GRAM: GRAPH-RETRIEVED AMPLITUDE MEMORY FOR ZERO-SHOT FORECASTING WITH FROZEN TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series foundation models (TSFMs) enable zero-shot forecasting through large-scale cross-domain pretraining, while retrieval augmentation further improves their performance by leveraging historical information. However, existing methods typically correct TSFM forecasts using the ground-truth futures of similar historical windows, which contain both predictive components already captured by the foundation model and sample-specific random fluctuation that is difficult to transfer. In contrast, recurring systematic model bias within prediction errors more directly characterizes the failure modes of a frozen TSFM and therefore provides more valuable correction signals. Effectively exploiting such model bias, however, poses two challenges: prediction errors at different numerical levels are difficult to compare due to scale differences, and the recurring bias must be extracted from prediction errors contaminated by random fluctuation. To address these challenges, we propose GRAM, a general retrieval-augmented framework for frozen TSFMs. GRAM first introduces an Amplitude Memory Module (AMM) that scales prediction errors by amplitude and aggregates them into retrievable prototypes. It then employs a Prototype Graph Module (PGM) to model relations among prototypes to aggregate consistent bias information while suppressing random fluctuation. During online forecasting, GRAM retrieves and expands prototypes relevant to the current query and generates per-horizon corrections to refine the original TSFM forecast. Experiments across multiple datasets and foundation models demonstrate consistent forecasting improvements.

1 INTRODUCTION

Time-series foundation models (TSFMs) are pretrained on large-scale cross-domain corpora Shi et al. (2025); Liu et al. (2024c); Goswami et al. (2024); Ekambaram et al. (2024); Rasul et al. (2023); Garza & Canseco (2023); Gruver et al. (2023); Jin et al. (2024); Zhou et al. (2023); Liu et al. (2024a); Liang et al. (2024); Wen et al. (2022), enabling a single model to transfer to diverse target scenarios for zero-shot forecasting without training a separate predictor for each dataset. This capability is particularly valuable when target-domain data are limited, model updates are costly, or the deployed predictor must remain frozen. However, despite their strong generalization ability, a pretrained TSFM may still exhibit systematic forecasting errors when transferred to a specific target domain Guo et al. (2017); Kim et al. (2022); Yao et al. (2022); Glahn & Lowry (1972). Effectively correcting TSFM prediction errors and improving forecasting accuracy while keeping the model parameters frozen is therefore of significant research and practical importance.

Recent retrieval-augmented forecasting methods alleviate this problem by exploiting historical data from the target domain Yang et al. (2024); Zhou et al. (2026). These methods retrieve historical windows with similar temporal patterns Agrawal & Srikant (1995); Mueen et al. (2009) and use the ground-truth future trajectories that follow them as auxiliary evidence. Their implicit assumption is that similar historical patterns correspond to transferable future trajectories. However, this strategy overlooks the forecasting capability that the frozen TSFM already possesses. For a historical window i, let $\hat{\mathbf{y}}_i^{\mathrm{tsfm}}$ denote the prediction of the frozen TSFM and $\mathbf{y}_i$ the ground-truth future. Conceptually,

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.

the ground-truth future can be expressed as

$$\mathbf{y}_i = \underbrace{\hat{\mathbf{y}}_i^{\mathrm{tsfm}}}_{\text{model prediction}} + \underbrace{\mathbf{b}_i}_{\text{model bias}} + \underbrace{\boldsymbol{\epsilon}_i}_{\text{sample-specific variation}},$$

where $\mathbf{b}_i$ denotes a recurring systematic bias under similar historical patterns, and $\boldsymbol{\epsilon}_i$ denotes sample-specific variation that is not reliably transferable. For the current query q, the frozen TSFM has already produced its prediction $\hat{\mathbf{y}}_q^{\mathrm{tsfm}}$. Directly retrieving $\mathbf{y}_i$ therefore introduces two components irrelevant to correction. The term $\hat{\mathbf{y}}_i^{\mathrm{tsfm}}$ is already covered by the query's own prediction $\hat{\mathbf{y}}_q^{\mathrm{tsfm}}$, while $\boldsymbol{\epsilon}_i$, as non-transferable random fluctuation, would interfere with the forecast. What is truly valuable for correction is only the recurring model bias $\mathbf{b}_i$.

For non-stationary time series Liu et al. (2022b), this problem is particularly pronounced: since the underlying generating mechanism of the series evolves over time, even two historical windows with similar patterns may lead to substantially different futures in level, amplitude, or trend. This motivates us to shift the retrieval target from complete future trajectories to prediction residuals. The residual of a historical window i is defined as

$$\mathbf{r}_i = \mathbf{y}_i - \hat{\mathbf{y}}_i^{\mathrm{tsfm}} = \mathbf{b}_i + \boldsymbol{\epsilon}_i.$$

After removing the model's prediction, the residual retains only the model's forecasting error, directly corresponding to the correction objective. However, a single residual is still not an ideal correction signal: the presence of $\boldsymbol{\epsilon}_i$ means that directly reusing the residual of one historical window may transfer random fluctuation along with the bias. Therefore, it is necessary to aggregate residual evidence from multiple related historical windows to extract the recurring model bias $\mathbf{b}_i$.

