RAGE: Reference-Guided Data Resampling via Adaptive Gradient Estimation for Time Series Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

Data allocation can improve the downstream performance of time-series foundation models by prioritizing source windows relevant to a target task. However, time-series trajectories are often delimited by data acquisition rather than semantic boundaries and can contain heterogeneous local training signals. Averaging gradients over an entire trajectory can obscure this variation, whereas computing a gradient for every overlapping window is costly. This creates a challenge of retaining useful local information for reference-guided allocation under a limited gradient budget. We propose RAGE, a reference-guided resampling method that estimates and scores local gradient shapes. RAGE places sparse gradient anchors using coverage and refinement guided by interpolation risk and direction cancellation. Neighboring anchors define linear gradient shapes on a common unit interval, encoded by jointly energy-normalized midpoint and slope components. RAGE aggregates signed source–reference shape similarities with trajectory-balanced reference weights and maps the resulting block-score ranks to source-window sampling probabilities. We evaluate RAGE on Chronos-Base, TimesFM 2.5, and TimerXL across seven forecasting tasks and two seeds, comparing against selection baselines and component ablations. Under a nominal 1.5% window-gradient budget, RAGE reduces MASE by 2.03% on average relative to random resampling and by 2.37% relative to LESS-TS. RAGE achieves the highest mean improvement among the evaluated methods and ablations overall and on each model.

1 Introduction

Data allocation is key to improving performance. Time series foundation models transfer forecasting knowledge through pre-training and adaptation on related data (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Goswami et al., 2024; Liu et al., 2025b; Shi et al., 2025; Liu et al., 2026b; Faw et al., 2025; Lee et al., 2026). A source pool supplies training windows, while a small reference set describes the downstream distribution. Under a fixed number of training draws, allocating more exposure to useful windows can improve adaptation without additional optimization steps: *which source windows should be sampled more often?*

Data selection identifies training examples useful for a target task (Albalak et al., 2024). Quality- and diversity-based methods offer complementary selection criteria (Wettig et al., 2024; Wu et al., 2025). Influence-based methods estimate how much each training example affects a target prediction (Koh & Liang, 2017; Pruthi et al., 2020). Gradient-matching methods select examples whose gradients are close to those of the training or validation set (Killamsetty et al., 2021). LESS and TSDS use a small target set or target distribution to find useful data for downstream fine-tuning (Xia et al., 2024; Liu et al., 2024).

Time-series data property makes gradient averaging more likely to obscure window-level training signals. Applying reference-guided gradient methods to time series reveals a key difference in how training samples are organized. Language data are usually written by people or generated by language models, and can often be naturally divided into meaningful semantic units to be directly used for training. In contrast, time series data are commonly organized as time-ordered trajectories, while forecasting models often process them into fixed-length context windows or patches (Nie et al.,

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

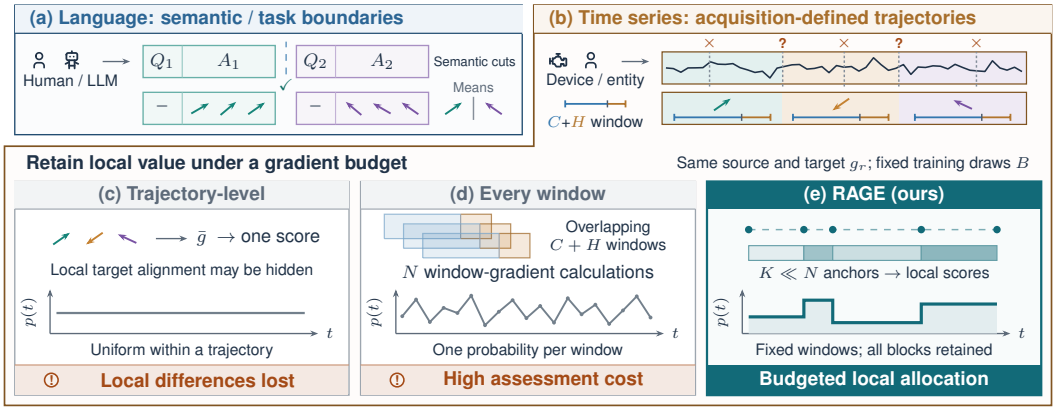

Figure 1: **Sample boundaries to budgeted local valuation.** Language has semantic boundaries (a), while recorded trajectories lack them (b). Trajectory averaging hides local differences (c); exhaustive window scoring is costly (d). RAGE balances local detail and computation (e).

2023; Ansari et al., 2024; 2025; Khwaja et al., 2026). Here, a trajectory may contain several latent patterns without clear boundaries (Lee et al., 2026; Liu et al., 2026a), and consequently, the average gradient over all its windows describes their combined update but can hide local differences relevant to resampling. Estimating a gradient for every legal window would reveal those differences, but heavily overlapping windows make this expensive to compute, store, and score (Kwon et al., 2024; Zhang et al., 2025; Wu et al., 2026). Recent gradient-based selection methods reduce valuation cost using influence-preserving proxies, landmark-based propagation, or compact target subspaces (Chen et al., 2026; Nikdan et al., 2025; Min et al., 2026); these approaches motivate cheaper estimation but do not specify where along a time trajectory to spend a limited window-gradient budget. Figure 1 summarizes this trade-off, raising a practical question: *How can we retain useful local information under a limited gradient-computation budget?*

Designing a reference-guided method to resample source windows with a limited gradient budget is nontrivial. This raises three connected challenges:

(C1) Where should limited gradient computations be spent? With a fixed number of gradient evaluations, anchor placement must cover both source and reference trajectories while resolving local gradient variation. Uniform spacing provides coverage, but long intervals can hide interpolation error, and opposing endpoint gradients can signal distinct local directions. Furthermore, as gradients remain unknown at unsampled points, the method must use known results as evidence to choose which interval to split next, thereby allocating the remaining budget.

(C2) How should sparse gradients represent local training signals? Given sparse anchors, the method must represent the gradient variation between them and assign every legal window to a scoring unit. A single mean direction discards how gradients change within that unit, while separately scoring every interpolated window repeats information supplied by the same anchors. A compact representation must retain both the local gradient level and its variation.

(C3) How should local reference information determine training exposure? A target task may be represented by reference trajectories of different lengths, each containing several local forecasting needs. Weighting all reference windows equally can let long trajectories dominate; weighting all local units equally can instead favor trajectories divided into more units. Their local gradient shapes may also disagree. Together, these properties make it difficult to combine reference signals when scoring source shapes and assigning sampling probabilities to source windows.

To address these challenges, we propose RAGE, a reference-guided **R**esampling method based on budgeted local **A**daptive **G**radient **E**stimates. RAGE computes gradients at sparse anchors across each trajectory and refines intervals with high interpolation risk or opposing direction cancellations. Neighboring anchors define compact blocks whose linearly reconstructed gradient shapes retain their midpoint and slope; weighted local reference blocks score these shapes. Finally, the scores are converted into window sampling probabilities, adjusting training exposure across and within trajectories.

We summarize our contributions as follows.

1. **A budgeted formulation of time-series foundation-model adaptation via reference-guided resampling.** Time-series trajectories can vary locally, and their fixed-length forecasting windows follow a protocol rather than semantic boundaries. We therefore formulate downstream adaptation as choosing a distribution over legal source windows to minimize expected target loss after B post-training draws (Definition 2.1 and Eq. (1)). The setting limits candidate assessment to at most K window-gradient evaluations and uses source–reference gradient-shape similarity to allocate exposure (Section 3.3).
2. **Budgeted gradient estimation and reference-guided block-shape scoring for time-series adaptation.** Our method covers each trajectory with sparse gradient anchors, then spends the remaining gradient budget where midpoint residuals signal interpolation risk or endpoint directions oppose each other (Eqs. (5) and (6)). The anchors estimate gradients at unobserved windows and divide each trajectory into nonoverlapping blocks represented by their reconstructed linear shapes (Eq. (11); Section 3.2). By scoring these shapes against references and converting the scores into a sampling distribution, RAGE gives more training exposure to source regions aligned with the target task (Sections 3.3–3.4; Algorithm 1).
3. **Forecasting gains across models and component ablations.** We report RAGE results on Chronos-Base, TimesFM 2.5, and TimerXL across seven tasks and two training seeds. RAGE reduces MASE relative to random resampling by 2.03% on average, with a median reduction of 1.32% and improvements in 28 of 42 paired cells. All seven task-level means and both seed-level means are positive. RAGE ranks first among the evaluated methods on all three model-level aggregates, with gains of 1.93%, 1.79%, and 2.38% on Chronos, TimesFM, and TimerXL, respectively. Paired ablations examine within-trajectory allocation, scoring granularity, shape representation, and reference aggregation, linking the forecasting gains to the main design choices.

2 Problem Formulation

Let $\mathcal{S}$ be a source pool of trajectories, $\mathcal{R}$ a reference set, f_{θ_0} a forecasting model with pretrained parameters θ_0 and the downstream distribution $\mathcal{D}_{\text{tar}}$ approximately represented by $\mathcal{R}$.

Definition 2.1 (Fixed-window forecasting task)**.** For a trajectory x_i of length L_i, fix context length C, forecast length H, and forecast-start stride D. The legal forecast start indices are $\mathcal{T}_i = \{C, C + D, \dots\} \cap [C, L_i - H]$. Each index t identifies the forecasting event $z_{i,t} = \big(x_i[t-C : t], x_i[t : t+H]\big)$, and the candidate source windows are indexed by $\mathcal{W} = \{(i,t) : i \in \mathcal{S},\ t \in \mathcal{T}_i\}$. The event $z_{i,t}$ pairs a context with the next H observations, and each pair $(i,t) \in \mathcal{W}$ identifies one source window. Thus $M_i = |\mathcal{T}_i|$ counts the windows available from trajectory i, and $N = |\mathcal{W}| = \sum_{i \in \mathcal{S}} M_i$ counts all source windows available for resampling. For an event loss $\ell(\theta; z)$ at model parameters θ, the downstream objective is $\mathcal{L}_{\text{tar}}(\theta) = \mathbb{E}_{z \sim \mathcal{D}_{\text{tar}}}[\ell(\theta; z)]$, where $\mathcal{D}_{\text{tar}}$ is a distribution over forecasting events under the same window protocol.

Definition 2.2 (Source-window resampling)**.** Choose a probability $p_{i,t}$ of drawing each source window $z_{i,t}$. We write $p \in \Delta(\mathcal{W})$ for the set of valid choices: every $p_{i,t} \geq 0$, and $\sum_{(i,t) \in \mathcal{W}} p_{i,t} = 1$. With B training draws with replacement, the expected exposure of that event is $Bp_{i,t}$. The total probability of trajectory i is $P_i = \sum_{t \in \mathcal{T}_i} p_{i,t}$. Trajectory-level sampling fixes $p_{i,t} = \frac{P_i}{M_i}$ within each trajectory; a finer-grained allocation permits different probabilities within that trajectory. Let $\text{Train}_B(\theta_0, p; \xi)$ be a fixed post-training procedure using B draws, with ξ denoting training randomness. The resampling problem is

$$\min_{p \in \Delta(\mathcal{W})} \mathbb{E}_\xi \left[\mathcal{L}_{\text{tar}}\big(\text{Train}_B(\theta_0, p; \xi)\big) \right]. \tag{1}$$

The training procedure and B are fixed; only the source-window probabilities change.

Definition 2.3 (Gradient alignment as motivation)**.** Reference gradients provide a local signal for the resampling objective in Definition 2.2. Let G_s and G_r be gradients of a source and a reference forecasting event at θ_0. For nonzero vectors, define

$$a_{\text{dot}}(G_s, G_r) = G_s^\top G_r, \qquad a_{\cos}(G_s, G_r) = \frac{G_s^\top G_r}{\|G_s\|_2 \|G_r\|_2}. \tag{2}$$

The dot product gives the first-order loss-change term; cosine measures normalized directional agreement in $[-1, 1]$, with $a_{\cos} = 0$ if either vector is zero. This motivates reference guidance. RAGE scores energy-normalized local gradient shapes (Eq. (7)), whose midpoint and slope jointly describe an interval. Appendix B distinguishes the single-update loss approximation from this shape criterion.

