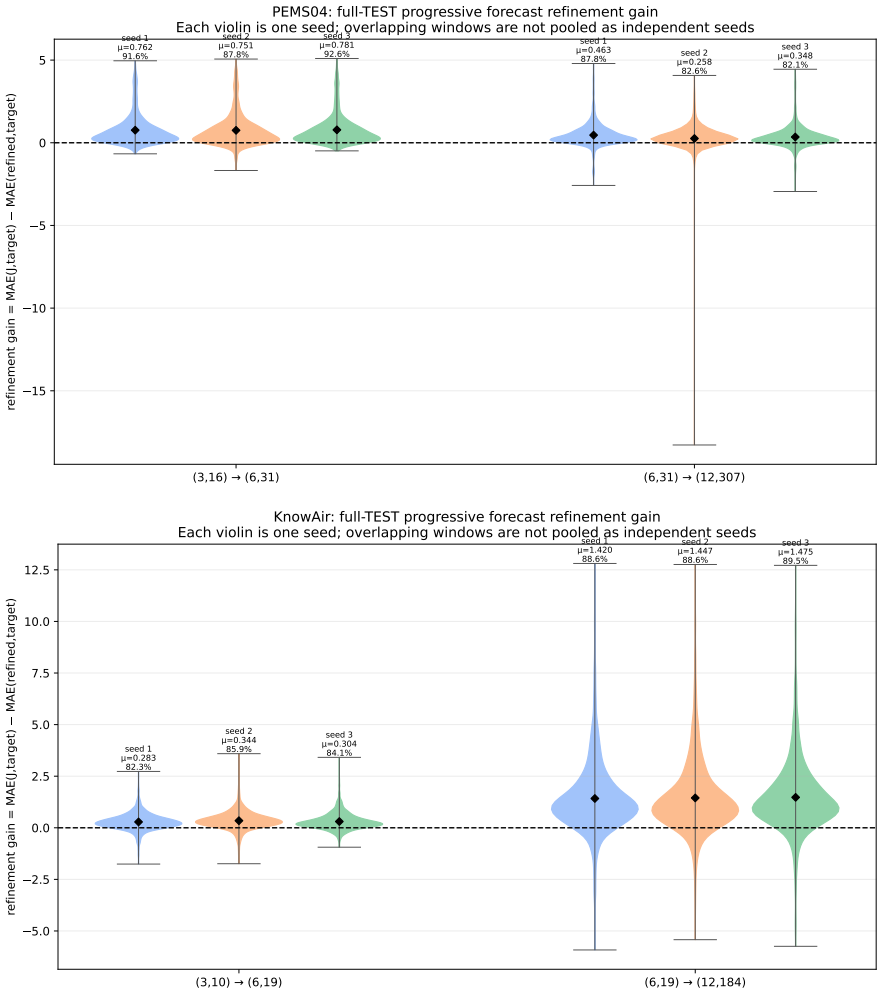


Figure 4: Full-test refinement gains on PEMS04 (top) and KnowAir (bottom). Each violin represents one seed and compares errors at the same selected scale. Diamonds mark means; annotations give the mean and percentage of windows with positive gain. Negative gains are retained.

Table 6: Mean window-level gain and percentage of improved windows. Subscripts denote sample standard deviations across seeds 1–3. Steps 1 and 2 are the two successive refinements in each fixed three-pair sequence.

	Step 1		Step 2	
Dataset	Gain	Improved (%)	Gain	Improved (%)
PEMS04	$.7648_{\pm.0151}$	$90.66_{\pm2.55}$	$.3563_{\pm.1024}$	$84.18_{\pm3.17}$
PEMS07	$1.2104_{\pm.0347}$	$96.22_{\pm1.36}$	$.6141_{\pm.0075}$	$91.94_{\pm.77}$
PEMS08	$.8000_{\pm.0324}$	$91.10_{\pm1.15}$	$.6583_{\pm.0345}$	$93.59_{\pm1.59}$
KnowAir	$.3105_{\pm.0311}$	$84.11_{\pm1.82}$	$1.4472_{\pm.0276}$	$88.88_{\pm.50}$

B.2 EVALUATION AND SAMPLE SELECTION

The evaluations use frozen checkpoints with $H = F = 12$ and seeds 1–3, without further training or checkpoint selection. Coarse spatial views are obtained through the frozen nested hierarchy, using

the selected spatial partition to align forecasts and targets. Only the three selected forecast states are shown. Internal spatial propagation does not define an additional forecast state.

For each window, MAE is computed over valid aligned elements and then averaged equally over test windows. Table 7 reports the resulting means and sample standard deviations across seeds. This analysis uses window-averaged MAE; its terminal MAE is not substituted for the main benchmark MAE. For example, PEMS04 terminal MAE is 18.1752 ± 0.1043 under equal-window averaging and 18.2836 ± 0.1053 under valid-element weighting. Refinement gains always compare errors at the same selected scale.

Table 7: Full-test forecast errors at each selected scale. Errors are window-averaged MAEs; subscripts denote sample standard deviations across three seeds. The first forecast initializes the sequence and has no preceding-forecast gain.