However, this process faces two challenges. First, absolute residuals are strongly affected by the local numerical level of the series. Two historical windows may exhibit the same relative prediction bias, yet produce absolute residuals of vastly different magnitudes due to different numerical levels. Standard normalization can only align the global scale of a series and cannot eliminate local numerical differences between windows. Therefore, absolute residuals cannot be directly compared across windows with different numerical levels, and a scale-invariant bias representation is required. Second, the residual of a single historical window contains random fluctuation, making it difficult to reliably identify recurring model bias. Moreover, model bias and historical patterns are not in one-to-one correspondence: different historical patterns may produce similar model bias, and similar historical patterns may also produce different model bias. Useful bias information may therefore be dispersed across multiple distinct but related historical windows, requiring explicit modeling of the relationships among windows to extract recurring model bias.

To address the above problems, we propose GRAM, a retrieval-augmented forecasting framework for frozen TSFMs. GRAM constructs a structured bias memory and uses model bias to correct the current forecast, comprising two phases: Offline Memory Construction and Online Forecasting. In the Offline Memory Construction phase, the Amplitude Memory Module (AMM) represents model bias using amplitude scaling error rather than absolute residuals, making bias comparable across historical windows with different numerical levels, and aggregates historical windows into retrievable prototypes. The Prototype Graph Module (PGM) establishes relationships among prototypes, organizing bias information dispersed across different historical windows into a unified structure. In the Online Forecasting phase, GRAM retrieves prototypes relevant to the current query and expands along the prototype graph to obtain broader bias information. The Graph Fusion Module (GFM) aggregates the retrieved bias information into per-horizon corrections, which are added to the original forecast of the frozen TSFM. Throughout the entire process, the TSFM and the retrieval encoder remain frozen.

Our contributions are summarized as follows:

- We propose retrieving the prediction residuals of the TSFM on historical windows, so that retrieval retains only the information directly relevant to model correction.
- We design the Amplitude Memory Module (AMM), which uses amplitude scaling error to characterize the prediction bias of the frozen TSFM under different historical patterns, and organizes related historical windows into retrievable prototypes.
- We design the Prototype Graph Module (PGM), which establishes associations among prototypes to aggregate bias information dispersed across multiple historical windows.

- We conduct extensive experiments on seven public benchmark datasets and four time-series foundation models. The results show that GRAM achieves the best forecasting performance across different datasets and foundation models.

2 RELATED WORK

2.1 TIME-SERIES FORECASTING WITH FOUNDATION MODELS

Time-series forecasting has progressed from statistical models Box et al. (2015) to deep architectures that capture long-range, cross-variable, frequency or multi-scale patterns Flunkert et al. (2017); Lim et al. (2019); Oreshkin et al. (2020); Challu et al. (2023); Zhou et al. (2022); Liu et al. (2022a); Zhang & Yan (2023); Chen et al. (2023); all of these models are trained per dataset and depend on target labels. TSFMs instead pretrain once on large cross-domain corpora and forecast zero-shot on unseen domains, removing the dependence on target labels. Parameter-efficient adaptation further reduces the number of updated parameters but still requires target-specific optimization. In contrast, GRAM keeps the TSFM frozen and corrects its systematic scaling errors through an external memory.

2.2 RETRIEVAL-AUGMENTED FORECASTING

Retrieval-augmented methods complement parametric models with non-parametric information selected from an external corpus Johnson et al. (2021). The principle originates in language modeling Lewis et al. (2020); Guu et al. (2020); Izacard et al. (2022); Karpukhin et al. (2020); Khandelwal et al. (2020); Borgeaud et al. (2022); Gao et al. (2023) and has recently been extended to time-series forecasting. Existing methods differ mainly in how the retrieved evidence is incorporated. TimeRAG Yang et al. (2024) builds a knowledge base of time-series segments and couples retrieval with a large language model; TimeRAF Zhang et al. (2025) trains a bi-encoder that retrieves historical windows and uses their futures as auxiliary signals for the forecast; TS-RAG Ning et al. (2025) augments a frozen TSFM with a retrieval knowledge base and fuses the retrieved patterns with the TSFM representation through an attention-based mixer; related designs adopt similar retrieve-then-fuse pipelines Han et al. (2025). These methods demonstrate that target-domain history can improve a frozen forecaster, but they generally treat retrieved windows as independent items and use similarity as the primary relevance signal. GRAM instead retrieves the scaling error that the frozen TSFM leaves on a pattern and reads it through a prototype graph: prototypes are connected by bias and shape relations, and the retrieved subgraph is processed by a trainable relational message-passing head whose output is gated by the evidence behind each retrieved prototype.