Together, Definitions 2.2 and 2.3 motivate allocating a fixed number of training draws using reference information. RAGE reconstructs local gradient shapes with at most K window-gradient computations, scores those shapes against references, and reallocates exposure within trajectories.

3 The RAGE Method

Estimate, divide, score, and resample. Figure 2 summarizes the four stages. Source and reference trajectories undergo the same gradient estimation and are divided into blocks. Their reconstructed gradient shapes are compared, and the resulting ranks determine a distribution over source windows for post-training.

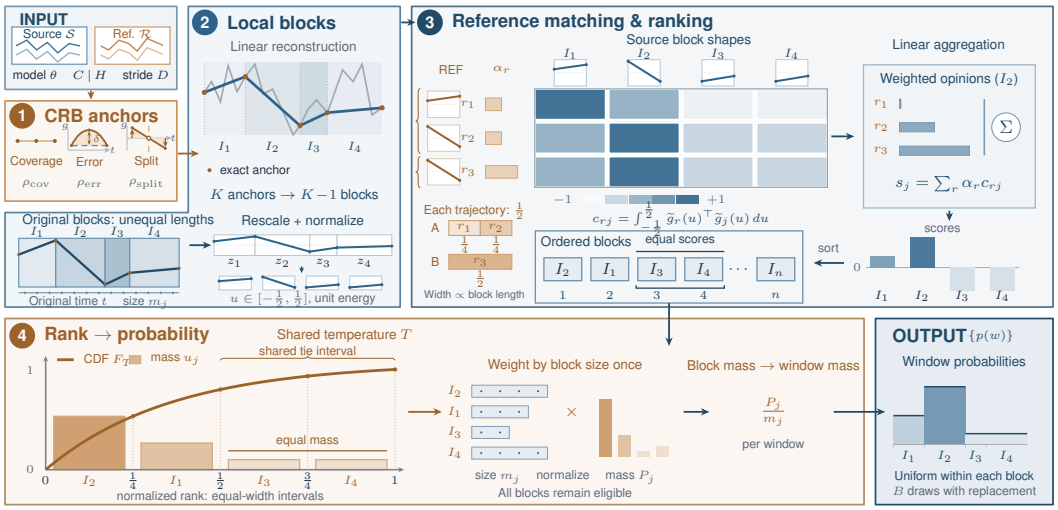

Figure 2: RAGE estimates sparse gradients (1), constructs normalized block shapes (2), scores them against weighted references (3), and maps score ranks to source-window probabilities (4).

3.1 Estimating Gradients with Adaptive Anchors

Continuous Reconstruction under a Budget (CRB) selects gradient anchors in three stages: coverage spreads them across legal forecast starts, reconstruction-risk refinement targets intervals with large observed midpoint residuals, and cancellation refinement targets intervals with opposing endpoint gradients.

Budget. Given source trajectories $\mathcal{S}$, reference trajectories $\mathcal{R}$, a pretrained forecasting model at θ_0, window dimensions (C, H, D), a fixed dimension-reducing projection Π, and stage densities $\rho = (\rho_{\text{cov}}, \rho_{\text{err}}, \rho_{\text{split}})$, CRB returns an anchor map A_i of selected starts and projected gradients for every $i \in \mathcal{S} \cup \mathcal{R}$. At each anchor t, it stores the dimension-reduced gradient $g_i(t) = \Pi \nabla_\theta \ell(\theta_0; z_{i,t}) \in \mathbb{R}^d$, where Π maps the full parameter gradient to d dimensions. All three stage budget ratios are 5×10^{-3}; we use $d = 4096$ and projection seed 42. Each anchor uses one gradient-computation slot. For trajectory i, let $M_i = |\mathcal{T}_i|$ be the number of legal starts; the densities allocate the following coverage, risk, and cancellation caps:

$$\begin{aligned} k_{i,\text{cov}} &= \min\{M_i, \max\{2, \text{round}(\rho_{\text{cov}} M_i)\}\}, \\ k_{i,\text{err}} &= \min\{\text{round}(\rho_{\text{err}} M_i),\ M_i - k_{i,\text{cov}}\}, \\ k_{i,\text{split}} &= \min\{\text{round}(\rho_{\text{split}} M_i),\ M_i - k_{i,\text{cov}} - k_{i,\text{err}}\}. \end{aligned} \tag{3}$$

Rounding uses the nearest integer with ties to even. Across both pools, CRB computes exactly $K = \sum_{i \in \mathcal{S} \cup \mathcal{R}} (k_{i,\text{cov}} + k_{i,\text{err}} + k_{i,\text{split}})$ gradients. This trajectory-wise refinement follows local approximation strategies (Chang et al., 1997; Jakeman & Roberts, 2011; Crombecq et al., 2011).

Coverage, reconstruction risk, and cancellation. CRB runs the three stages in order. For unit stride, each stage repeatedly bisects a splittable interval $[a, b]$ at $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ and computes the gradient there until its budget is spent (Algorithm 2 in Appendix F).

1. **Coverage maximizes uniform spread.** Compute gradients at the first and last legal starts, then repeatedly split the longest current interval until the coverage cap is reached. This greedily prevents a long unobserved span from remaining while the rest of the trajectory receives denser observations. Each new midpoint also reveals the residual δ_I between its computed gradient and the gradient predicted by linearly interpolating the parent interval's endpoint gradients (Eq. (4)).
2. **Risk refinement targets integrated interpolation error.** For a split parent interval $I = [a, b]$, compare its computed midpoint gradient with the linear interpolation estimate $\widehat{g}_i(m)$ defined in Eq. (11):

$$\delta_I = \|g_i(m) - \widehat{g}_i(m)\|_2, \qquad \widehat{c}_I = \frac{2\delta_I}{(m-a)(b-m)}. \tag{4}$$

   Here δ_I is the observed midpoint residual, $\widehat{c}_I$ estimates local gradient curvature, and $\|\cdot\|_2$ is the Euclidean norm. Both children inherit the new curvature estimate. An active interval of width $h_I = b - a$ has priority

$$R_I = \frac{h_I^5 \widehat{c}_I^2}{120}. \tag{5}$$

   Equations (4) and (5) follow from a local quadratic approximation of the gradient between anchors; the derivation is in Appendix D. Thus, R_I estimates integrated squared interpolation error. Split the interval with the largest R_I, then update its children immediately. An interval without an inherited curvature observation uses the largest curvature estimate observed up to the current risk-refinement step; if none exists, it uses 1.
3. **Cancellation refinement targets opposing directions.** Endpoint gradients can have a weak mean even when both are strong. For each current interval define

$$\kappa_I = \frac{\|g_i(a) + g_i(b)\|_2}{\|g_i(a)\|_2 + \|g_i(b)\|_2}. \tag{6}$$

   Smaller κ_I indicates stronger endpoint cancellation. Split the interval with the smallest current κ_I, then recompute the children's priorities before the next choice. If both endpoint gradients are exactly zero, set $\kappa_I = 1$; one exactly zero endpoint also gives 1. The cancellation score uses raw projected gradients.

All three stages use their full capacity-clipped budgets without a score threshold or early stop. Exact priority ties select the rightmost interval. The risk and cancellation stages address different signals: opposing endpoints need not imply a large interpolation residual. Gradient interpolation and projected-feature details are in Appendix C.

3.2 Constructing Local Gradient Shapes

Gradients between neighboring anchors are estimated by linear interpolation without further backpropagation, so we group their legal starts into blocks. Neighboring anchors $a_j < b_j$ define a block j of m_j legal starts. Blocks own starts in $[a_j, b_j)$, with the last block also owning its right endpoint, so each legal start belongs to exactly one block. A trajectory with one legal start forms a singleton block with zero slope. Shape features always use the two observed anchor gradients. Rescale each block to $u \in [-\frac{1}{2}, \frac{1}{2}]$ and write its reconstructed gradient as $g_j(u) = \bar{g}_j + u d_j$, where $\bar{g}_j = \frac{1}{2}(g_i(a_j) + g_i(b_j))$ and $d_j = g_i(b_j) - g_i(a_j)$. Its integrated squared energy is $E_j = \|\bar{g}_j\|_2^2 + \frac{\|d_j\|_2^2}{12}$. Appendix C derives this normalization. We represent the block by the unit shape

$$z_j = \frac{[\bar{g}_j;\ \frac{d_j}{\sqrt{12}}]}{\sqrt{E_j}}, \qquad z_j = 0 \text{ if } E_j = 0. \tag{7}$$

Reference blocks use the same construction, giving shapes z_r and sizes m_r. The $2d$-dimensional shape uses cached endpoint gradients without another backward pass. It retains both midpoint and slope. We also use unit($\bar{g}_j$) as an ablation: it retains only the direction of the average of the two endpoint gradients and discards the slope component.

3.3 Scoring Local Gradient Shapes with References

Weight local reference requirements. A reference block r scores source block j by the normalized shape inner product $c_{rj} = z_r^\top z_j$. For nonzero shapes, this equals $1 - \frac{\|z_r - z_j\|_2^2}{2}$, comparing their linearly reconstructed gradients on a common unit interval. Let traj(r) be the reference trajectory containing block r, and let $M_{\text{traj}(r)}$ be its number of legal forecast starts. To balance reference trajectories while accounting for the windows covered by each block, assign $\alpha_r = \frac{1}{|\mathcal{R}|} \frac{m_r}{M_{\text{traj}(r)}}$. Then $\sum_r \alpha_r = 1$ across all reference blocks, and each reference trajectory contributes the same total weight.

Combine reference opinions. RAGE uses **Linear** aggregation of reference opinions, $s_j = \sum_r \alpha_r c_{rj}$. This preserves the signed average alignment of each source shape with the reference shapes. Aggregation ablations use $\sum_r \alpha_r e^{c_{rj}}$, $\sum_r \alpha_r \max(c_{rj}, 0)$, and $\max_r(\alpha_r c_{rj})$ for **Exp**, **ZeroClip**, and **Max**, respectively. These variants are denoted RAGE/Exp, RAGE/ZeroClip, and RAGE/Max in the experiments. The exponential transform uses temperature one; Max takes the largest weighted opinion. All four rank every source block with the same probability rule, without hard gating.

3.4 Resampling Training Windows from Block Scores

Rank scores with a fixed preference strength. Directly mapping scores to sampling mass makes selection strength depend on their scale and spacing, which can vary across scoring methods. Even after scale normalization, scores may be tightly clustered or widely spread. We therefore sort the Q source blocks by descending score and use ranks to control selection strength through a separate temperature $T > 0$. Let $k \in \{1, \ldots, Q\}$ be a block's rank and $F_T(x)$ the cumulative rank curve at normalized position $x \in [0, 1]$. Assign the rank mass u_k by

$$u_k = F_T\left(\frac{k}{Q}\right) - F_T\left(\frac{k-1}{Q}\right), \qquad F_T(x) = \frac{1 - e^{-\frac{x}{T}}}{1 - e^{-\frac{1}{T}}}. \tag{8}$$

If an exact tie spans rank positions l through h, each tied block receives $\frac{F_T(\frac{h}{Q}) - F_T(\frac{l-1}{Q})}{h-l+1}$. Smaller T concentrates more mass on higher ranks. We set $T = \frac{1}{2 \ln 4}$, giving $F_T(\frac{1}{2}) = 0.8$ before block-size weighting; ties crossing the halfway boundary share mass across it.

Account for the windows represented by a block. A block with more legal forecast starts contains more training events. Write u_j for the rank mass assigned to source block j. Weight it by block size to obtain $P_j = \frac{m_j u_j}{\sum_{q=1}^{Q} m_q u_q}$, then assign each event $t \in j$ probability $p_{i,t} = \frac{P_j}{m_j}$. These probabilities are nonnegative and sum to one. If all scores tie, every u_j is equal and every source window receives probability $\frac{1}{N}$. Thus, the volume factor restores uniform window exposure when all preferences are equal. Score differences then produce local reallocation.

Finally sample from block to window. For each training draw, sample block $j \sim P$, sample one of its origins uniformly, and extract the corresponding context–forecast pair. Repeat this for the fixed exposure budget and update the model with its forecasting loss. Algorithm 1 in Appendix E gives the full procedure.

In the end, RAGE returns $p = \{p_{i,t}\}$, a sampling distribution over all legal source forecast starts, together with the source-block map $\mathcal{J}$, block masses P, and post-trained parameters θ_B. At the nominal 1.5% anchor density, it computes roughly one window gradient per 67 legal starts on long trajectories; scoring and probability construction then use the stored features without further backpropagation.