Dataset	Scale	Expanded MAE	Refined MAE	Gain	Improved (%)
PEMS04	$(3, 16)$	—	$5.3155_{\pm.2686}$	—	—
	$(6, 31)$	$7.8362_{\pm.1995}$	$7.0714_{\pm.2041}$	$.7648_{\pm.0151}$	$90.66_{\pm2.55}$
	$(12, 307)$	$18.5315_{\pm.0635}$	$18.1752_{\pm.1043}$	$.3563_{\pm.1024}$	$84.18_{\pm3.17}$
KnowAir	$(3, 10)$	—	$7.9324_{\pm.1056}$	—	—
	$(6, 19)$	$9.4890_{\pm.0374}$	$9.1785_{\pm.0089}$	$.3105_{\pm.0311}$	$84.11_{\pm1.82}$
	$(12, 184)$	$15.3975_{\pm.0591}$	$13.9503_{\pm.0354}$	$1.4472_{\pm.0276}$	$88.88_{\pm.50}$

All heatmaps use seed 1. The representative window has terminal MAE closest to that seed's test median, with ties resolved by sample index: index 2374 for PEMS04 and 1637 for KnowAir. The negative-gain window minimizes gain across both steps, with ties resolved by sample index and then selected-scale index: index 2069 for PEMS04 and 154 for KnowAir. Their terminal-step gains are -2.5812 and -5.9200, respectively. These rules select examples by numerical criteria rather than visual appearance. Each dataset uses identical value and absolute-error color limits for its representative and negative-gain examples. The supplied exports identify windows by global time indices and calendar codes; no absolute dates or unverified physical units are inferred.

B.3 Recursive correction interventions

Table 8: Frozen-weight recursive correction interventions at $H = F = 12$. Terminal MAE is evaluated with the formal batch-averaged masked metric; subscripts are sample standard deviations over seeds 1–3. All variants share the same trained checkpoint and scale-selection sequence within each run.

Intervention	PEMS04	PEMS07	PEMS08	KnowAir
Full ProST	$\mathbf{18.1588_{\pm.1041}}$	$\mathbf{19.1952_{\pm.0270}}$	$\mathbf{13.8876_{\pm.0563}}$	$\mathbf{13.9504_{\pm.0352}}$
Without parent update	$18.3017_{\pm.0476}$	$19.3688_{\pm.0635}$	$14.0522_{\pm.0390}$	$15.3304_{\pm.0459}$
Without detail	$18.4447_{\pm.1127}$	$19.8098_{\pm.0934}$	$14.6148_{\pm.0871}$	$14.1279_{\pm.0216}$
Undamped correction	$18.1756_{\pm.1110}$	$19.2307_{\pm.0259}$	$13.9539_{\pm.0734}$	$14.0799_{\pm.0023}$

Local gains and terminal accuracy. Removing detail reduces the first-step mean gain to -0.0310, -0.0226, 0.0067, and 0.0226 on PEMS04, PEMS07, PEMS08, and KnowAir, respectively. A larger local gain does not necessarily yield a better terminal forecast: on PEMS04, removing the parent update raises the first-step gain from 0.7648 to 0.7926, yet increases terminal MAE. Because each intervention changes the forecast supplied to the next step, local gains are measured from different preceding states. Appendix B.3 shows the intervention gain distributions and per-horizon errors, together with paired seed-level differences and numerical checks of the executed temporal decomposition.

The intervention exporter decomposes corrections with temporal mean pooling and piecewise-constant expansion, applied along the time dimension. Hierarchy-based spatial aggregation supplies the scale-aligned forecast views. The intervention results therefore test the executed temporal parent/detail decomposition within the forecasting system; the numerical checks below cover these temporal operators.