3 METHOD

We study zero-shot forecasting with a frozen time-series foundation model (TSFM). The input is a context window $\mathbf{x}_q$ of recent observations, and the goal is to predict the next H steps $\mathbf{y}_q$ without using any label of the target series. Two components stay frozen throughout: a TSFM F that produces the zero-shot forecast $\widehat{\mathbf{y}}_q^{\mathrm{tsfm}} = F(\mathbf{x}_q)$, and a retrieval encoder E that maps any window to a unit-norm embedding; we write e_q for the query and $e_1, \ldots, e_n$ for the historical windows. Retrieval-augmented forecasting keeps this pipeline but reuses the ground-truth futures that follow the retrieved windows. Such a future, however, contains the model's own prediction and non-transferable sample-specific fluctuation on top of the recurring model bias, so reusing it transfers both a component already covered by the query's own forecast and fluctuation that interferes with it; only the recurring bias is useful for correction. GRAM keeps the same retrieval pipeline but changes the target of retrieval accordingly: each prototype stores, for a recurring pattern, the amplitude of the systematic error of the TSFM, and a trainable read-out turns the retrieved subgraph into an error component added to the frozen forecast.

Let t_q denote the last observed step of the query window and S the number of steps that define the recent level. The recent level, the predicted excursion and the realized excursion are

$$m_q \;=\; \frac{1}{S}\sum_{t=t_q-S+1}^{t_q} x_{q,t}, \qquad \mathrm{exc}_q(h) \;=\; \widehat{y}_q^{\mathrm{tsfm}}(h) - m_q, \qquad \mathrm{res}_q(h) \;=\; y_q(h) - m_q, \quad (1)$$

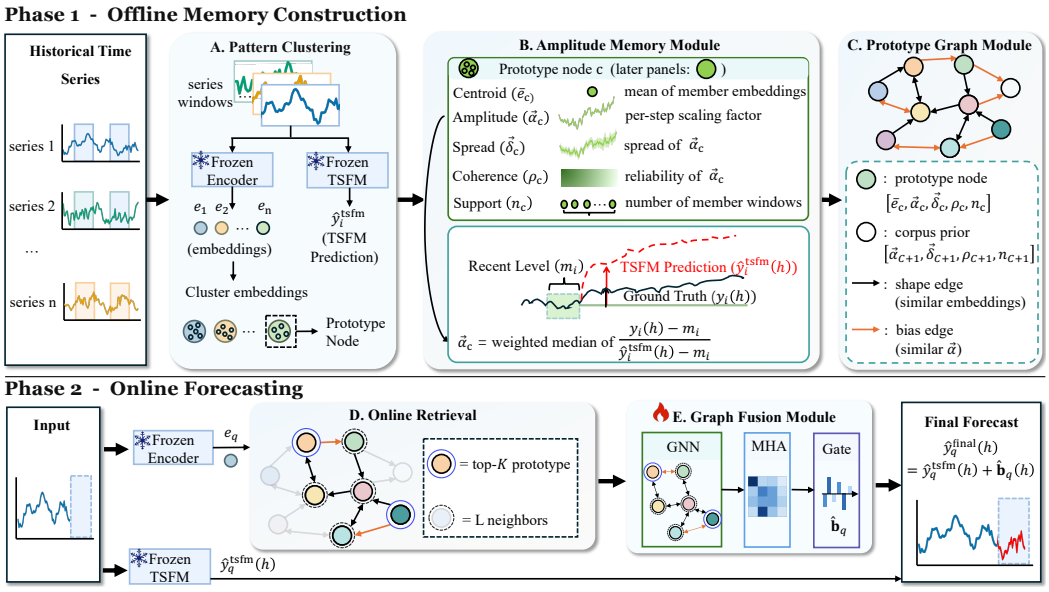

Figure 1: Overview of GRAM. Phase 1 (Offline Memory Construction): clustering (A) groups window embeddings into prototypes, the amplitude memory (B) stores each prototype's centroid, amplitude, spread, coherence and support, and the prototype graph (C) links them by bias and shape edges. Phase 2 (Online Forecasting): retrieval (D) grows the top-K prototypes into a subgraph and attaches the anchor node, and graph fusion (E), the only trainable module, outputs the error component added to the frozen forecast.

for $h \in \{1, \ldots, H\}$. Their ratio $\gamma_q(h) = \text{res}_q(h)/\text{exc}_q(h)$ yields the factorization $y_q(h) - \hat{y}_q^{\text{tsfm}}(h) = \text{res}_q(h) - \text{exc}_q(h) = \big(\gamma_q(h) - 1\big)\,\text{exc}_q(h)$: the first factor is dimensionless and states how the TSFM mis-scales the departure, while the second carries the local numerical level. Storing the first factor and re-applying it to the query's own excursion is what makes the memory scale-invariant.

3.1 Overview

Figure 1 shows the pipeline. Offline, pattern clustering (A) groups the window embeddings into C prototypes, the amplitude memory (B) stores a prototype record for each of them, and the prototype graph (C) links them by the bias and shape relations (Section 3.2). Online, retrieval (D) grows the top-K prototypes into a subgraph and attaches an anchor node, and the graph fusion module (E), the only trainable component, turns the subgraph into an error component $\hat{\mathbf{b}}_q(h)$ that the final forecast adds to the frozen prediction (Section 3.3). Setting $\hat{\mathbf{b}}_q(h) \equiv 0$ recovers the frozen TSFM exactly.