4 Experiments

4.1 Experimental Setup

Tasks and data. We report seven forecasting tasks: Azure CPU, Darwin water quality, NEM demand, NEM price, port activity, solar, and coastal temperature and salinity. These tasks come from six TIME benchmark configurations (Qiao et al., 2026). Full dataset identifiers and sampling frequencies are in Appendix G.

We measure source-pool size in *window-equivalents*: a trajectory of length L contributes $\left\lfloor \frac{L}{576} \right\rfloor$, where 576 is the common context-plus-forecast length of the three models. This unit measures data volume and sets the nominal epoch budget, rather than counting all overlapping legal windows. The tasks share a feasible source pool of 65,136 window-equivalents, constructed from complete trajectories sampled with a fixed pool seed from a collection combining Chronos pretraining data (Ansari et al., 2024), LOTSA (Woo et al., 2024), Monash (Godahewa et al., 2021), GIFT-Eval (Aksu et al., 2024), Time-300B, and UTSD. For each training seed in $\{0, 1\}$, we draw a complete-trajectory subset with exactly 16,284 window-equivalents. The three models and all methods for a seed use the same subset; different seeds may select overlapping trajectories.

Each task uses its full pre-evaluation target history as reference data; this full history excludes the evaluation period. We reserve the trailing evaluation period before constructing reference windows, so every reference window, including its forecast target, ends before that period. Reference windows guide source scoring and are not used for post-training draws.

Metrics. We report held-out **MASE** using a seasonal-naive scale computed from all observed history preceding each forecast origin (lower is better). In rolling evaluation, later origins include previously observed evaluation points in this scale, while the current forecast targets are excluded. Table 1 gives two-seed means at epoch 2, paired average improvement $100\frac{\text{MASE}_{\text{Rand}}-\text{MASE}_{\text{method}}}{\text{MASE}_{\text{Rand}}}$ and wins over 42 cells; blue shades mark trained-method ranks 1–5 per model–task setting. Appendix Table 3 adds overall and model-level means and medians. RAGE uses the same Shape+Linear configuration across all seven tasks.

Recorded training time and selection-file size cover all 42 cells; preparation and selection timing cover 36 cells, excluding Coastal (Appendix I).

Models and training. Chronos-Base and TimesFM 2.5 use $512 \to 64$ context–forecast windows; TimerXL uses $480 \to 96$ to align its context and forecast lengths with its native 96-time-step patches (five context patches and one forecast patch). All methods use full-parameter BF16 AdamW, batch size 32, and replacement sampling. Appendix G gives the pretrained checkpoint identifiers and training recipes; Appendix C specifies the gradient projection.

Baselines. We compare RAGE with seven baselines: **Orig** is the pretrained model without post-training, and **Rand** samples source windows uniformly. **LESS-TS** adapts gradient matching to time series (Xia et al., 2024); **TSDS** combines target-distribution matching with diversity (Liu et al., 2024); and **LP-Mirror** uses linear-proxy valuation and mirrored influence (Ao & Chen, 2026). **G-Dot** ranks source windows by the dot product between projected source gradients and a trajectory-balanced reference mean; **GraNd-Static** ranks them by projected gradient norm without reference information. These two window-level baselines use uniformly spaced anchors under the same gradient budget and interpolate gradients at uncomputed starts. All trained methods use the same seed-specific pool and nominal exposure budget. Appendix H specifies the adaptations, scoring units, and key baseline configurations.

Component ablations. RAGE represents each local block using both the midpoint and the slope of its reconstructed gradient, as defined in Eq. (7), and combines reference scores with **Linear** aggregation. RAGE/Exp, RAGE/ZeroClip, and RAGE/Max retain this representation and change only the reference aggregation rule.

Block ablations average the two endpoint gradients of each block and normalize the resulting vector, discarding the slope component. Trajectory ablations instead use one mean gradient direction for each entire trajectory, discarding variation between blocks. Both groups use **Linear**, **Exp**, **ZeroClip**,

and **Max** aggregation to compare the effect of retaining gradient slope and scoring locally rather than at the trajectory level. Four allocation ablations use the block mean representation and cross **Exp** or **Linear** aggregation with **Block-Uni** or **Win-Uni** allocation. **Block-Uni** divides a trajectory's mass equally among its blocks; **Win-Uni** divides it among its legal windows. All ablations use the same task, model, seed-specific pool, and training exposure budget.

Table 1: **All forecasting methods and ablations across seven tasks.** Entries are two-seed mean MASE (lower is better). Avg. is the mean paired percentage improvement over Rand across 42 seed–model–task cells; Wins counts positive improvements. Darker blue marks ranks 1–5 among trained candidates; Orig is unranked.

Method	Azure			Darwin			NEM-D			NEM-P			Port			Solar			Coastal			Avg.	Wins
	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	(%)	/42
RAGE	1.2324	0.7639	**1.1011**	1.6561	1.0210	1.2202	1.2808	0.6997	0.9147	1.0528	0.7170	0.8841	0.8884	0.7287	**0.7701**	1.1277	**1.1088**	1.3047	2.0732	1.6225	1.9204	**+2.03**	**28**
S-Exp	1.2534	0.7647	1.1249	1.6663	1.0229	1.2209	1.3909	0.7096	0.9426	1.0567	0.7195	0.8856	0.8882	0.7298	0.7767	1.1348	1.1289	1.3069	2.0844	1.6305	1.9080	+1.01	24
S-ZC	**1.1999**	0.7634	1.1499	1.6500	1.0208	1.2506	1.3804	0.7167	0.9490	1.0659	0.7240	0.9105	0.8881	0.7284	0.7924	**1.0949**	1.1246	**1.2923**	2.0818	1.6295	1.9448	+0.89	25
S-Max	1.2748	0.7631	1.1530	1.6584	1.0359	1.2674	1.4219	0.7426	0.9271	1.0496	0.7298	0.8869	0.8880	0.7267	0.8002	1.3600	1.1296	1.3158	2.0715	1.6360	1.9426	-0.79	17
Orig	1.3318	0.7638	1.2847	1.6639	1.0697	1.3571	1.7385	0.8819	0.9168	1.0622	0.7492	0.8504	0.8821	0.7162	0.7978	1.2167	1.1293	1.3442	2.0731	1.6595	2.0427	-3.68	12
Rand	1.2245	0.7613	1.1510	1.6527	1.0415	1.3064	1.3864	0.7611	0.9321	1.0683	0.7275	0.8715	0.8897	0.7215	0.7975	1.2032	1.1192	1.3144	2.0548	1.6473	1.9561	+0.00	0
LESS-TS	1.2272	0.7664	1.1978	1.6522	1.0341	1.2429	1.3517	0.7368	0.9819	1.0869	0.7203	0.8812	0.8828	0.7269	0.8021	1.2078	1.1296	1.3703	2.0811	1.6352	1.9438	-0.36	19
TSDS	1.2618	0.7668	1.1811	1.6537	1.0405	1.3663	1.3527	0.7107	0.9223	1.0533	0.7223	0.9376	0.9203	0.7388	0.7931	1.2668	1.1150	1.3074	**2.0450**	1.6483	1.9334	-0.72	18
LP-Mir	1.3635	0.7703	1.1524	1.6973	1.0710	1.4046	2.2340	0.8241	0.9463	1.0917	0.7435	0.8938	0.8896	0.7208	0.7941	1.5106	1.1274	1.3047	2.6005	1.6523	2.1473	-7.77	10
G-Dot	1.2331	0.7638	1.3353	1.6527	**1.0159**	**1.2181**	1.3290	**0.6937**	0.9123	1.0616	0.7252	0.9432	0.8955	0.7267	0.7937	1.2657	1.1212	1.4991	2.0727	1.6319	1.9192	-0.90	19
GraNd	1.2182	0.7630	1.3104	1.6585	1.0381	1.3361	1.2949	0.7509	1.0357	**1.0421**	0.7286	0.9765	0.9032	**0.7196**	0.8241	1.1873	1.1333	1.3960	2.0788	1.6365	1.9915	-2.02	20
B-Lin	1.2523	0.7638	1.1455	1.6494	1.0224	1.2186	1.3588	0.7215	**0.8938**	1.0433	0.7163	0.8990	0.8910	0.7288	0.7744	1.1654	1.1375	1.3239	2.0860	1.6221	1.9225	+1.05	27
B-Exp	1.2392	0.7647	1.1735	1.6456	1.0203	1.2477	1.4239	0.7160	0.9922	1.0602	0.7166	0.8824	0.8802	0.7292	0.7754	1.2319	1.1245	1.3296	2.0799	**1.6209**	1.9190	+0.05	26
B-ZC	1.2392	0.7663	1.1488	1.6600	1.0263	1.2476	1.3261	0.7146	0.9147	1.0520	0.7207	0.8996	0.8839	0.7265	0.7951	1.2940	1.1247	1.3304	2.0592	1.6351	1.9250	+0.39	26
B-Max	1.2463	0.7623	1.1553	1.6586	1.0322	1.2903	1.3926	0.7406	0.9547	1.0645	0.7305	0.8834	0.8928	0.7264	0.7942	1.3012	1.1338	1.3492	2.0747	1.6378	1.9143	-0.65	18
T-Lin	1.2316	0.7662	1.2658	1.6628	1.0341	1.2700	1.3736	0.7309	1.0059	1.0804	0.7155	0.9553	0.8946	0.7269	0.8023	1.2560	1.1353	1.3603	2.0496	1.6368	1.9091	-1.39	15
T-Exp	1.2135	0.7651	1.2104	1.6619	1.0373	1.2707	1.4047	0.7254	0.9979	1.0816	0.7153	0.9473	0.8896	0.7267	0.7989	1.2560	1.1353	1.3603	2.0709	1.6287	**1.9065**	-1.05	17
T-ZC	1.2305	**0.7594**	1.1992	1.6666	1.0364	1.2751	1.4849	0.7175	1.0275	1.0778	0.7196	0.9392	**0.8774**	0.7277	0.7921	1.2340	1.1446	1.3576	2.0690	1.6340	1.9402	-1.33	16
T-Max	1.2011	0.7654	1.2022	**1.6455**	1.0350	1.2710	1.4465	0.7330	1.0433	1.0668	**0.7142**	0.9646	0.8886	0.7265	0.8096	1.2560	1.1353	1.3603	2.0535	1.6349	1.9247	-1.42	20
B-Lin/BU	1.2475	0.7632	1.4503	1.6556	1.0409	1.3839	1.3516	0.7592	0.9247	1.0462	0.7250	0.9826	0.8868	0.7232	0.8105	1.1748	1.1247	1.5252	2.0745	1.6235	2.1503	-3.17	16
B-Exp/BU	1.2589	0.7642	1.4992	1.6509	1.0418	1.4838	**1.2726**	0.7625	0.9099	1.0572	0.7228	0.9617	0.8922	0.7234	0.8196	1.2392	1.1399	1.6705	2.0742	1.6259	2.1609	-4.35	16
B-Lin/WU	1.2237	0.7640	1.2085	1.6652	1.0416	1.2916	1.2960	0.7346	0.9613	1.0602	0.7248	**0.8557**	0.8928	0.7239	0.7962	1.2371	1.1320	1.3443	2.0656	1.6387	1.9256	-0.03	20
B-Exp/WU	1.2038	0.7609	1.2497	1.6566	1.0377	1.3015	1.3417	0.7433	0.9701	1.0493	0.7247	0.9129	0.8897	0.7237	0.7974	1.2092	1.1192	1.2979	2.0713	1.6393	1.9726	-0.42	23

Ch=Chronos-Base; TF=TimesFM 2.5; TX=TimerXL. RAGE uses Shape+Linear. S/B/T=Shape/Block/Trajectory; Lin=Linear; ZC=ZeroClip; BU=Block-Uni; WU=Win-Uni. All entries use epoch 2; each method has 42 measured cells.

4.2 Results

Main results. Table 1 compares 23 methods and ablations. RAGE has the highest mean paired improvement over Rand (+2.03%), a positive median (+1.32%), and 28 wins in 42 cells. LESS-TS, the strongest external selection baseline, gives −0.36%. Pairing RAGE directly with LESS-TS gives 2.37% mean MASE reduction and 31 wins in 42 cells, using LESS-TS as denominator. RAGE uses a nominal 1.5% window-gradient budget and the same post-training exposure budget as the trained baselines. Table 2 reports stagewise costs.