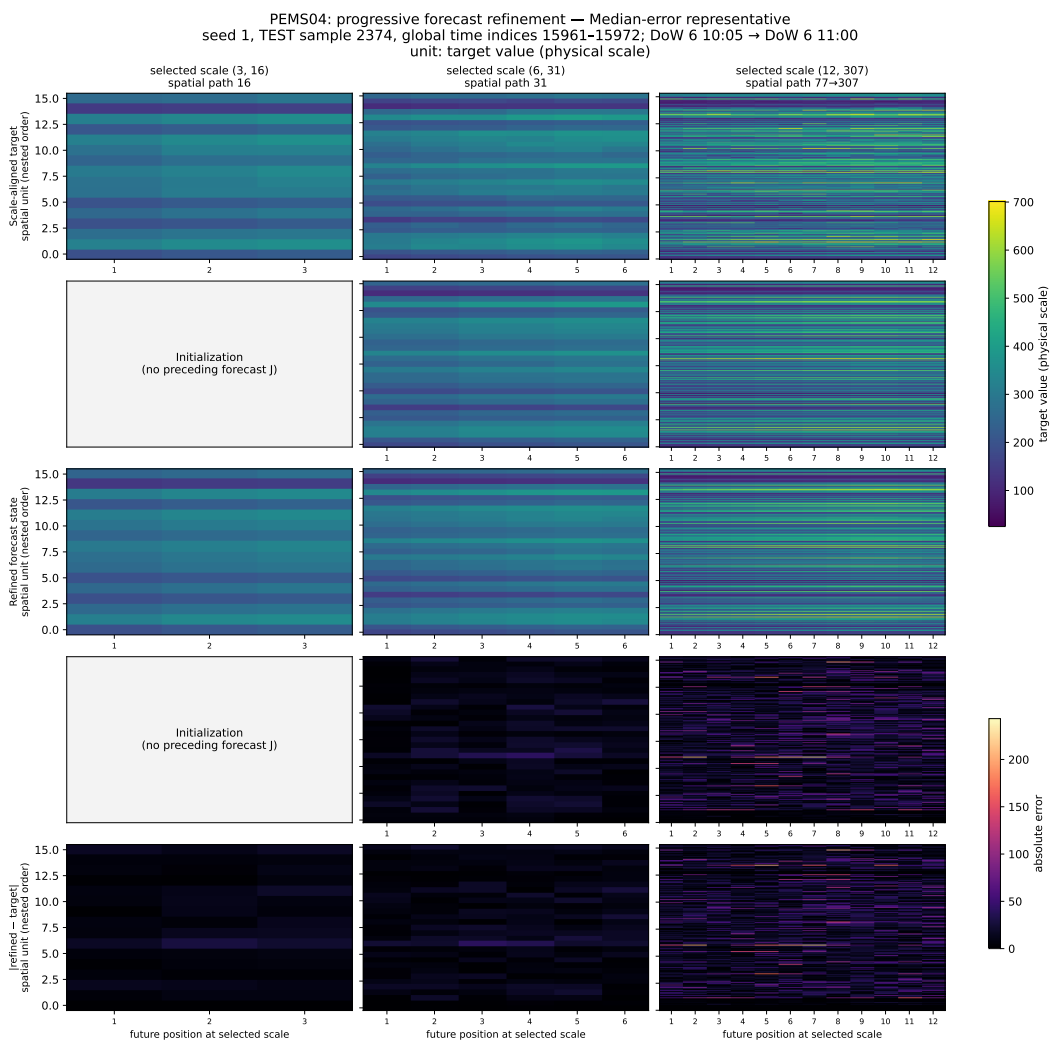


Figure 5: Representative PEMS04 window. Columns correspond to selected temporal–spatial scales. Rows show the aligned target, expanded preceding forecast, refined forecast, and their absolute errors. Coarse views use hierarchical aggregation; the initial scale has no preceding forecast.

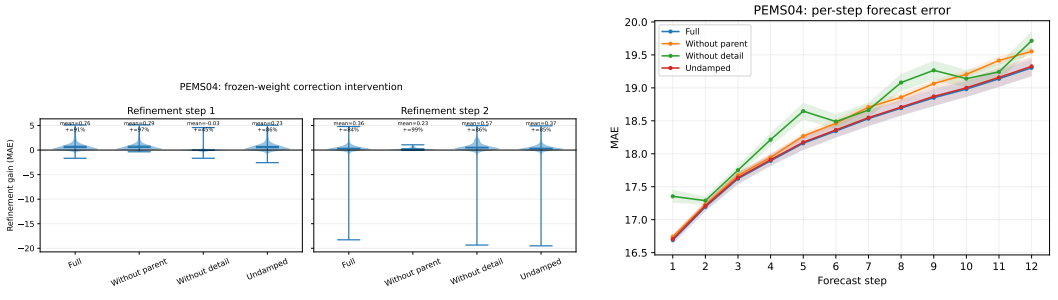


Figure 6: Recursive correction interventions on PEMS04. Left: gain distributions at both refinement steps, including negative gains. Right: MAE at each of the 12 forecast steps. All interventions retain the trained parameters and recursively propagate their own forecast outputs.