3.2 Offline Memory Construction

The offline phase distills the target history into two artifacts: the amplitude memory, which records how the TSFM errs on each pattern, and the prototype graph, which records which patterns err alike. Amplitudes rather than absolute residuals keep the memory dimensionless and reusable across corpora, and links between patterns that err alike let evidence recorded on one pattern be reused on another. The phase runs once per dataset, TSFM and horizon, and only on the memory-construction segment, so it never sees a test label.

3.2.1 Pattern Clustering

Module A runs the two frozen components over every window of the target history, producing the embeddings $e_1, \ldots, e_n$ and the forecasts $\hat{y}_1^{\text{tsfm}}, \ldots, \hat{y}_n^{\text{tsfm}}$, and groups the embeddings into C prototype nodes Sculley (2010); Snell et al. (2017). The centroid $\bar{e}_c$ of prototype c is the unit-normalized mean of its member embeddings; it is the identity of the node and the key used for

retrieval. A window is assigned to the closest centroid, $z_i = \arg\min_c \|e_i - \bar{e}_c\|_2$, and each cluster records its support n_c, the number of member windows behind it. The segment split and the values of C and K are reported in Appendix A.

3.2.2 AMPLITUDE MEMORY

The amplitude memory records, for every cluster, how the TSFM mis-scales the excursions of its member windows. For a member window i of cluster c, the per-step ratio $\gamma_i(h) = \mathrm{res}_i(h)/\mathrm{exc}_i(h)$ is the factor between the realized and predicted excursions at step h. The cluster aggregates these ratios into an amplitude vector $\vec{\alpha}_c \in \mathbb{R}^H$ and a spread vector $\vec{\delta}_c \in \mathbb{R}^H$,

$$\alpha_c(h) \;=\; \mathrm{med}_{i:z_i=c}\big[\gamma_i(h);\ \omega_i(h) = |\mathrm{exc}_i(h)|\big], \tag{2}$$

$$\delta_c(h) \;=\; \frac{\sum_{i:z_i=c} \omega_i(h)\,\big|\gamma_i(h) - \alpha_c(h)\big|}{\sum_{i:z_i=c} \omega_i(h)}, \tag{3}$$

where med denotes the $\omega_i(h)$-weighted median; the weight down-weights windows whose excursion is close to zero, because they have no departure to rescale. We call $\alpha_c(h) - 1$ the *amplitude scaling error* of pattern c at lead h: in the decomposition of the introduction, $\big(\alpha_c(h) - 1\big)\,\mathrm{exc}_i(h)$ is the recurring bias $\mathbf{b}_i$ of a member window expressed relative to its own excursion, while $\vec{\delta}_c$ summarizes the sample-specific component $\boldsymbol{\epsilon}_i$. The cluster also records a coherence score

$$\rho_c \;=\; \frac{1}{H}\sum_{h=1}^{H} \frac{\alpha_c(h)^2}{\alpha_c(h)^2 + \delta_c(h)^2 + \varepsilon_0}, \tag{4}$$

which is close to one when the member windows agree on how the TSFM erred on the pattern. Each prototype therefore carries the five-tuple $\big[\bar{e}_c, \vec{\alpha}_c, \vec{\delta}_c, \rho_c, n_c\big]$, of which the centroid serves as the identity of the node for retrieval and the remaining attributes are consumed by the read-out.

3.2.3 PROTOTYPE GRAPH

The prototype graph connects the C prototypes with two relations, each prototype keeping its strongest neighbors under each relation; the out-degree is a hyper-parameter reported in the experiments. For the *bias relation*, we subtract one from each amplitude vector, forming the scaling-error curve $\vec{\alpha}_c - \mathbf{1}$, and ℓ_2-normalize it; the bias similarity is the cosine of these curves, and a bias edge requires the neighbor's support to exceed a floor $n_{\min}$, without which low-support prototypes could be linked by noise. For the *shape relation*, the similarity is the cosine of the centroids $\bar{e}_c$; since a prototype with very small support has an amplitude close to one, because the weighted median over few members shrinks toward one, its shape neighbors provide the usable evidence for the read-out. The bias relation states that two prototypes are mis-scaled in the same way, which is the property that allows an amplitude correction to transfer between them. Finally, a corpus prior summarizes the whole library as the support-weighted average,

$$\vec{\alpha}_{C+1} = \frac{\sum_c n_c\,\vec{\alpha}_c}{\sum_c n_c}, \qquad \vec{\delta}_{C+1} = \frac{\sum_c n_c\,\vec{\delta}_c}{\sum_c n_c}, \qquad \rho_{C+1} = \frac{\sum_c n_c\,\rho_c}{\sum_c n_c}, \qquad n_{C+1} = \sum_c n_c, \tag{5}$$

attached to the graph as an embedding-free node: it carries no centroid, is excluded from message passing, and only votes in the anchor aggregation of the online phase.