Gains are positive on seven tasks and all three models. RAGE has positive mean gains on all seven tasks: NEM demand 5.76%, Darwin 2.77%, Solar 2.48%, Azure 1.10%, Port 0.86%, Coastal 0.78%, and NEM price 0.48%. Seed-level means are 2.53% and 1.54%, with losses in individual model–seed cells. Appendix Table 3 gives the overall and model-level summaries.

RAGE ranks first on each model, with mean gains of 1.93%, 1.79%, and 2.38% on Chronos-Base, TimesFM 2.5, and TimerXL. The ratio-of-mean-MASE gain is also 2.03%. Individual tasks favor alternatives: Shape-ZeroClip on Solar, G-Dot on Darwin, Block-Exp on Port, Linear/Win-Uni on NEM price, and Traj-Linear on Coastal.

Score-based local allocation outperforms uniform allocation. Against Block-Uni and Win-Uni, score-based block allocation gains 4.40 and 0.47 percentage points with Exp, and 4.23 and 1.08 points with Linear, respectively (Table 1). These ablations preserve each trajectory's total sampling mass, so the aggregate gains show the value of directing exposure toward selected local regions within a trajectory. Uniform allocation can nevertheless win on individual tasks: both Exp-Uni ablations outperform Block-Exp on NEM demand, Exp/Win-Uni wins on Solar, and Linear/Win-Uni wins on NEM price.

Local shape matching outperforms block and trajectory means in aggregate. With Exp, Linear, and ZeroClip aggregation, retaining the local gradient shape gives higher mean MASE improvement than using only a block mean or a trajectory mean (Table 1). With Linear aggregation, RAGE improves the Rand-relative mean by 0.98 percentage points over Block-Linear and 3.43 points over Traj-Linear, with lower MASE in 25 and 32 of 42 paired cells, respectively. Shape-Max improves over Traj-Max by 0.63 points but trails Block-Max by 0.14 points. Block means also outperform trajectory means by 1.10 points with Exp and 2.44 points with Linear. Together, these comparisons support scoring local regions and retaining their gradient slope with the summed aggregation rules.

Linear aggregation gives the strongest shape-based results. RAGE yields $+2.03\%$ improvement over Rand. Replacing its linear aggregation with Exp, ZeroClip, or Max yields $+1.01\%$, $+0.89\%$, and -0.79%, respectively (Table 1). Linear also leads the block mean ablations at $+1.05\%$, while Exp leads the trajectory-level ablations at -1.05%. Max trails all three summed rules for shapes and blocks, so a single strongest reference is less effective at these granularities.

Computational cost under a sparse gradient budget. Table 2 separates gradient preparation, selection, post-training, and output files. Shape and Block reuse the same sparse gradients. To estimate RAGE selection time, we charge it the entire recorded workflow that also computes three alternative rules for combining reference similarities: Exp, ZeroClip, and Max. This conservative proxy averages 65.8 seconds, versus 131.8 for TSDS and 178.9 for LP-Mirror: approximately 50% and 63% less selection time, respectively. Shared gradient preparation is counted separately. RAGE output files average 2.54 MiB versus 81.00 MiB for LP-Mirror and 159.31 MiB for G-Dot, excluding gradient/feature caches. These are stagewise estimates; cache state and hardware differences prevent an end-to-end speedup claim.

Table 2: **Resource costs.** RAGE uses a conservative selection-time estimate; other entries are recorded stage costs.

(a) Mean method costs

Method	Select (s)	Train (s)	File (MiB)
Cells	36	42	42
RAGE	65.8*	114.0	2.54
RAGE/Block-Linear	1.6	145.7	2.37
RAGE/Traj-Linear	0.6	147.6	0.11
Rand	0.5	144.3	0.08
LESS-TS	9.3	146.4	0.12
TSDS	131.8	147.1	0.11
LP-Mirror	178.9	148.1	81.00
G-Dot	46.2	143.5	159.31
GraNd	25.2	145.1	157.28

(b) Shared stages by model

Model	Prep. (h)	Joint select (s)
Summary	Total, $n = 12$	Mean, $n = 12$
Chronos-Base	15.00	58.4
TimesFM 2.5	9.13	62.0
TimerXL	3.66	76.9

Mean reusable gradient counts per setting: 136,080 source and 11,889 reference window backpropagations ($n = 36$). Shape and Block share these gradients; slope scoring adds no window-gradient backpropagation.

*Estimated RAGE selection cost: the entire measured workflow is charged to RAGE, including its Linear rule and the Exp, ZeroClip, and Max ablations. These rules combine the same source–reference shape similarities into block scores. Panel (b) reports the same joint selection time by model. Preparation totals are shared wall-clock receipts, not GPU-hours. Preparation/selection cover six tasks (Coastal has no timer); selection includes saving/warm loading and shared-work attribution. Training is recorded through epoch 2, excludes evaluation, and is affected by resumed runs and hardware/concurrency differences. File sizes exclude gradient/feature caches and raw data. Appendix I gives the accounting details.

5 Conclusion

RAGE estimates gradients at sparse adaptive anchors, matches local gradient shapes to task references, and converts score ranks into source-window sampling probabilities. Ablations support local scoring, gradient-slope features, and score-based allocation within trajectories. Across seven tasks and three models, RAGE reduces MASE by 2.03% on average relative to random resampling and by 2.37% relative to LESS-TS under a nominal 1.5% gradient budget and the same post-training exposure budget. It leads the overall and all three model-level aggregates, while individual tasks favor different candidates. Slope features reuse the block-mean anchors without additional window-gradient backpropagation.

Ethics Statement

This work uses time-series datasets and pretrained forecasting models for research evaluation. We do not recruit human participants or collect new personal data. The experiments use forecasting observations to study model adaptation and do not make decisions about individual people.

Reproducibility Statement

An anonymous repository for this work is available at https://anonymous.4open.science/r/RAGE-iclr2027/.
Section 4.1 describes the tasks, data splits, models, training recipes, and sampling settings; the method and appendix specify the gradient estimation and scoring procedures.

AI Use Statement

Generative AI tools were used to assist and polish paper writing. The authors manually checked the AI-assisted work and take responsibility for the final text, claims, and artifacts. Chronos-Base (Ansari et al., 2024), TimesFM 2.5 (Das et al., 2024), and Timer-XL (Liu et al., 2025) are used as the time-series foundation models in our post-training experiments.

References

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *CoRR*, abs/2410.10393, 2024. doi: 10.48550/ARXIV.2410.10393. URL https://doi.org/10.48550/arXiv.2410.10393.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=XfHWcNTSHp.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda-Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *CoRR*, abs/2510.15821, 2025. doi: 10.48550/ARXIV.2510.15821. URL https://doi.org/10.48550/arXiv.2510.15821.

Xiang Ao and Mengru Chen. Efficient data selection for time series forecasting using a lightweight linear proxy framework. *Scientific Reports*, 2026. URL https://www.nature.com/articles/s41598-026-43645-y.

James Z. Chang, Jan P. Allebach, and Charles A. Bouman. Sequential linear interpolation of multidimensional functions. *IEEE Trans. Image Process.*, 6(9):1231–1245, 1997. doi: 10.1109/83.623187. URL https://doi.org/10.1109/83.623187.

Sirui Chen, Yunzhe Qi, Mengting Ai, Yifan Sun, Ruizhong Qiu, Jiaru Zou, and Jingrui He. Influence-preserving proxies for gradient-based data selection in LLM fine-tuning. *CoRR*, abs/2602.17835, 2026. doi: 10.48550/ARXIV.2602.17835. URL https://doi.org/10.48550/arXiv.2602.17835.

Karel Crombecq, Dirk Gorissen, Dirk Deschrijver, and Tom Dhaene. A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. *SIAM J. Sci. Comput.*, 33(4): 1948–1974, 2011. doi: 10.1137/090761811. URL https://doi.org/10.1137/090761811.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Matthew Faw, Rajat Sen, Yichen Zhou, and Abhimanyu Das. In-context fine-tuning for time-series foundation models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/faw25b.html.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/eddea82ad2755b24c4e168c5fc2ebd40-Abstract-round2.html.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Divij Gupta, Anubhav Bhatti, Suraj Parmar, Chen Dan, Yuwei Liu, Bingjie Shen, and San Lee. Low-rank adaptation of time series foundational models for out-of-domain modality forecasting. In Hayley Hung, Catharine Oertel, Mohammad Soleymani, Theodora Chaspari, Hamdi Dibeklioglu, Jainendra Shukla, and Khiet P. Truong (eds.), *Proceedings of the 26th International Conference on Multimodal Interaction, ICMI 2024, San Jose, Costa Rica, November 4-8, 2024*, pp. 382–386. ACM, 2024. doi: 10.1145/3678957.3685724. URL https://doi.org/10.1145/3678957.3685724.

John D. Jakeman and Stephen G. Roberts. Local and dimension adaptive sparse grid interpolation and quadrature. *CoRR*, abs/1110.0010, 2011. URL http://arxiv.org/abs/1110.0010.

Junguang Jiang, Baixu Chen, Junwei Pan, Ximei Wang, Dapeng Liu, Jie Jiang, and Mingsheng Long. Forkmerge: Mitigating negative transfer in auxiliary-task learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/60f9118a849e8e9a0c67e2a36ad80ebf-Abstract-Conference.html.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, Chenghao Liu, Ameet Talwalkar, and David Asker. Toto 2.0: Time series forecasting enters the scaling era. *CoRR*, abs/2605.20119, 2026. doi: 10.48550/ARXIV.2605.20119. URL https://doi.org/10.48550/arXiv.2605.20119.

KrishnaTeja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh K. Iyer. GRAD-MATCH: gradient matching based data subset selection for efficient deep model training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 2021. URL http://proceedings.mlr.press/v139/killamsetty21a.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceed-*

ings of Machine Learning Research, pp. 1885–1894. PMLR, 2017. URL http://proceedings.mlr.press/v70/koh17a.html.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=9m02ib92Wz.

Thomas L. Lee, Edoardo M. Ponti, and Amos J. Storkey. Adapting time series foundation models through data mixtures. *CoRR*, abs/2603.02840, 2026. doi: 10.48550/ARXIV.2603.02840. URL https://doi.org/10.48550/arXiv.2603.02840.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 18878–18890, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Wenqin Liu, Weizhi Quan, Aoqi Zuo, Erdun Gao, Vu Nguyen, Dino Sejdinovic, Howard D. Bondell, and Mingming Gong. Timelava: Learning-agnostic valuation for time series data. *CoRR*, abs/2606.18729, 2026a. doi: 10.48550/ARXIV.2606.18729. URL https://doi.org/10.48550/arXiv.2606.18729.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=KMCJXjlDDr.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025b. URL https://proceedings.mlr.press/v267/liu25be.html.

Yong Liu, Xingjian Su, Shiyu Wang, Haoran Zhang, Haixuan Liu, Yuxuan Wang, Zhou Ye, Yang Xiang, Jianmin Wang, and Mingsheng Long. Timer-s1: A billion-scale time series foundation model with serial scaling. *CoRR*, abs/2603.04791, 2026b. doi: 10.48550/ARXIV.2603.04791. URL https://doi.org/10.48550/arXiv.2603.04791.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. TSDS: data selection for task-specific model finetuning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/13848b5893119ff772b69812c95914fa-Abstract-Conference.html.

Guanghui Min, Tianhao Huang, Ke Wan, and Chen Chen. GIST: targeted data selection for instruction tuning via coupled optimization geometry. *CoRR*, abs/2602.18584, 2026. doi: 10.48550/ARXIV.2602.18584. URL https://doi.org/10.48550/arXiv.2602.18584.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446. PMLR, 2022. URL https://proceedings.mlr.press/v162/navon22a.html.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Mahdi Nikdan, Vincent Cohen-Addad, Dan Alistarh, and Vahab Mirrokni. Efficient data selection at scale via influence distillation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván

Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/d2d4f6858cc2d21fd0230244fcb34f1d-Abstract-Conference.html.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: attributing model behavior at scale. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's TIME: towards the next generation of time series forecasting benchmarks. *CoRR*, abs/2602.12147, 2026. doi: 10.48550/ARXIV.2602.12147. URL https://doi.org/10.48550/arXiv.2602.12147.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Ege Onur Taga, Halil Alperen Gozeten, Kutay Tire, Rahul Dalvi, Reinhard Heckel, and Samet Oymak. Filter, augment, forecast: Online data selection for robust time series forecasting. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300, pp. 1081–1089. PMLR, 2026. URL https://proceedings.mlr.press/v300/taga26a.html.

Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=HD6bWcj87Y.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52915–52971. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/wettig24a.html.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Gholamreza Haffari. The best of both worlds: Bridging quality and diversity in data selection with bipartite graph. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/wu25ac.html.

Shunyu Wu, Tianyue Li, Yixuan Leng, Jingyi Suo, Jian Lou, Dan Li, and See-Kiong Ng. Lightweight time series data valuation on time series foundation models via in-context finetuning. In Hyungsoo Jung, Tianzheng Wang, Masashi Toyoda, Hyuk-Yoon Kwon, and Jae-woong Lee (eds.), *Database Systems for Advanced Applications - 31st International Conference, DASFAA 2026, Jeju, South Korea, April 27-30, 2026, Proceedings, Part III*, volume 16537 of *Lecture Notes in Computer*

Science, pp. 678–694. Springer, 2026. doi: 10.1007/978-981-92-0369-7_42. URL https://doi.org/10.1007/978-981-92-0369-7_42.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: selecting influential data for targeted instruction tuning. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Yizi Zhang, Jingyan Shen, Xiaoxue Xiong, and Yongchan Kwon. Timeinf: Time series data contribution via influence functions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Vz0CWFMPUe.

A Overall, Model-Level, and Per-Seed Results

Table 3 summarizes RAGE's paired percentage MASE improvements over Rand across the seven tasks and by model. Each model has 14 paired cells (seven tasks, two seeds), with the same Shape+Linear configuration throughout. Table 1 reports the two-seed method–task means for all 23 methods and ablations.

Table 3: **Overall and model-level results across seven tasks.** Mean and median are paired percentage MASE reductions against Rand; positive values favor RAGE. The same RAGE configuration is used in every row.

Model	Pairs	Mean (%)	Median (%)
All seven tasks	42	+2.03	+1.32
Chronos-Base	14	+1.93	+0.15
TimesFM 2.5	14	+1.79	+1.32
TimerXL	14	+2.38	+3.43

Tables 4 and 5 report the individual seed results using the same layout, method order, and ranking colors as Table 1. Each seed contains 21 model–task cells; Avg. and Wins are computed within that seed against its matched Rand results.

Table 4: **All forecasting methods and ablations across seven tasks: seed 0.** Entries are MASE for seed 0 (lower is better). Avg. is the mean paired percentage improvement over Rand across 21 model–task cells; Wins counts positive improvements. Darker blue marks ranks 1–5 among trained candidates; Orig is unranked.

Method	Azure			Darwin			NEM-D			NEM-P			Port			Solar			Coastal			Avg.	Wins
	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	(%)	/21
RAGE	1.2351	0.7645	**1.0924**	1.6568	1.0264	1.1896	1.3374	0.6984	0.9180	1.0491	0.7192	0.8855	0.8937	0.7297	0.7731	**1.0446**	1.1067	1.2760	2.0598	1.6220	1.9246	**+2.53**	**14**
S-Exp	1.2493	0.7679	1.1451	1.6567	1.0190	1.1974	1.3705	0.7074	0.9659	1.0525	0.7159	0.8928	0.8966	0.7300	0.7721	1.2397	1.1141	1.3198	2.0843	1.6277	1.8864	+0.86	12
S-ZC	**1.1421**	0.7643	1.1558	1.6512	1.0237	1.2151	1.4490	0.7179	0.9668	1.0462	0.7263	0.9062	0.8917	0.7298	0.7921	1.0738	1.1192	1.2865	2.0812	1.6263	1.9285	+1.29	13
S-Max	1.2980	0.7625	1.2264	1.6637	1.0315	1.2582	1.4546	0.7364	0.9496	1.0531	0.7328	0.8515	0.8887	0.7274	0.7870	1.2281	1.1228	1.3315	2.0887	1.6370	1.9196	-0.37	9
Orig	1.3318	0.7638	1.2847	1.6639	1.0697	1.3571	1.7385	0.8819	0.9168	1.0622	0.7492	0.8504	0.8821	0.7162	0.7978	1.2167	1.1293	1.3442	2.0731	1.6595	2.0427	-3.41	6
Rand	1.2331	**0.7614**	1.1673	1.6489	1.0395	1.2560	1.4405	0.7788	0.8894	1.0637	0.7302	0.8705	0.8949	0.7214	0.7948	1.2313	1.1201	1.3708	2.0707	1.6450	1.9031	+0.00	0
LESS-TS	1.2274	0.7696	1.2083	**1.6397**	1.0269	1.2306	1.4426	0.7343	0.9758	1.0922	0.7210	0.8642	0.8835	0.7278	0.7965	1.1850	1.1258	1.3702	2.0807	1.6303	1.9043	-0.06	11
TSDS	1.2753	0.7700	1.1713	1.6627	1.0414	1.3389	1.4302	0.6960	0.9423	1.0497	**0.7153**	0.9738	0.9430	0.7396	0.7938	1.3827	1.1093	**1.2605**	2.0471	1.6524	1.9265	-1.29	8
LP-Mir	1.4496	0.7756	1.1731	1.7154	1.0818	1.3655	3.1029	0.8964	0.9287	1.1285	0.7570	0.9298	0.9050	0.7211	0.7931	1.6814	1.1360	1.2715	2.9369	1.6462	2.2692	-13.38	3
G-Dot	1.2219	0.7664	1.4267	1.6542	**1.0121**	**1.1749**	**1.2755**	**0.6746**	0.9136	1.0636	0.7262	0.9850	0.8994	0.7267	0.7999	1.1867	1.1228	1.6695	2.0735	1.6261	1.9098	-1.11	9
GraNd	1.2230	0.7662	1.3355	1.6629	1.0375	1.2825	1.3201	0.7710	1.0598	1.0425	0.7354	0.9639	0.9221	**0.7192**	0.8266	1.0850	1.1428	1.4552	2.0569	1.6301	1.9446	-1.90	9
B-Lin	1.2461	0.7675	1.1062	1.6558	1.0205	1.1868	1.3885	0.7207	**0.8854**	**1.0329**	0.7172	0.9281	0.8946	0.7300	**0.7696**	1.1754	1.1373	1.2829	2.0808	1.6204	1.9177	+1.50	13
B-Exp	1.2267	0.7663	1.2357	1.6430	1.0171	1.2379	1.3650	0.7168	1.0105	1.0683	0.7162	0.8691	0.8824	0.7288	0.7763	1.2266	1.1383	1.3205	2.0741	**1.6197**	1.9166	+0.24	13
B-ZC	1.2372	0.7705	1.1749	1.6694	1.0276	1.2228	1.3683	0.7159	0.8861	1.0571	0.7240	0.8933	0.8883	0.7266	0.7935	1.3082	1.1260	1.3327	2.0685	1.6311	1.8889	+0.50	13
B-Max	1.2448	0.7623	1.1533	1.6628	1.0353	1.2735	1.4707	0.7306	0.9616	1.0689	0.7316	0.9035	0.8935	0.7273	0.7889	1.4198	1.1472	1.2985	2.0745	1.6403	1.8971	-1.05	8
T-Lin	1.2009	0.7709	1.2741	1.6702	1.0361	1.2690	1.3528	0.7335	1.0059	1.0784	0.7167	0.9800	0.8989	0.7283	0.7992	1.2842	1.1324	1.3829	**2.0392**	1.6363	**1.8803**	-1.34	8
T-Exp	1.2297	0.7685	1.2572	1.6589	1.0404	1.2549	1.4232	0.7272	0.9840	1.0874	0.7169	0.9518	0.8904	0.7276	0.7928	1.2842	1.1324	1.3829	2.0693	1.6242	1.8911	-1.23	10
T-ZC	1.2384	0.7642	1.2057	1.6667	1.0343	1.2655	1.4973	0.7381	0.9634	1.0825	0.7182	0.9339	0.8876	0.7300	0.7880	1.2353	1.1531	1.3755	2.0976	1.6334	1.8912	-1.09	7
T-Max	1.2309	0.7692	1.1915	1.6448	1.0345	1.2478	1.5274	0.7443	0.9943	1.0598	0.7191	0.9814	0.8933	0.7270	0.8006	1.2842	1.1324	1.3829	2.0442	1.6374	1.8919	-1.48	11
B-Lin/BU	1.2498	0.7628	1.4624	1.6444	1.0444	1.3201	1.2917	0.7867	0.8968	1.0395	0.7238	0.9301	0.8917	0.7238	0.7977	1.2299	**1.1055**	1.5163	2.0611	1.6292	2.0836	-2.14	9
B-Exp/BU	1.2814	0.7650	1.6271	1.6452	1.0368	1.5167	1.3016	0.7593	0.9211	1.0575	0.7236	0.8632	**0.8823**	0.7246	0.8045	1.1568	1.1299	1.7279	2.0837	1.6298	2.0774	-3.96	10
B-Lin/WU	1.2013	0.7666	1.2259	1.6577	1.0434	1.2815	1.2787	0.7526	0.9210	1.0502	0.7265	0.8602	0.8928	0.7253	0.7888	1.2789	1.1383	1.4032	2.0614	1.6324	1.8824	+0.14	11
B-Exp/WU	1.2132	0.7631	1.2308	1.6591	1.0401	1.2592	1.3618	0.7465	0.9688	1.0465	0.7245	**0.8493**	0.8904	0.7248	0.7957	1.2756	1.1265	1.2755	2.0672	1.6390	1.9203	+0.14	10

Ch=Chronos-Base; TF=TimesFM 2.5; TX=TimerXL. RAGE uses Shape+Linear. S/B/T=Shape/Block/Trajectory; Lin=Linear; ZC=ZeroClip; BU=Block-Uni; WU=Win-Uni. All entries use epoch 2; each method has 21 measured cells.

Table 5: **All forecasting methods and ablations across seven tasks: seed 1.** Entries are MASE for seed 1 (lower is better). Avg. is the mean paired percentage improvement over Rand across 21 model–task cells; Wins counts positive improvements. Darker blue marks ranks 1–5 among trained candidates; Orig is unranked.