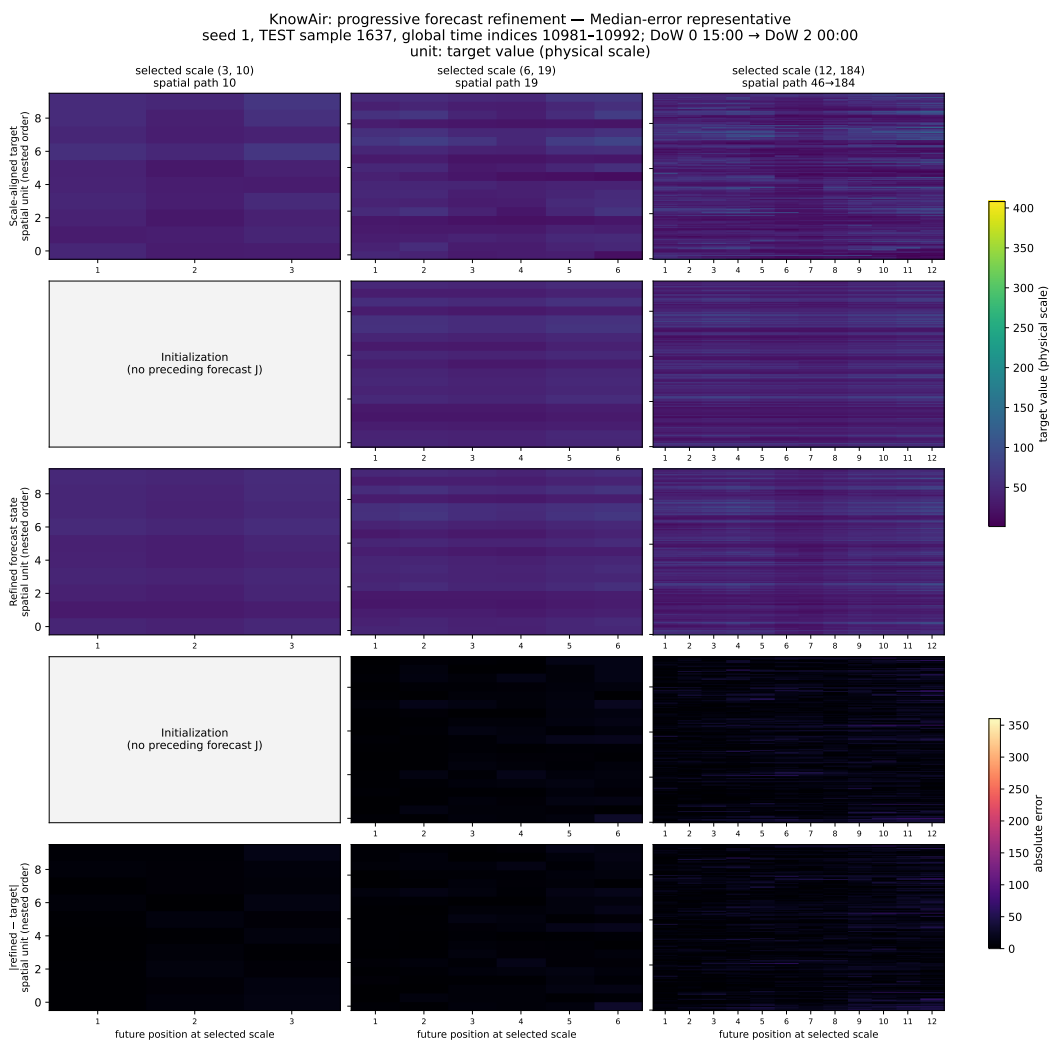


Figure 7: Representative KnowAir window. Columns correspond to selected temporal–spatial scales. Rows show the aligned target, expanded preceding forecast, refined forecast, and their absolute errors. Coarse views use hierarchical aggregation; the initial scale has no preceding forecast.

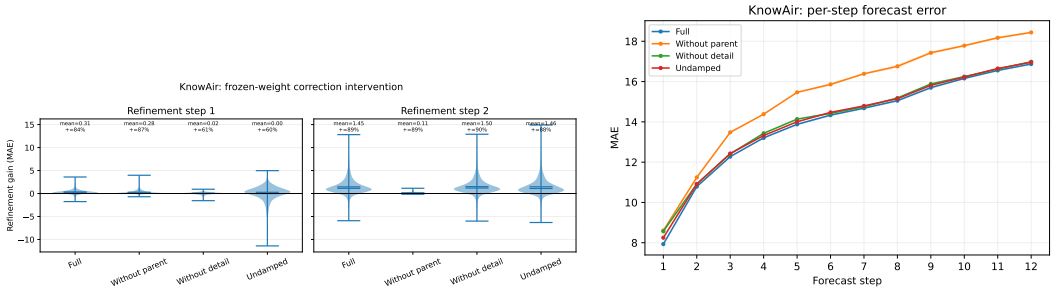


Figure 8: Recursive correction interventions on KnowAir. Left: gain distributions at both refinement steps, including negative gains. Right: MAE at each of the 12 forecast steps. All interventions retain the trained parameters and recursively propagate their own forecast outputs.