3.3 ONLINE FORECASTING

The online phase serves one query at a time without accessing any target label. The aggregated retrieval answer is attached as an anchor node and enters the correction with a fixed weight, so the read-out always contains the simplest retrieval baseline and every gain over it has to come from the graph; the trained part is gated by the evidence behind the vote, so it shrinks when the retrieved prototypes disagree.

3.3.1 ONLINE RETRIEVAL

Retrieval compares e_q with the centroids, keeps the K best prototypes as seeds $\mathcal{S}_q = \{c_1, \ldots, c_K\}$, and grows them one hop along the two relations, keeping the L strongest neighbors of each seed.

The stored record of every node in this subgraph is loaded from the amplitude memory together with the seed similarities $\mathrm{sim}_{q,c}$, and the aggregated answer over the seeds and the corpus prior is attached as the anchor node:

$$\tilde{\alpha}_q(h) \;=\; \sum_{j \in \mathcal{S}_q \cup \{C+1\}} \tilde{w}_j \, \alpha_j(h), \qquad \tilde{w}_j \;\propto\; \mathrm{softmax}_{j \in \mathcal{S}_q \cup \{C+1\}}(\mathrm{sim}_{q,j}) \, \rho_j, \tag{6}$$

where the corpus prior, which carries no centroid, is given the neutral similarity $\mathrm{sim}_{q,C+1} = \frac{1}{K} \sum_{j \in \mathcal{S}_q} \mathrm{sim}_{q,j}$. The anchor carries the aggregated record in the same five-tuple format (Appendix A.4); since it joins the subgraph without any edge, attending to it alone reproduces the fixed-fraction baseline that this curve defines.

3.3.2 Graph Fusion

The graph fusion module maps the subgraph to the per-horizon error component $\hat{\mathbf{b}}_q$ through three blocks, GNN, MHA and Gate, as shown in module E of Figure 1. It is fitted once on a large multi-domain corpus by minimizing the error of the corrected forecast, under the same frozen-TSFM contract: only the head is trained, the TSFM and the retrieval encoder remain frozen, and no label of the evaluation domains is used (Appendix A.4). Each node is featurized with its stored curves and scalar statistics, and a relational message-passing network with a separate projection per relation propagates them over the two relations (Appendix A.4); the anchor is excluded from message passing because it carries no edges. The query is encoded from scale statistics of its own excursion (Appendix A.4); a multi-head attention layer Vaswani et al. (2017) takes this vector as the only query and the propagated node representations as keys and values, and returns an attention-pooled vector together with the attention weights $\boldsymbol{\pi}_q$, one per subgraph node. A two-layer head maps the pooled and query features to a per-lead gain $g(h) \geq 0$ that weights the voted scaling, and its last linear layer is initialized so that $g \equiv 1$ at the start of training: the head begins by passing the vote through unweighted and learns only the per-lead departures from it.

Gate block. The attention weights define a voted curve,

$$v(h) \;=\; \sum_{c} \pi_{q,c} \, \big(\alpha_c(h) - 1\big), \tag{7}$$

a parameter-free, attention-weighted scaling error. The gate caps the scaling with the evidence behind the vote,

$$\phi(h) \;=\; \sigma\Big(b + \kappa_0 \sum_{c} \pi_{q,c} \, \mathrm{rel}(n_c) + \kappa_1 \sum_{c} \pi_{q,c} \, \rho_c - \kappa_2 \sum_{c} \pi_{q,c} \, \delta_c(h) - \kappa_3 \sum_{c} \pi_{q,c} \, \big|\alpha_c(h) - 1 - v(h)\big|\Big), \tag{8}$$

where $\mathrm{rel}(n_c) = \log(1 + n_c) - \log(1 + \mathrm{med}_{c'} n_{c'})$ is the support relative to the median of the C prototypes, the gains $\kappa_i = \mathrm{softplus}(\cdot)$ are non-negative, and the sums run over the nodes of the subgraph; Appendix A.4 details each term. The signs are fixed: larger support and coherence can raise the ceiling, while larger spread or node disagreement must lower it. Since $\phi(h)$ lies in $(0, 1)$ and $v(h)$ is bounded by the stored amplitudes, the gate can shrink the trained part of the correction but cannot change its sign, while the two fixed terms hold it at the level of the retrieved answer and of the corpus.

3.3.3 Final Forecast

The final forecast adds the error component to the frozen prediction,

$$\widehat{y}_q^{\,\mathrm{final}}(h) \;=\; \widehat{y}_q^{\,\mathrm{tsfm}}(h) + \hat{\mathbf{b}}_q(h), \qquad h = 1, \ldots, H, \tag{9}$$

$$\hat{\mathbf{b}}_q(h) \;=\; \big(a(h) - 1\big) \, \mathrm{exc}_q(h), \tag{10}$$

$$a(h) \;=\; 1 + \mu\big(\tilde{\alpha}_q(h) - 1\big) + \nu\Big[\big(\alpha_{C+1}(h) - 1\big) + \phi(h) \, g(h) \, v(h)\Big]. \tag{11}$$