Method	Azure			Darwin			NEM-D			NEM-P			Port			Solar			Coastal			Avg.	Wins
	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	Ch	TF	TX	(%)	/21
RAGE	1.2296	0.7634	1.1098	1.6554	**1.0156**	1.2508	**1.2241**	0.7011	0.9114	1.0566	0.7149	0.8826	0.8832	0.7277	**0.7671**	1.2109	1.1110	1.3334	2.0865	1.6230	**1.9163**	**+1.54**	**14**
S-Exp	1.2576	0.7615	1.1047	1.6758	1.0268	**1.2443**	1.4113	0.7118	0.9193	1.0610	0.7231	0.8785	0.8798	0.7296	0.7812	**1.0298**	1.1437	1.2941	2.0844	1.6333	1.9296	+1.16	12
S-ZC	1.2578	0.7625	1.1440	1.6488	1.0179	1.2860	1.3119	0.7155	0.9312	1.0857	0.7217	0.9147	0.8845	0.7270	0.7927	1.1159	1.1299	1.2981	2.0825	1.6327	1.9611	+0.48	12
S-Max	1.2516	0.7636	**1.0797**	1.6530	1.0402	1.2767	1.3893	0.7487	0.9045	1.0461	0.7267	0.9223	0.8872	0.7259	0.8134	1.4919	1.1364	1.3000	2.0543	1.6349	1.9657	-1.21	8
Orig	1.3318	0.7638	1.2847	1.6639	1.0697	1.3571	1.7385	0.8819	0.9168	1.0622	0.7492	0.8504	0.8821	0.7162	0.7978	1.2167	1.1293	1.3442	2.0731	1.6595	2.0427	-3.96	6
Rand	1.2159	0.7611	1.1347	1.6566	1.0436	1.3567	1.3323	0.7434	0.9747	1.0729	0.7249	0.8725	0.8846	0.7216	0.8001	1.1751	1.1182	**1.2580**	**2.0390**	1.6496	2.0091	+0.00	0
LESS-TS	1.2270	0.7633	1.1872	1.6647	1.0413	1.2552	1.2609	0.7393	0.9879	1.0817	0.7196	0.8983	0.8821	0.7259	0.8077	1.2305	1.1335	1.3704	2.0814	1.6401	1.9832	-0.65	8
TSDS	1.2482	0.7635	1.1909	1.6446	1.0395	1.3938	1.2751	0.7255	0.9023	1.0570	0.7294	0.9015	0.8976	0.7380	0.7924	1.1509	1.1207	1.3543	2.0429	1.6442	1.9403	-0.14	10
LP-Mir	1.2775	0.7649	1.1318	1.6792	1.0603	1.4438	1.3650	0.7518	0.9640	1.0548	0.7299	0.8578	0.8742	0.7205	0.7951	1.3398	1.1188	1.3380	2.2641	1.6584	2.0254	-2.16	7
G-Dot	1.2442	0.7612	1.2438	1.6512	1.0196	1.2613	1.3824	0.7128	0.9110	1.0595	0.7241	0.9015	0.8916	0.7267	0.7875	1.3447	1.1197	1.3286	2.0720	1.6377	1.9286	-0.69	10
GraNd	1.2134	0.7598	1.2852	1.6541	1.0386	1.3897	1.2698	0.7308	1.0115	**1.0417**	0.7218	0.9890	0.8842	**0.7199**	0.8215	1.2896	1.1237	1.3368	2.1006	1.6430	2.0384	-2.15	11
B-Lin	1.2585	0.7601	1.1849	**1.6430**	1.0243	1.2504	1.3290	0.7224	0.9021	1.0536	0.7154	0.8700	0.8874	0.7276	0.7791	1.1553	1.1376	1.3650	2.0913	1.6239	1.9274	+0.60	**14**
B-Exp	1.2517	0.7630	1.1114	1.6481	1.0235	1.2574	1.4829	0.7152	0.9738	1.0521	0.7170	0.8957	0.8780	0.7295	0.7745	1.2372	**1.1106**	1.3386	2.0857	1.6220	1.9214	-0.14	13
B-ZC	1.2412	0.7621	1.1227	1.6506	1.0249	1.2724	1.2839	0.7132	0.9432	1.0469	0.7175	0.9058	0.8794	0.7264	0.7967	1.2797	1.1233	1.3281	2.0499	1.6390	1.9610	+0.27	13
B-Max	1.2478	0.7624	1.1573	1.6543	1.0291	1.3071	1.3144	0.7507	0.9477	1.0602	0.7294	0.8633	0.8922	0.7254	0.7995	1.1826	1.1203	1.3999	2.0749	1.6353	1.9314	-0.25	10
T-Lin	1.2623	0.7615	1.2575	1.6553	1.0322	1.2711	1.3943	0.7283	1.0058	1.0824	0.7144	0.9306	0.8903	0.7254	0.8055	1.2278	1.1381	1.3378	2.0601	1.6374	1.9378	-1.44	7
T-Exp	1.1974	0.7617	1.1637	1.6648	1.0342	1.2864	1.3862	0.7235	1.0118	1.0759	0.7136	0.9427	0.8888	0.7258	0.8051	1.2278	1.1381	1.3378	2.0724	1.6333	1.9218	-0.86	7
T-ZC	1.2225	**0.7546**	1.1926	1.6665	1.0385	1.2847	1.4725	**0.6969**	1.0915	1.0732	0.7209	0.9446	**0.8671**	0.7253	0.7961	1.2326	1.1361	1.3397	2.0403	1.6346	1.9891	-1.56	9
T-Max	**1.1714**	0.7615	1.2129	1.6461	1.0355	1.2942	1.3656	0.7218	1.0923	1.0738	**0.7092**	0.9479	0.8839	0.7260	0.8186	1.2278	1.1381	1.3378	2.0627	1.6323	1.9575	-1.36	9
B-Lin/BU	1.2452	0.7637	1.4381	1.6668	1.0374	1.4477	1.4116	0.7316	0.9527	1.0530	0.7262	1.0352	0.8819	0.7227	0.8234	1.1197	1.1438	1.5340	2.0878	**1.6178**	2.2171	-4.21	7
B-Exp/BU	1.2365	0.7634	1.3713	1.6565	1.0469	1.4510	1.2437	0.7656	**0.8987**	1.0569	0.7221	1.0602	0.9021	0.7222	0.8346	1.3215	1.1498	1.6131	2.0646	1.6220	2.2444	-4.74	6
B-Lin/WU	1.2461	0.7615	1.1911	1.6727	1.0398	1.3017	1.3134	0.7167	1.0016	1.0702	0.7230	**0.8513**	0.8929	0.7225	0.8036	1.1954	1.1257	1.2855	2.0699	1.6450	1.9688	-0.20	9
B-Exp/WU	1.1944	0.7586	1.2686	1.6541	1.0353	1.3438	1.3216	0.7400	0.9713	1.0521	0.7250	0.9764	0.8890	0.7226	0.7991	1.1429	1.1120	1.3203	2.0753	1.6397	2.0249	-0.97	13

Ch=Chronos-Base; TF=TimesFM 2.5; TX=TimerXL. RAGE uses Shape+Linear. S/B/T=Shape/Block/Trajectory; Lin=Linear; ZC=ZeroClip; BU=Block-Uni; WU=Win-Uni. All entries use epoch 2; each method has 21 measured cells.

B First-Order Motivation and Gradient-Shape Similarity

Assume that $\mathcal{L}_{\text{tar}}$ is twice differentiable near θ_0. Let $G_{\text{tar}} = \nabla_\theta \mathcal{L}_{\text{tar}}(\theta_0)$ be its gradient, let $G_{i,t} = \nabla_\theta \ell(\theta_0; z_{i,t})$ be the gradient of source event $z_{i,t}$, and let $\eta > 0$ be a small update step size. Taylor's theorem gives

$$\mathcal{L}_{\text{tar}}(\theta_0 - \eta G_{i,t}) = \mathcal{L}_{\text{tar}}(\theta_0) - \eta G_{\text{tar}}^\top G_{i,t} + \frac{\eta^2}{2} G_{i,t}^\top H_{\text{tar}}(\theta_0 - \tau_{i,t}\eta G_{i,t}) G_{i,t}, \tag{9}$$

where $H_{\text{tar}}(\theta) = \nabla_\theta^2 \mathcal{L}_{\text{tar}}(\theta)$ is the target-loss Hessian and $\tau_{i,t} \in (0,1)$ selects a point along the update. Averaging over a source event drawn from p yields

$$\begin{aligned}\mathbb{E}_{(i,t)\sim p}[\mathcal{L}_{\text{tar}}(\theta_0 - \eta G_{i,t})] = \mathcal{L}_{\text{tar}}(\theta_0) - \eta G_{\text{tar}}^\top \sum_{(i,t)\in\mathcal{W}} p_{i,t} G_{i,t} \\ + \frac{\eta^2}{2} \sum_{(i,t)\in\mathcal{W}} p_{i,t} G_{i,t}^\top H_{\text{tar}}(\theta_0 - \tau_{i,t}\eta G_{i,t}) G_{i,t}.\end{aligned} \tag{10}$$

Let $\widehat{\mathcal{L}}_{\mathcal{R}}$ be an empirical estimate of target forecasting loss from the reference set and $\widehat{G}_{\mathcal{R}} = \nabla_\theta \widehat{\mathcal{L}}_{\mathcal{R}}(\theta_0)$ its gradient. Replacing G_{tar} by a reference gradient estimate identifies the dot product in Eq. (2) as a first-order score for a source event. Cosine additionally normalizes both gradients and measures directional agreement. RAGE compares the normalized reconstructed gradient functions of two blocks through $z_r^\top z_j$ (Eq. (7)). The shape criterion retains midpoint and slope agreement, as derived in Appendix C; it is not a Taylor loss-change estimate for a single window update. The Taylor motivation also omits the Hessian term, and multi-step training changes the gradients as model parameters evolve.

C CRB Gradient Interpolation and Features

Estimate between anchors. An anchor is a legal forecast start with a computed gradient. For neighboring anchors $a < b$, CRB estimates every legal start t between them by linear interpolation:

$$\widehat{g}_i(t) = \frac{b-t}{b-a} g_i(a) + \frac{t-a}{b-a} g_i(b), \qquad a \le t \le b. \tag{11}$$

Here $\widehat{g}_i(t)$ is the estimated projected gradient; it equals the observed gradient at either endpoint and requires no new backpropagation at unobserved starts. Thus, CRB returns a budgeted anchor set and a local gradient estimate across each trajectory.

Projected gradient features. At each selected start on a source or reference trajectory, CRB obtains the per-window gradient by backpropagating once at the fixed pretrained state θ_0. Applying the fixed projection Π adds no window-gradient computation. CRB retains the projected gradient's raw magnitude for interpolation, risk estimation, and cancellation testing. The midpoint and slope of each reconstructed block are normalized together as in Eq. (7) before shape scoring.

We implement Π as CountSketch over the concatenated gradients of all trainable parameters, with shared parameters included once. For parameter coordinate k, draw a bucket h_k uniformly from $\{1, \dots, d\}$ and a sign s_k uniformly from $\{-1, +1\}$. The projected coordinate is

$$[\Pi v]_b = \sum_{k:h_k=b} s_k v_k, \qquad b = 1, \dots, d. \tag{12}$$

The implementation uses NumPy's `default_rng(42)` to draw all bucket assignments followed by all signs in the model's parameter order, with $d = 4096$. Parameter-tensor contributions are accumulated in FP32 into the same sketch, without an additional projection scale factor or per-window unit normalization. For each model, the bucket and sign assignments remain fixed across source and reference windows, tasks, and training seeds. RAGE, LESS-TS, G-Dot, and GraNd-Static use this same full-parameter projection at the pretrained checkpoint listed in Appendix G.

Energy-normalized shape representation. For a reconstructed block line $g_j(u) = \bar{g}_j + u d_j$ on $u \in [-\frac{1}{2}, \frac{1}{2}]$, the constant and linear components are orthogonal under continuous integration:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} u \, \mathrm{d}u = 0, \qquad \int_{-\frac{1}{2}}^{\frac{1}{2}} u^2 \, \mathrm{d}u = \frac{1}{12}.$$

Expanding the squared norm and integrating therefore gives

$$\begin{aligned} E_j &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \|\bar{g}_j + u d_j\|_2^2 \, \mathrm{d}u \\ &= \|\bar{g}_j\|_2^2 + \frac{\|d_j\|_2^2}{12} = \left\| \left[\bar{g}_j;\ \frac{d_j}{\sqrt{12}} \right] \right\|_2^2. \end{aligned}$$

Thus, the factor $\frac{1}{\sqrt{12}}$ places the slope component in the same energy geometry as the midpoint component. Dividing the concatenated vector by $\sqrt{E_j}$ yields Eq. (7). For two nonzero-energy blocks, their shape inner product also equals the integrated inner product of the normalized reconstructed lines:

$$z_r^\top z_j = \frac{\bar{g}_r^\top \bar{g}_j + \frac{d_r^\top d_j}{12}}{\sqrt{E_r E_j}} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{g_r(u)}{\sqrt{E_r}} \right)^\top \left(\frac{g_j(u)}{\sqrt{E_j}} \right) \mathrm{d}u.$$

This identity uses continuous integration on a unit interval, rather than an exact average over a block's discrete legal starts. Block size enters reference weighting and source exposure separately.