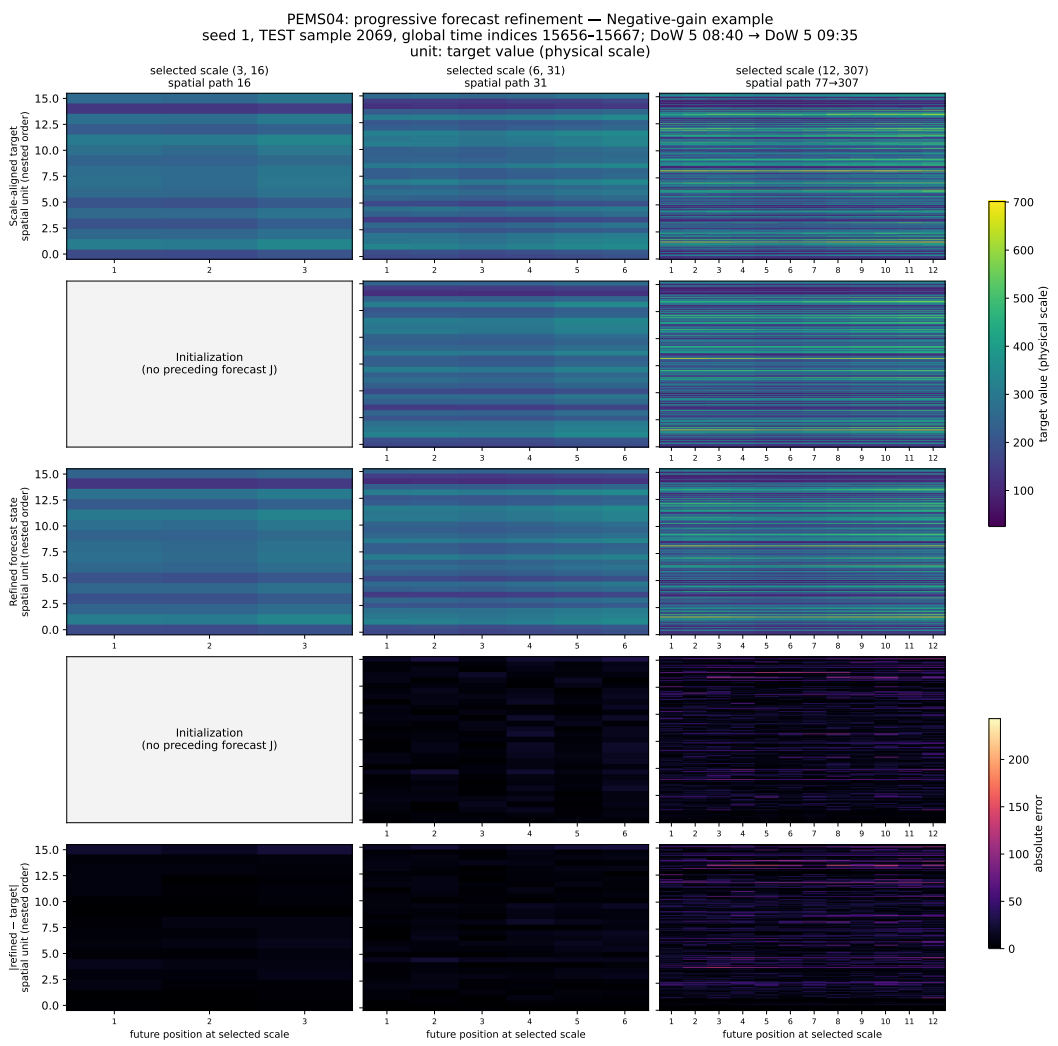


Figure 9: Negative-gain PEMS04 window. Columns correspond to selected temporal–spatial scales. Rows show the aligned target, expanded preceding forecast, refined forecast, and their absolute errors. Coarse views use hierarchical aggregation; the initial scale has no preceding forecast.

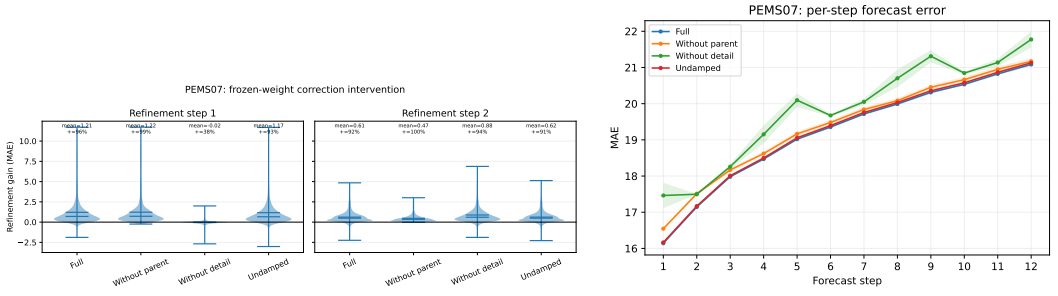


Figure 10: Recursive correction interventions on PEMS07. Left: gain distributions at both refinement steps, including negative gains. Right: MAE at each of the 12 forecast steps. All interventions retain the trained parameters and recursively propagate their own forecast outputs.