The ratio draws on three sources, of which the first two are fixed before training and carry the constants μ and ν (Appendix A.4). The pooled retrieval answer of Eq. equation 6, scaled by the anchor weight μ, keeps the simplest retrieval baseline as a standing component of the correction; the corpus prior of Eq. equation 5, scaled by the residual weight ν, supplies the library-wide scaling error, so that a quiet neighbourhood still receives a conservative correction. What training adds is the

last term: the voted scaling of Eq. equation 7, capped by the gate $\phi(h)$ and reweighted per lead by the gain $g(h)$. Every prototype therefore contributes $\big(\alpha_c(h) - 1\big)\mathrm{exc}_q(h)$, that is, how much it says the TSFM overstated ($\alpha_c(h) < 1$) or understated ($\alpha_c(h) > 1$) the departure at step h, and the anchor and the corpus prior enter in the same units. Read against the decomposition of the introduction, $\hat{\mathbf{b}}_q$ is an estimate of the recurring bias $\mathbf{b}_q$ whose trained part is gated by the evidence behind it. Setting $\hat{\mathbf{b}}_q(h) \equiv 0$ recovers $\widehat{y}_q^{\mathrm{tsfm}}$ exactly, so the frozen TSFM is the identity point of the correction.

4 EXPERIMENTS

Datasets. We evaluate on seven widely used benchmarks: ETTh1, ETTh2, ETTm1, and ETTm2 Zhou et al. (2021), together with Weather, Electricity, and Exchange Rate Wu et al. (2023), following their canonical chronological splits. Dataset statistics and the exact split boundaries are reported in Appendix A.1.

TSFMs. GRAM is attached to four TSFMs with different architectures and pretraining corpora: Chronos-Bolt Ansari et al. (2024), Sundial Liu et al. (2025), TimesFM Das et al. (2024), and Moirai Woo et al. (2024). All TSFMs remain frozen, and retrieval embeddings are computed by a frozen Chronos encoder shared across TSFMs, so a single correction pipeline is applied unchanged to every model.

Setup. The context length is fixed to 512, and the forecast horizon is $H \in \{96, 192, 336, 720\}$. Each channel is forecast independently. For each dataset–horizon pair and each TSFM, the official training split is divided chronologically into a memory portion and a calibration portion, separated from the test split by a horizon-length gap; the Amplitude Memory Module (AMM) and the Prototype Graph Module (PGM) are built only from these observed portions, and the Graph Fusion Module (GFM) is fitted once on the LOTSA corpus Woo et al. (2024) and then frozen for all target benchmarks and TSFMs. Only the GFM is trained; the TSFM and the retrieval encoder stay frozen, and no label of the evaluation domains is used. Since the only fitted part is fixed on an unrelated corpus before the test portion of a target cell is read, every comparison reported below is zero-shot with respect to the evaluation domains. The complete protocol, with the hyper-parameters and the machine used, is reported in Appendix A.1.

Baselines and metrics. The baseline is the frozen TSFM itself: for each of the four models we compare its zero-shot forecast with the same forecast corrected by GRAM, on the same dataset and horizon, so that every comparison below is a like-for-like one between *Ori* and +GRAM. We also report LoRA fine-tuned versions of the four TSFMs Hu et al. (2022), which show whether the correction survives once the model has been adapted to the target domain. To ensure consistency with previous studies, we adopt Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics, always comparing MSE with MSE and MAE with MAE; lower is better in both. Their implementation details are reported in Appendix A.5.

4.1 MAIN RESULTS

Zero-shot correction across TSFMs. Table 1 reports, for each of the four TSFMs and each dataset, the zero-shot forecast of the frozen model (*Ori*) and the same forecast corrected by GRAM (+GRAM), averaged over the four horizons. GRAM improves both metrics in all 56 dataset–TSFM–metric cells of the table: every one of the four TSFMs is improved on every one of the seven datasets, in MSE and in MAE. The largest reductions come from the weakest frozen model at the longest horizon and reach 34.5% of MSE and 16.7% of MAE—Moirai on Electricity at $H = 720$, where MSE falls from 1.476 to 0.966 and MAE from 0.942 to 0.785—with 30.0% of MSE under Sundial as well; fourteen of the 112 settings remove at least a tenth of the MSE, and ten at least a fifth. Complete per-setting results are in Appendix B.1.

Correction after parameter-efficient adaptation. Table 2 asks whether GRAM remains useful once the TSFM has been adapted. Each TSFM is fine-tuned with LoRA on chronological training windows and gets its own AMM and PGM; *FT* denotes the fine-tuned model and +GRAM the same model corrected, with MSE and MAE averaged over the four horizons. The correction still

Table 1: Zero-shot forecasting comparison between frozen foundation models and the same models corrected by GRAM across seven datasets. For each TSFM, *Ori* is the bare frozen model and +GRAM the corrected forecast; MSE is compared with MSE and MAE with MAE, and **red** marks the better of the two. Results are averaged over horizons $H \in \{96, 192, 336, 720\}$.