D Quadratic Approximation for Anchor-Risk Priority

Consider an interval $I = [a, b]$ with computed projected gradients at its endpoints and at an interior legal start m. Let $h_I = b - a$, and let $\widetilde{g}_I(\xi)$ be the linear interpolation of $g_i(a)$ and $g_i(b)$ at position $\xi \in [a, b]$; at legal starts, it equals $\widehat{g}_i$ in Eq. (11). Fit a vector-valued quadratic $Q_I(\xi)$ through the three observed gradients. Its second derivative is a constant vector w_I, and the quadratic and linear interpolants agree at both endpoints. Their difference is therefore

$$Q_I(\xi) - \widetilde{g}_I(\xi) = \frac{w_I}{2}(\xi - a)(\xi - b). \tag{13}$$

At the computed midpoint m, the norm of this difference is $\delta_I = \frac{\|w_I\|(m-a)(b-m)}{2}$. Solving for the curvature magnitude gives $\|w_I\| = \frac{2\delta_I}{(m-a)(b-m)} = \widehat{c}_I$, which is Eq. (4). The fitted quadratic's

squared error against endpoint interpolation, integrated over the interval, is

$$\int_a^b \|Q_I(\xi) - \widetilde{g}_I(\xi)\|^2 \,\mathrm{d}\xi = \frac{\|w_I\|^2}{4} \int_a^b (\xi - a)^2 (b - \xi)^2 \,\mathrm{d}\xi = \frac{\widehat{c}_I^2}{4} \frac{h_I^5}{30} = \frac{h_I^5 \widehat{c}_I^2}{120}, \tag{14}$$

which gives the priority in Eq. (5). The equality is exact for the fitted quadratic. It approximates the unknown gradient path between the observed starts, and the integral is a continuous proxy for error over discrete legal starts. An active child interval uses the curvature estimate inherited from its split parent until that child is itself split.

E End-to-End RAGE Procedure

Algorithm 1: RAGE: from local gradient shapes to window probabilities and post-training

Input: Source $\mathcal{S}$, reference $\mathcal{R}$, model θ_0, protocol (C, H, D), projection Π, densities $\rho = (\rho_{\text{cov}}, \rho_{\text{err}}, \rho_{\text{split}})$, rank temperature T, draws B
Output: Source blocks $\mathcal{J}$, block masses P, window distribution p, model θ_B

for *each trajectory* i *in* $\mathcal{S} \cup \mathcal{R}$ **do**
  Enumerate legal starts $\mathcal{T}_i$ and compute stage caps by Eq. (3);
  Obtain anchor map A_i from CRB (Algorithm 2);
  Build neighboring-anchor blocks; compute sizes m_j and joint midpoint–slope shapes z_j by Eq. (7);
Weight reference blocks by α_r as in Section 3.3;
for *each source block* j **do**
  $s_j \leftarrow \sum_r \alpha_r z_r^\top z_j$;
Rank s_j and share exact-tie rank mass to obtain u_j using Eq. (8);
$P_j \leftarrow \frac{m_j u_j}{\sum_q m_q u_q}$; set $p_{i,t} \leftarrow \frac{P_j}{m_j}$ for $t \in j$;
$\theta_B \leftarrow \text{Train}_B(\theta_0, p)$, drawing $j \sim P$ then $t \sim \text{Uniform}(j)$;
return $\mathcal{J}, P, p, \theta_B$;

F CRB Anchor-Selection Procedure

Algorithm 2: CRB: strict-budget adaptive gradient anchors

Input: Trajectory i, ordered legal starts $\mathcal{T}_i$, stage caps from Eq. (3), gradient routine $g_i(t)$
Output: Anchor map A_i of selected starts and computed gradients

if $M_i = 0$ **then**
  return $\varnothing$;
Let $t_{\text{min}}, t_{\text{max}}$ be the first and last starts in $\mathcal{T}_i$;
Compute $g_i(t_{\text{min}})$; $A_i \leftarrow \{t_{\text{min}} \mapsto g_i(t_{\text{min}})\}$;
if $M_i = 1$ **then**
  return A_i;
Compute $g_i(t_{\text{max}})$; add $t_{\text{max}} \mapsto g_i(t_{\text{max}})$ to A_i;
$(n_{\text{cov}}, n_{\text{err}}, n_{\text{split}}) \leftarrow (2, 0, 0)$;
for $q \in \{\text{cov}, \text{err}, \text{split}\}$ **do**
  while $n_q < k_{i,q}$ **do**
    Select a splittable interval $I = [a, b]$: longest for coverage, largest R_I for risk, or smallest κ_I for cancellation; exact ties choose the rightmost;
    Choose the lower legal midpoint $m \in (a, b)$; compute $g_i(m)$ once and add $m \mapsto g_i(m)$ to A_i;
    Replace I by $[a, m]$ and $[m, b]$;
    if $q = \text{cov}$ *or* $q = \text{err}$ **then**
      Compute $\delta_I, \widehat{c}_I$ by Eq. (4); give both children the new curvature estimate;
    else
      Recompute both children's cancellation scores from their endpoint gradients;
    Update stage priorities; $n_q \leftarrow n_q + 1$;
return A_i;

G Task Identifiers and Post-Training Recipes

Pretrained checkpoints. All selection gradients are computed at the initial pretrained state, which also initializes post-training. The checkpoint identifiers and pinned repository revisions are:

- **Chronos-Base**: `amazon/chronos-t5-base`.
  Revision `ad294eaacead15db499b740ea4122266dd2a81a2`, with 201,374,976 parameters.
- **TimesFM 2.5**: `google/timesfm-2.5-200m-transformers`.
  Revision `5a9806b9b291fad9233b5249d88263f1846304d3`, with 231,289,280 parameters.
- **TimerXL**: `thuml/timer-base-84m`.
  Revision `70077a71acce1b4c00d98332fcaabc694255d8e5`, with 84,142,080 parameters. This is the univariate checkpoint released through the Timer-XL repository, with 96-point patches.

Orig evaluates these checkpoints directly. The main comparison reports each trained arm at the common epoch-2 checkpoint, without choosing a checkpoint by its test score.

Training configuration. Chronos-Base and TimesFM 2.5 use a $512 \to 64$ context–forecast protocol; TimerXL uses $480 \to 96$. All models use full-parameter BF16 AdamW with batch size 32 and sample source windows with replacement. The seed-specific pool has 16,284 window-equivalents; this is a nominal epoch budget, not an exact draw count. At epoch 2, the nominal exposure budget is 32,568 window draws, rounded up to $\lceil \frac{32{,}568}{32} \rceil = 1018$ full-batch optimizer steps (32,576 actual draws). Table 6 lists the recipes. AdamW uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, gradient-norm clipping at 1, and no warmup. The learning-rate schedules span six nominal epochs, including when results are read at epoch 2, and decay to zero at the end of that schedule. We report seven target tasks from six TIME benchmark dataset configurations (Qiao et al., 2026): `time_azure_i_cpu_5t` (`avg_cpu` from `azure2019_I/5T`), `time_darwin_water_15t` (`Water_Quality_Darwin/15T`), `time_nem_demand_5t` and `time_nem_price_5t` (demand and price for five regions of `OpenElectricity_NEM/5T`), `time_port_activity_d` (`Port_Activity/D`), `time_solar_h` (`Australia_Solar/H`), and `time_coastal_ts_h` (`Coastal_T_S/H`). The suffixes `5T`, `15T`, `H`, and `D` denote five-minute, 15-minute, hourly, and daily sampling.

Table 6: **Post-training recipes.**

Model	Parameters	$C \to H$	Learning rate	Weight decay	Schedule
Chronos-Base	201M	$512 \to 64$	10^{-3}	0	Linear
TimesFM 2.5	231M	$512 \to 64$	3×10^{-5}	0.01	Cosine
TimerXL	84M	$480 \to 96$	10^{-4}	0.01	Cosine

H Baseline Adaptations and Key Configurations

The baselines use the same seed-specific source pool, task reference histories, model window sizes, and post-training recipes as RAGE. Their scoring and probability construction are specified below.

Orig and Rand. Orig evaluates the pretrained checkpoints in Appendix G. Rand assigns equal probability to every legal source-window start and samples with replacement.

LESS-TS. The time-series adaptation uses one source or reference trajectory as a scoring unit. On each trajectory, it places uniformly spaced anchors, including endpoints, using the sum of the three CRB stage caps with densities $(0.005, 0.005, 0.005)$. It averages the raw projected anchor gradients before normalizing the trajectory representative. A source trajectory's score is the maximum cosine similarity to a reference trajectory, weighted by $1/N_{\mathcal{R}}$. Scores use a single pretrained checkpoint and full-parameter loss gradients; Adam gradient modulation, warmup checkpoints, and LoRA from the original LESS procedure are omitted in this adaptation. Trajectory scores use the rank-interval mapping in Section 3.3, with $T = 1/(2 \ln 4)$, followed by weighting by legal-window count and normalization. Windows are drawn uniformly within the sampled trajectory.

TSDS. TSDS represents each trajectory by the mean frozen-model context embedding at uniformly spaced window starts, using the same three stage densities to set the feature-sampling count. It applies nearest-neighbor optimal-transport assignment with KDE diversity. The experiment uses

`max_k` $= 64$, `kde_k` $= 16$, $\sigma = 4$, $\alpha = 0.1$, and $c = 5$. Both neighbor counts are capped at $\lfloor N_{\mathcal{S}}/10 \rfloor$. FAISS IVF uses L2 distance, `nlist` $= \lfloor \sqrt{N}/2 \rfloor$ for the indexed set of size N, and `nprobe` $= 10$. The resulting native OT trajectory masses are used directly as sampling probabilities, with uniform legal-window draws within each trajectory. These masses do not pass through RAGE's rank mapping.

LP-Mirror. We use the ForwardINF loss-drop score with a channel-independent linear proxy mapping C context values to H forecast values. Windows are standardized using their trajectory's mean and standard deviation; target statistics use the pre-evaluation reference history. The proxy is trained for one pass over all legal source windows using Adam, learning rate 0.01, batch size 4096, and MSE loss. It is then adapted for one epoch on reference windows using SGD, learning rate 10^{-3}, and batch size 256. Reference windows have stride 8 and are capped at 4096. Each source window is scored by its proxy MSE before adaptation minus its MSE after adaptation. The post-training probability is proportional to $\exp(s/0.05)$ over all source windows, including finite negative scores. This experiment uses that window distribution with replacement sampling in place of a fixed top-k subset.

G-Dot and GraNd-Static. For each trajectory, these baselines match RAGE's actual number of distinct anchor-gradient computations, using unique rounded uniformly spaced starts including endpoints; a one-anchor trajectory uses its leftmost start. They reconstruct uncomputed starts by piecewise linear interpolation. G-Dot averages reconstructed reference gradients over legal starts within each reference trajectory and then equally across reference trajectories, and scores each source start by the signed dot product with this mean. GraNd-Static scores each source start by its projected gradient's ℓ_2 norm and uses no reference gradients. Both rank all legal source starts using the same rank-interval mapping and $T = 1/(2 \ln 4)$, with exact ties sharing mass equally. Gradient norms and dot products are evaluated in FP64 from the stored projected vectors.

I Runtime and Storage Accounting

We account for resource costs in four stages: shared gradient preparation, scoring and selection, post-training, and output storage. Table 7 pairs seven-task quality with the recorded costs. Quality, epoch-2 training time, and selection-file size cover 42 seed–model–task cells per method. Preparation receipts and selection timers cover 36 cells across Azure, Darwin, NEM demand, NEM price, Port, and Solar; Coastal has no preparation or selection timer in this archive. Each column states its count rather than treating missing costs as zero.

Table 7: **Forecasting quality and recorded stage costs across seven tasks.** Gain, training time, and selection-file size cover all 42 seed–model–task cells. Selection timing covers 36 cells across the six tasks other than Coastal. Entries are equal-cell means; Gain is paired percentage MASE improvement over Rand.

Method	Gain (%)	Select (s)	Train (s)	File (MiB)
Cells	42	36	42	42
RAGE	+2.03	65.8*	114.0	2.54
RAGE/Block-Linear	+1.05	1.6	145.7	2.37
RAGE/Traj-Linear	-1.39	0.6	147.6	0.11
Rand	+0.00	0.5	144.3	0.08
LESS-TS	-0.36	9.3	146.4	0.12
TSDS	-0.72	131.8	147.1	0.11
LP-Mirror	-7.77	178.9	148.1	81.00
G-Dot	-0.90	46.2	143.5	159.31
GraNd	-2.02	25.2	145.1	157.28

*Recorded joint selection workflow for RAGE (Linear) and three alternative rules for combining reference similarities (Exp, ZeroClip, Max). Table 2 charges this entire workflow to RAGE as a conservative selection-time proxy, rather than dividing its duration among the four rules. Selection times include saving or warm loading and depend on shared-work attribution. Training times exclude evaluation and may reflect resumed runs and hardware/concurrency differences; they do not establish a training speedup. Files exclude shared gradients, Shape feature caches, and selected raw data.