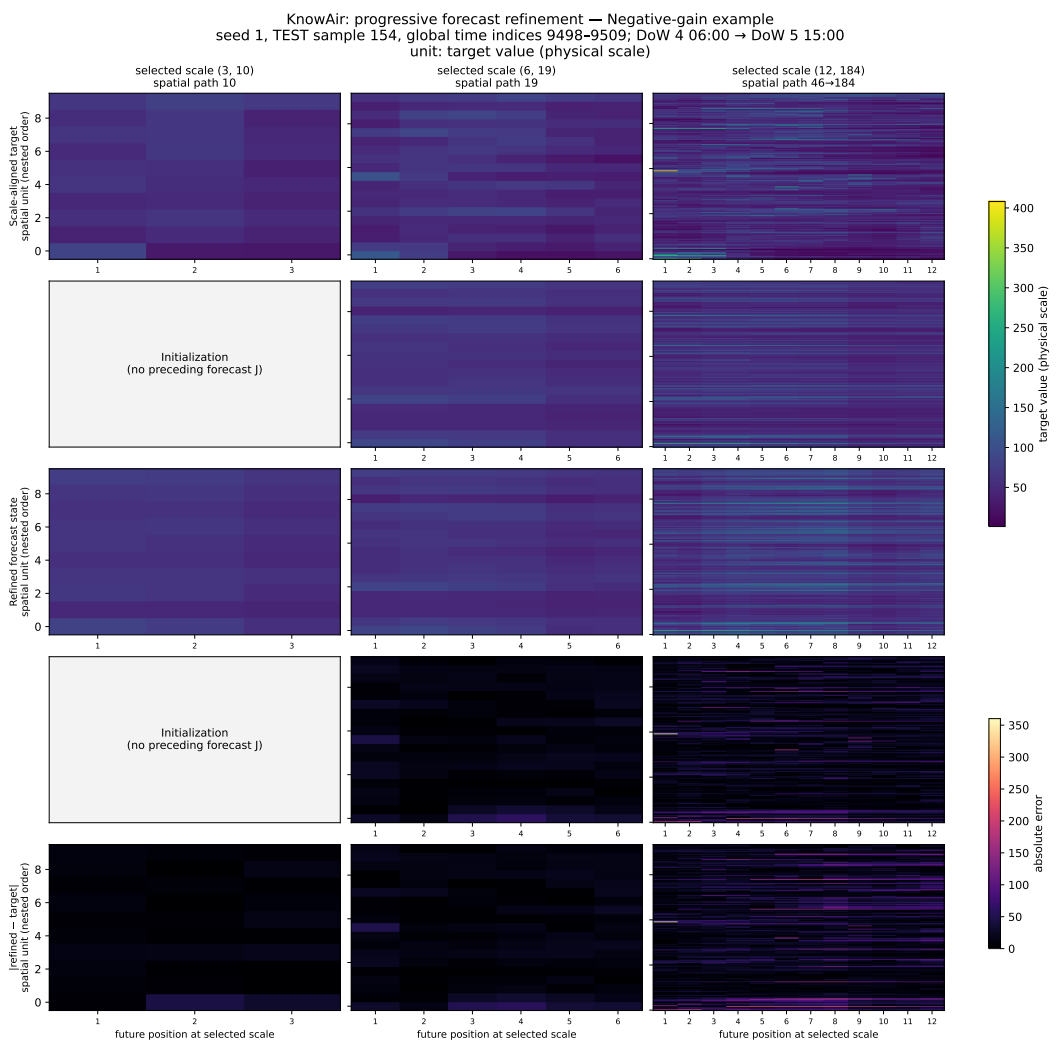


Figure 11: Negative-gain KnowAir window. Columns correspond to selected temporal–spatial scales. Rows show the aligned target, expanded preceding forecast, refined forecast, and their absolute errors. Coarse views use hierarchical aggregation; the initial scale has no preceding forecast.

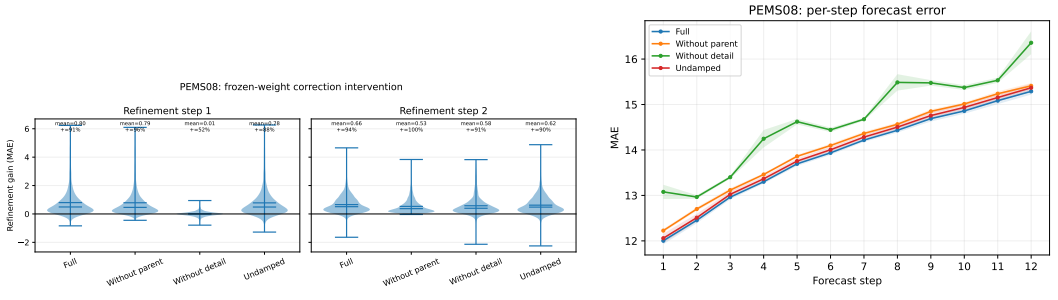


Figure 12: Recursive correction interventions on PEMS08. Left: gain distributions at both refinement steps, including negative gains. Right: MAE at each of the 12 forecast steps. All interventions retain the trained parameters and recursively propagate their own forecast outputs.

Parameters are frozen, and each intervention's output is used recursively at the next selected pair. Terminal MAE follows the formal evaluator: masked MAE is calculated for each test batch of size 16 and averaged across batches. Window-level gains instead assign one MAE difference to each test window. Table 9 reports paired terminal-MAE differences from the full update. In float32 normalized space, both the maximum absolute aggregated-detail residual and the maximum full-update identity residual are 4.768×10^{-7}, below the reporting tolerance of 10^{-6}.

Table 9: Paired terminal-MAE increases relative to the full update. Subscripts are sample standard deviations of seed-wise differences. Each entry is positive in all three seeds.

Dataset	Without parent	Without detail	Undamped
PEMS04	$.1430_{\pm.1046}$	$.2859_{\pm.0098}$	$.0168_{\pm.0070}$
PEMS07	$.1737_{\pm.0504}$	$.6147_{\pm.0938}$	$.0355_{\pm.0096}$
PEMS08	$.1646_{\pm.0240}$	$.7272_{\pm.0343}$	$.0663_{\pm.0242}$
KnowAir	$1.3801_{\pm.0274}$	$.1775_{\pm.0145}$	$.1295_{\pm.0342}$

Figures 6, 8, 10, and 12 show the gain distributions and forecast-step errors. The distributions are descriptive window-level summaries; paired uncertainty is assessed across the three seeds in Table 9.

B.4 Scale-selection sequence error and cost

Figure 13 supplements the training comparison in Section 6.2. It compares terminal error across evaluated sequences under fixed-sequence training and training over scale selections, at unchanged sequence-specific forecasting costs.