Models	Sundial				Chronos				TimesFM				Moirai			
	Ori		+GRAM (Ours)		Ori		+GRAM (Ours)		Ori		+GRAM (Ours)		Ori		+GRAM (Ours)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.397	0.412	**0.389**	**0.408**	0.448	0.418	**0.436**	**0.414**	0.421	0.409	**0.417**	**0.407**	0.620	0.504	**0.530**	**0.473**
ETTh2	0.343	0.383	**0.339**	**0.379**	0.367	0.380	**0.362**	**0.377**	0.372	0.388	**0.369**	**0.385**	0.451	0.438	**0.405**	**0.415**
ETTm1	0.366	0.370	**0.360**	**0.367**	0.421	0.383	**0.411**	**0.379**	0.381	0.372	**0.378**	**0.370**	0.740	0.523	**0.617**	**0.485**
ETTm2	0.275	0.321	**0.264**	**0.313**	0.291	0.317	**0.287**	**0.315**	0.291	0.318	**0.285**	**0.315**	0.404	0.388	**0.344**	**0.360**
ECL	0.148	0.243	**0.147**	**0.242**	0.159	0.244	**0.159**	**0.244**	0.164	0.249	**0.164**	**0.249**	0.520	0.446	**0.391**	**0.407**
Weather	0.244	0.275	**0.236**	**0.269**	0.263	0.276	**0.257**	**0.273**	0.263	0.268	**0.253**	**0.264**	0.374	0.333	**0.315**	**0.310**
Exchange	0.558	0.499	**0.439**	**0.440**	0.371	0.412	**0.358**	**0.402**	0.404	0.420	**0.379**	**0.410**	0.601	0.488	**0.463**	**0.443**

Table 2: Forecasting with LoRA fine-tuned TSFMs. For each TSFM, *FT* is the LoRA fine-tuned model and +GRAM the same fine-tuned model corrected by GRAM; MSE and MAE are averaged over $H \in \{96, 192, 336, 720\}$, and **red** marks the better of the two within each TSFM–metric pair. Lower is better.

Models	Sundial				Chronos				TimesFM				Moirai			
	FT		+GRAM (Ours)		FT		+GRAM (Ours)		FT		+GRAM (Ours)		FT		+GRAM (Ours)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.403**	**0.415**	0.411	0.417	0.461	**0.421**	**0.452**	0.422	0.489	0.438	**0.489**	**0.438**	0.594	0.493	**0.505**	**0.462**
ETTh2	0.338	0.376	**0.335**	**0.374**	0.375	0.384	**0.370**	**0.383**	0.366	0.388	**0.365**	**0.387**	0.432	0.422	**0.394**	**0.404**
ETTm1	0.351	0.362	**0.343**	**0.361**	0.349	0.354	**0.341**	**0.352**	0.304	0.355	**0.303**	**0.354**	0.566	0.451	**0.477**	**0.422**
ETTm2	0.274	0.319	**0.262**	**0.311**	0.275	0.308	**0.274**	**0.307**	0.269	0.305	**0.268**	**0.305**	0.318	0.342	**0.293**	**0.329**
Weather	0.249	0.279	**0.235**	**0.269**	0.236	0.247	**0.229**	**0.245**	0.229	0.250	**0.228**	**0.250**	0.280	0.288	**0.256**	**0.275**
Exchange	0.496	0.462	**0.432**	**0.432**	0.396	0.420	**0.375**	**0.407**	**0.400**	**0.423**	0.426	0.435	0.599	0.487	**0.470**	**0.445**
ECL	0.146	0.242	**0.146**	**0.242**	0.157	0.243	**0.157**	**0.243**	0.159	0.246	**0.159**	**0.246**	0.467	0.430	**0.376**	**0.402**

improves both metrics on all seven datasets under Moirai and on six of them under Chronos and Sundial, again most where the adapted model is weakest: fine-tuned Moirai on ETTh1 at $H = 720$ goes from 1.077 to 0.753 MSE, a 30.1% reduction. On TimesFM the correction is nearly switched off — the adapted checkpoints already track the target, so the gate suppresses the correction and the average metrics move by less than 0.7%, the intended behaviour of an evidence-driven correction rather than a failure. Complete results are in Appendix B.2.

4.2 ANALYSIS

Ablation of the core components. Table 3 reports absolute MSE and MAE at $H = 96$, averaged over the four TSFMs, for the complete method and for five variants that either remove one component or replace one stored quantity. Every component is needed: the complete method is the best of the six rows in all 14 dataset–metric columns, and every variant is worse on the average of both metrics, the two replacements losing the most. Appendix B.3 breaks the variants down dataset by dataset, and Appendix B.4 compares GRAM with a retrieval-augmented method that reuses the futures of similar windows.

Reliability of the correction. The gate and the two fixed weights of Eq. equation 11 trade peak gain for coverage: the same correction without the graph, applying the pooled retrieval answer alone, improves both metrics in only 74 of the 112 settings and can lose up to 8.4% of MSE, whereas the deployed configuration improves both metrics in 107 of the 112 settings. The correction is therefore the conservative of the two, and the one whose gains hold in 33 more settings.