Table 8: **Shared gradient preparation and scoring by model.** Preparation is the cumulative recorded wall-clock time over 12 asset receipts per model (six tasks, two seeds). Scoring time is the mean joint selection time for Linear, Exp, ZeroClip, and Max over the same 12 settings. BP columns are mean required source/reference window-gradient counts for the matched Block-Linear/Shape assets.

Model	Shared prep. (h)	Joint select (s)	Source BP	Reference BP
Chronos-Base	15.00	58.4	136,080	11,889
TimesFM 2.5	9.13	62.0	136,080	11,889
TimerXL	3.66	76.9	136,080	11,889

Preparation receipts cover shared assets used by multiple methods; these totals are not standalone RAGE costs or GPU-hours. BP counts describe required reusable gradients, not new computations charged again to each arm. Cached/imported rows and method preparation deltas depend on execution order. Scoring introduces no additional window-gradient backpropagation.

Shared preparation and computation demand. Table 8 separates asset preparation from subsequent Shape scoring. Shape and Block use the same source and reference gradient anchors and the same nominal 1.5% budget. The required gradient counts are reusable inputs to each method; they are not additional backpropagation charged to every aggregation arm. Shared asset receipts include work used by multiple methods, imported rows, and cache hits. Their cumulative wall-clock totals therefore describe the recorded preparation workload, rather than isolated RAGE preparation time or GPU-hours. The measured workflow computes four ways to combine reference shape similarities: Linear for RAGE, plus Exp, ZeroClip, and Max ablations (Section 3.3). We define the proxy $\widehat{t}_{\text{RAGE}} = t_{\text{joint}}$, charging the entire workflow to RAGE in Table 2, including shared feature/scoring work and the extra rules. This is an accounting proxy for selection with prepared gradients, not a measured single-rule latency; dividing the total by four would instead describe an arbitrary allocation of shared work. Ratios of the mean recorded selection times give approximately 50% lower time than TSDS and 63% lower time than LP-Mirror under this conservative charge.

Training and selection timing. Selection records include saving or warm loading, and shared-work attribution can affect the recorded arm time. Training times are cumulative recorded time through epoch 2 and exclude evaluation; resumed runs and hardware/concurrency differences affect them. In particular, the 114.0-second RAGE mean in Table 7 does not establish a training speedup over Rand. These observations support stagewise cost reporting, not a controlled end-to-end speed ranking. Isolated cold-start latency, GPU device-time, FLOPs, peak CPU memory, and peak CUDA memory were not instrumented.

Selection files and reusable cache inventory. The 2.54 MiB RAGE mean and 2.37 MiB Block-Linear mean are output selection-file sizes over all seven tasks. They exclude shared gradient caches, Shape feature/score caches, and the physical footprint of selected raw trajectories; they are not algorithmic storage lower bounds. The receipt-referenced gradient-cache namespaces occupy 18.67, 18.33, and 18.66 GiB for Chronos, TimesFM, and TimerXL, respectively. These physical inventories include assets beyond the measured task scope and are reused across arms, so they are not seven-task RAGE storage requirements or marginal method costs.

Per-model measured-task details. Tables 9–11 retain the detailed baseline and Block/Traj allocation records on the six tasks with cost receipts. Each trained arm has 12 selections per model (six tasks, two seeds). Their file column is the total over 12 files, whereas Table 7 reports a per-cell mean over all 42 cells. The BP columns give mean required source/reference window-gradient counts; dashes indicate unavailable measurements. The shared Shape-batch timing is reported separately in Table 8.

Table 9: Observed resource costs for Chronos-Base for the measured cells across two seeds (12 selections per trained method). Selection time is the mean per selection; selection-file size is the total over 12 files. Post-training time is the mean of measured cells, with its count in parentheses. Missing measurements are shown as dashes.

Method	Selection (s)	Training (s; n)	Files (MiB)	BP (source, ref.)
Block-Exp	1.79	257.64 (12)	27.79	(136080, 11889)
Orig	–	–	–	(–, –)
Rand	0.57	248.43 (12)	1.01	(–, –)

Continued on next page

Table 9: Observed resource costs for Chronos-Base (continued).

Method	Selection (s)	Training (s; n)	Files (MiB)	BP (source, ref.)
LESS-TS	9.50	256.75 (12)	1.42	(136080, 11889)
TSDS	294.61	257.47 (12)	1.29	(–, –)
LP-Mirror	155.97	260.46 (12)	977.53	(–, –)
G-Dot	40.06	246.66 (12)	1949.13	(136080, 11889)
GraNd-Static	24.97	259.34 (12)	1926.87	(136080, 0)
Block-Linear	1.88	257.01 (12)	28.61	(136080, 11889)
Block-ZeroClip	1.83	256.96 (12)	28.40	(136080, 11889)
Block-Max	1.89	256.78 (12)	28.39	(136080, 11889)
Traj-Exp	0.62	256.33 (12)	1.30	(136080, 11889)
Traj-Linear	0.64	257.35 (12)	1.31	(136080, 11889)
Traj-ZeroClip	0.59	257.22 (12)	1.28	(136080, 11889)
Traj-Max	0.67	256.93 (12)	1.30	(136080, 11889)
Block-Exp/Block-Uni	4.01	293.41 (12)	36.57	(136080, 11889)
Block-Linear/Block-Uni	3.95	291.71 (12)	37.39	(136080, 11889)
Block-Exp/Win-Uni	28.52	291.98 (12)	32.97	(136080, 11889)
Block-Linear/Win-Uni	4.12	293.02 (12)	33.79	(136080, 11889)

Table 10: Observed resource costs for TimesFM 2.5 for the measured cells across two seeds (12 selections per trained method). Selection time is the mean per selection; selection-file size is the total over 12 files. Post-training time is the mean of measured cells, with its count in parentheses. Missing measurements are shown as dashes.

Method	Selection (s)	Training (s; n)	Files (MiB)	BP (source, ref.)
Block-Exp	1.77	148.64 (12)	27.21	(136080, 11889)
Orig	–	–	–	(–, –)
Rand	0.42	148.03 (12)	1.00	(–, –)
LESS-TS	9.51	147.54 (12)	1.40	(136080, 11889)
TSDS	94.47	149.13 (12)	1.29	(–, –)
LP-Mirror	204.45	143.67 (12)	977.52	(–, –)
G-Dot	42.03	141.06 (12)	1839.46	(136080, 11889)
GraNd-Static	20.56	141.90 (12)	1804.18	(136080, 0)
Block-Linear	1.60	148.49 (12)	28.19	(136080, 11889)
Block-ZeroClip	1.63	147.58 (12)	27.99	(136080, 11889)
Block-Max	1.67	148.12 (12)	28.01	(136080, 11889)
Traj-Exp	0.58	150.64 (12)	1.29	(136080, 11889)
Traj-Linear	0.62	149.07 (12)	1.29	(136080, 11889)
Traj-ZeroClip	0.57	148.61 (12)	1.28	(136080, 11889)
Traj-Max	0.58	149.13 (12)	1.29	(136080, 11889)
Block-Exp/Block-Uni	3.06	144.04 (12)	35.95	(136080, 11889)
Block-Linear/Block-Uni	3.05	144.53 (12)	36.93	(136080, 11889)
Block-Exp/Win-Uni	27.08	151.53 (12)	32.28	(136080, 11889)
Block-Linear/Win-Uni	2.64	149.57 (12)	33.26	(136080, 11889)

Table 11: Observed resource costs for TimerXL for the measured cells across two seeds (12 selections per trained method). Selection time is the mean per selection; selection-file size is the total over 12 files. Post-training time is the mean of measured cells, with its count in parentheses. Missing measurements are shown as dashes.

Method	Selection (s)	Training (s; n)	Files (MiB)	BP (source, ref.)
Block-Exp	1.56	30.99 (12)	27.75	(136080, 11889)
Orig	–	–	–	(–, –)
Rand	0.46	33.37 (12)	1.01	(–, –)
LESS-TS	8.90	33.46 (12)	1.41	(136080, 11889)
TSDS	6.45	30.95 (12)	1.29	(–, –)

Continued on next page

Table 11: Observed resource costs for TimerXL (continued).

Method	Selection (s)	Training (s; n)	Files (MiB)	BP (source, ref.)
LP-Mirror	176.26	32.60 (12)	933.10	(–, –)
G-Dot	56.51	33.26 (12)	1945.73	(136080, 11889)
GraNd-Static	30.13	33.66 (12)	1931.08	(136080, 0)
Block-Linear	1.43	30.18 (12)	28.74	(136080, 11889)
Block-ZeroClip	1.50	30.90 (12)	28.53	(136080, 11889)
Block-Max	1.51	30.77 (12)	28.53	(136080, 11889)
Traj-Exp	0.39	30.49 (12)	1.30	(136080, 11889)
Traj-Linear	0.42	30.56 (12)	1.30	(136080, 11889)
Traj-ZeroClip	0.40	30.69 (12)	1.27	(136080, 11889)
Traj-Max	0.39	30.62 (12)	1.30	(136080, 11889)
Block-Exp/Block-Uni	3.03	33.99 (12)	36.60	(136080, 11889)
Block-Linear/Block-Uni	3.52	33.78 (12)	37.60	(136080, 11889)
Block-Exp/Win-Uni	20.24	34.06 (12)	32.92	(136080, 11889)
Block-Linear/Win-Uni	2.53	33.98 (12)	33.92	(136080, 11889)

J Related Work

Foundation-model adaptation. Chronos, TimesFM, Moirai, MOMENT, Sundial, Time-MoE, and Timer-S1 show forecasting transfer through large-scale pretraining and downstream adaptation (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Goswami et al., 2024; Liu et al., 2025b; Shi et al., 2025; Liu et al., 2026b). Timer-XL uses causal attention for unified time series forecasting (Liu et al., 2025a). In-context fine-tuning uses related sequences to adapt a forecasting model (Faw et al., 2025); LoRA fine-tuning has also been evaluated for out-of-domain time-series modalities (Gupta et al., 2024). MixFT partitions fine-tuning series into subdomains with a Bayesian mixture and trains a separate LoRA adapter for each subdomain (Lee et al., 2026). RAGE studies source exposure during adaptation. It assigns probabilities to local forecasting events and uses them with a fixed training procedure for each model.

Target-guided data valuation. Influence functions connect training examples to changes in model predictions (Koh & Liang, 2017). TracIn estimates training influence from checkpoint gradients (Pruthi et al., 2020); TRAK uses projected gradients for scalable model attribution (Park et al., 2023), while DataInf approximates influence in LoRA-tuned models (Kwon et al., 2024). LESS uses low-dimensional gradient features to select data for target capabilities (Xia et al., 2024). TSDS combines target-distribution matching with diversity (Liu et al., 2024). GRAD-MATCH selects weighted subsets to match a target gradient (Killamsetty et al., 2021). In-Run Data Shapley measures contribution over a training run and uses efficient gradient products (Wang et al., 2025). Recent methods further reduce gradient-based valuation cost with influence-preserving low-rank proxies, landmark propagation, and target-gradient subspaces (Chen et al., 2026; Nikdan et al., 2025; Min et al., 2026). For time series, TimeInf aggregates influence over overlapping blocks containing a time point (Zhang et al., 2025); LTSV approximates influence through one-step in-context fine-tuning and aggregates overlapping block values (Wu et al., 2026). TimeLAVA instead values segments by their effect on a wavelet-based distributional discrepancy (Liu et al., 2026a); FAF filters examples online using reducible holdout loss and augments the retained time series (Taga et al., 2026). These methods estimate data contribution or segment value, whereas RAGE matches reconstructed local gradient shapes to references and converts shape-score ranks into post-training exposure probabilities.

Model-aware data preparation and time series selection. DoReMi uses a proxy model to choose domain proportions for language-model pretraining (Xie et al., 2023). For forecasting, LP-Mirror uses a lightweight linear proxy for sample valuation (Ao & Chen, 2026). Conflicting task gradients have been studied in multi-task optimization through projection, conflict-averse updates, and bargaining-based gradient combination (Yu et al., 2020; Liu et al., 2021; Navon et al., 2022). Negative transfer also depends on task generalization, not gradient conflict alone (Jiang et al., 2023). RAGE uses opposing anchor directions to decide where another gradient should be computed. It keeps the full source pool through a soft window distribution; its allocation variable is how often each legal origin enters post-training.