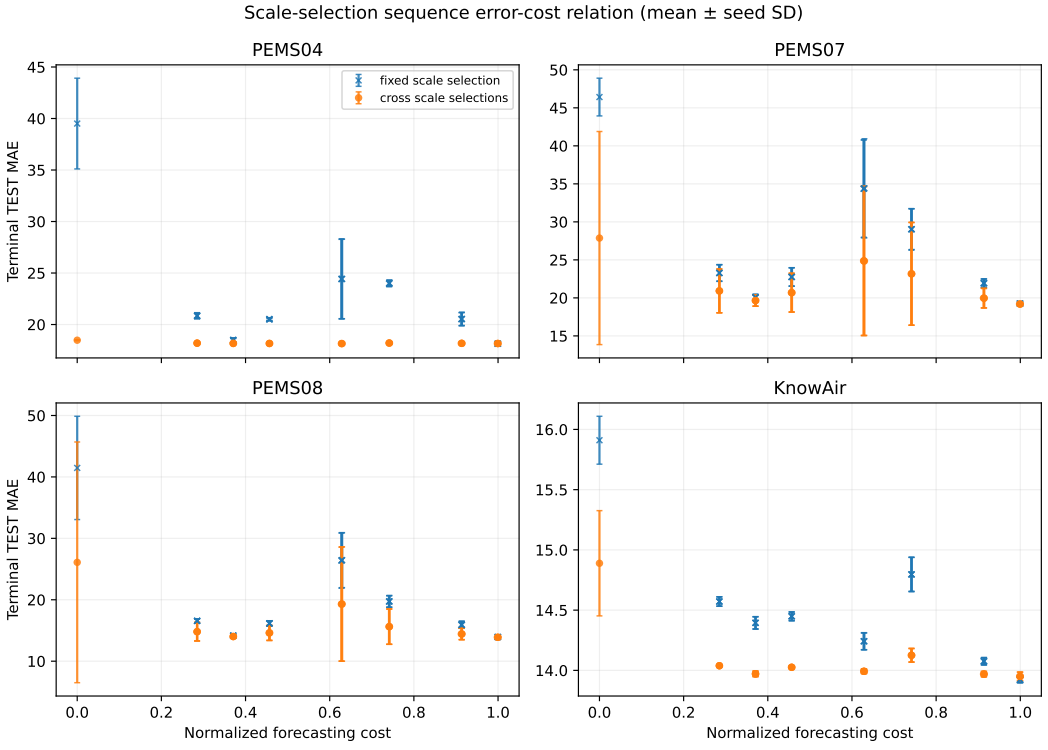


Figure 13: Terminal test MAE versus normalized forecasting cost for the evaluated scale-selection sequences. Error bars show standard deviations across seeds. This comparison evaluates training procedures over a fixed candidate set, rather than learned selection at inference.

C Adaptive Scale Selection Analysis

C.1 Cost and selection protocol

Budget definition. For each $\Omega \in \mathfrak{X}$, let $C(\Omega)$ be profiler-supported FLOPs for history encoding and execution of that forecasting sequence on one validation sample. With $C_{\min}$ and $C_{\max}$ denoting the extrema over the 22 candidates, its normalized cost is

$$c(\Omega) = \frac{C(\Omega) - C_{\min}}{\max(C_{\max} - C_{\min}, 1)}. \tag{56}$$

A budget b restricts selection to $c(\Omega) \leq b + 10^{-8}$. We call the two input-dependent settings *Adaptive–Tight* ($b = 0.50$, 10 feasible sequences) and *Adaptive–Loose* ($b = 0.75$, 16 feasible sequences). These counts hold for PEMS04 and KnowAir across all three seeds. The names refer to cost ceilings, not actual average consumption: Adaptive–Tight places the raw-cost threshold halfway between the minimum and maximum, approximately 81.5% of the maximum in these experiments.

Two fixed references complete the evaluation. *Min-cost* selects direct prediction at $(12, N)$; the original settings $b = 0$ and 0.25 both admit this single sequence and are represented by this one reference. *Train-best fixed* ($b = 1$) selects the sequence with the lowest mean training MAE among all 22 candidates. Both references bypass the predictor.

Selection and calibration. The evaluated selector ranks centered terminal-loss predictions with a cost penalty. Each sequence's mean training MAE is subtracted from its per-sample loss target. For Adaptive–Tight and Adaptive–Loose, the predicted residual is used directly, without restoring that mean, and the score adds $\beta(1 - b)c(\Omega)/b$, where β is the calibrated cost coefficient. The feasible sequence with the lowest score is selected; ties follow the fixed candidate ordering. This experimental score differs from the absolute terminal-loss predictor in Appendix A.2.5. Adaptive–Tight and Adaptive–Loose share the same predictor and β, with effective cost coefficients β and $\beta/3$, respectively.

A chronological 80/20 training split supports predictor fitting and calibration. For each minimum leaf size, β is searched from 0.05 to 1.50 in increments of 0.01. Calibration maximizes the smaller matched-cost gain under Adaptive–Tight and Adaptive–Loose plus one quarter of their mean. Matched-cost gain compares the selector's full-split MAE with the best input-independent mixture of fixed sequences at its achieved mean cost. After refitting each predictor on all training data, validation prioritizes positive gain at both budgets, then the same score; if no candidate improves both, it selects the highest score overall. Accuracy relative to Train-best fixed is recorded separately from the model-selection criterion. Test targets are excluded from fitting, calibration, and model selection.

FLOPs reduction is one minus the ratio of mean selected-sequence FLOPs to the FLOPs of Train-best fixed from the same run. The latter need not be the maximum-cost sequence. Costs include history encoding and forecasting operations supported by the profiler, and exclude selector features, predictor execution, and exhaustive counterfactual evaluation.

C.2 Operating points and paired comparisons

Figure 14 shows the operating points on PEMS04 and KnowAir. All strategies use the same three seeds and test samples for each dataset. Table 11 reports paired differences. The oracle uses test targets and gives a test-informed lower bound over the feasible set.

C.3 Selection frequencies and matched-cost comparison

Selection behavior. Figure 15 shows that direct prediction remains the most frequent choice in the displayed distributions, especially on KnowAir. The looser budget shifts some selections toward longer or finer intermediate sequences.

The frequency plots summarize aggregate choices, rather than per-window assignments. Labels use (r, s_l) with $s_1 < s_2 < s_3 < N$; each dataset uses its own spatial hierarchy. The displayed sequences retain the frequencies from the supplied evaluation plots.

Table 10: Full-test selection results ($H = F = 12$, seeds 1–3). MAE is mean ± sample standard deviation; FLOPs is the mean forecasting cost in billions and excludes selector overhead. Tight/Loose admit 10/16 sequences. Min-cost uses direct prediction; Train-best fixed minimizes mean training MAE over all 22 sequences.

Dataset	Selection setting	Test MAE ↓	Forecasting FLOPs (G) ↓
PEMS04	Min-cost	18.4956 ± 0.0471	7.624
	Adaptive–Tight	18.3635 ± 0.0852	8.162
	Adaptive–Loose	18.3198 ± 0.0787	8.385
	Train-best fixed	18.1746 ± 0.1050	12.091
KnowAir	Min-cost	14.8897 ± 0.4369	4.568
	Adaptive–Tight	14.7936 ± 0.3454	4.626
	Adaptive–Loose	14.7298 ± 0.3253	4.657
	Train-best fixed	13.9499 ± 0.0358	7.247

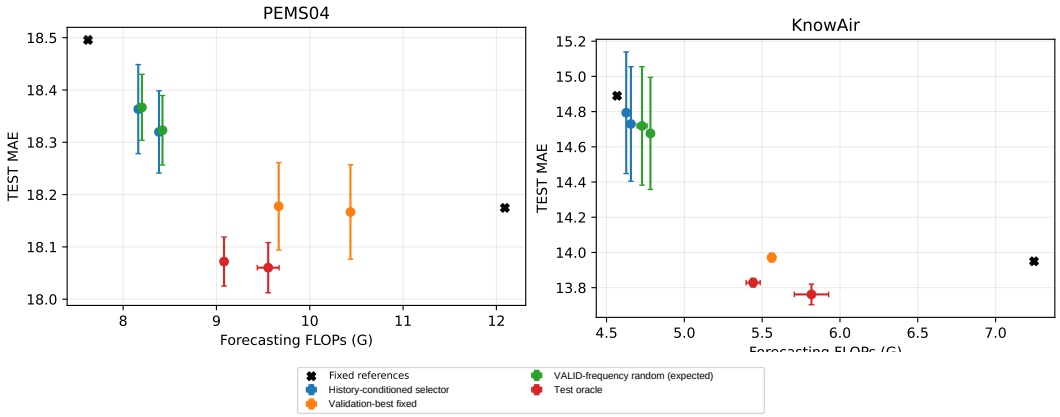


Figure 14: Full-test MAE versus forecasting FLOPs. Adaptive–Tight and Adaptive–Loose give two operating points per strategy; fixed references are Min-cost and Train-best fixed. Error bars show sample standard deviations across three seeds. Table 11 reports paired comparisons. The random strategy uses frozen validation frequencies, so its achieved test cost need not equal the selector's. Lines connect evaluated settings, not an interpolated frontier.

Table 11: Paired MAE differences (baseline minus selector) and mean oracle regret (selector minus oracle). Positive differences favor the selector. All available seeds are included; ± denotes sample standard deviation. Tight and Loose abbreviate the two adaptive settings.

Dataset	Setting	Seeds	Validation-fixed	VALID-frequency random	Oracle regret
PEMS04	Tight	3	-0.1859 ± 0.0028	$+0.0034 \pm 0.0222$	0.2915
	Loose	3	-0.1531 ± 0.0139	$+0.0032 \pm 0.0132$	0.2595
KnowAir	Tight	3	-0.8225 ± 0.3621	-0.0749 ± 0.0143	0.9651
	Loose	3	-0.7587 ± 0.3424	-0.0536 ± 0.0156	0.9681

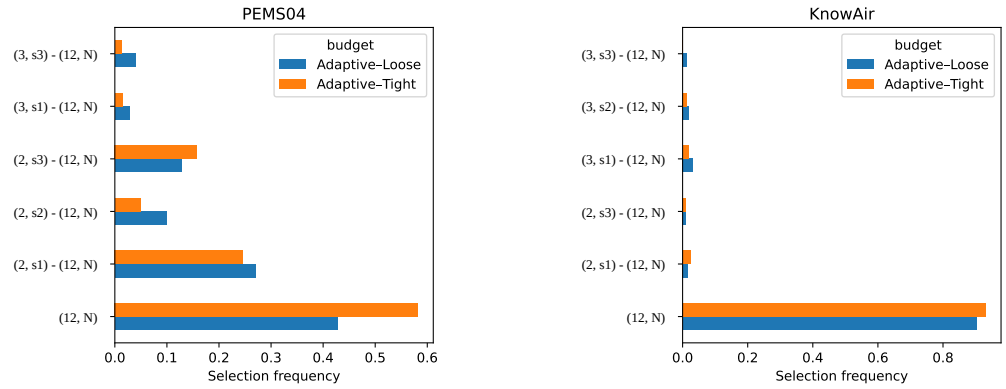


Figure 15: Selection frequencies under Adaptive–Tight and Adaptive–Loose. Each label lists successive temporal–spatial scale pairs; $s1, s2, s3$ denote s_1, s_2, s_3 and N denotes the dataset's node count. Direct prediction at $(12, N)$ remains prominent under both settings, while the looser ceiling permits additional intermediate scales. The panels show the supplied subset of sequence frequencies.

The matched-cost comparison fits an input-independent mixture post hoc from test sequence means at the selector's actual mean test cost. Its gain is mixture MAE minus selector MAE, evaluated over all test windows. Table 12 includes all three seeds, including negative gains. This comparator uses test targets and is distinct from the validation-selected fixed and random strategies above.

Table 12: Gain against the post-hoc test-cost-matched mixture: all-seed mean $\pm$ sample standard deviation, with the number of positive-gain seeds in parentheses.

Dataset	Adaptive–Tight	Adaptive–Loose
PEMS04	$+0.0107 \pm 0.0220$ (2/3)	$+0.0044 \pm 0.0134$ (2/3)
KnowAir	$+0.0257 \pm 0.0514$ (2/3)	$+0.0569 \pm 0.0531$ (2/3)