Diagnostics of the pattern-conditioned bias. Figure 2 examines the bias over all 112 settings and yields three insights. First, the bias is real and pattern-conditioned: the frozen TSFMs systematically overstate how far the series departs from its recent level, and that overstatement varies about three times more inside a prototype than between prototypes, so the nearest pattern does not determine

Table 3: Ablation at $H = 96$: absolute MSE / MAE, averaged over the four TSFMs and the seven datasets. The first three rows remove one module, the last two replace the stored quantity or the recent level; **red** marks the best average.

Variants	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Exchange		ECL	
	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae
w/o shape relation	0.412	0.421	0.318	0.362	0.378	0.371	0.196	0.268	0.192	0.223	0.101	0.227	0.149	0.243
w/o bias relation	0.427	0.411	0.331	0.353	0.362	0.389	0.205	0.276	0.185	0.212	0.098	0.234	0.142	0.252
w/o anchor	0.403	0.432	0.324	0.341	0.387	0.365	0.191	0.258	0.201	0.225	0.104	0.218	0.153	0.237
amplitude → mean residuals	0.438	0.431	0.327	0.378	0.396	0.382	0.212	0.285	0.194	0.235	0.113	0.245	0.147	0.241
recent level → naive ratio	0.409	0.413	0.342	0.392	0.371	0.360	0.214	0.295	0.189	0.221	0.125	0.258	0.151	0.253
GRAM (full)	**0.381**	**0.387**	**0.295**	**0.335**	**0.344**	**0.349**	**0.183**	**0.254**	**0.179**	**0.209**	**0.093**	**0.212**	**0.138**	**0.227**

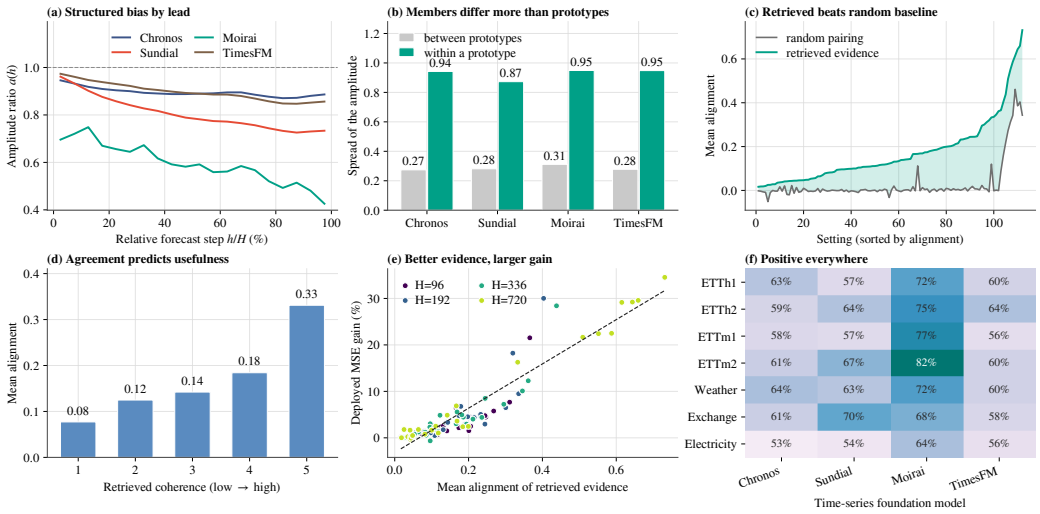

Figure 2: Diagnostics of the pattern-conditioned bias over all 112 settings. Panels (a)–(b) characterise the bias the memory stores, (c)–(e) the evidence retrieval returns and its link to the gain, and (f) the coverage across datasets and TSFMs.

the bias of a window and the memory has to aggregate related prototypes. Second, the retrieved evidence is useful: the alignment between the correction it implies and the realised error beats a random-pairing baseline in all 112 settings and predicts the per-setting MSE gain of Table 1 with a correlation of 0.93. Third, the evidence is available almost everywhere, with 53% to 82% of windows positively aligned depending on dataset and TSFM.

The remaining analyses, including the comparison with a retrieval-augmented method that reuses futures and the efficiency measurements, are provided in Appendix B.4 and Appendix B.5.

5 CONCLUSION

GRAM corrects the forecasts of a frozen time-series foundation model by retrieving, from the target history, the amplitude of the error that the model itself leaves on recurring patterns, which keeps the evidence dimensionless and comparable across windows. Across seven benchmarks, four horizons and four heterogeneous frozen models it improves both metrics on every dataset under all four TSFMs, reaching reductions of 34.5% in MSE and 16.7% in MAE on the weakest model at the longest horizon, and it adds only 7.7 to 12.2 microseconds per window on top of the frozen pipeline. Two directions follow naturally: updating the memory online as new windows arrive, and extending the bias retrieval to multivariate settings.

AI USE STATEMENT

We used ChatGPT to assist with polishing the manuscript and to improve the clarity of the writing. All AI-assisted content was carefully reviewed by the authors. All method design and experimental analysis were carried out independently by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

This work focuses on time-series forecasting research and does not involve human subjects or the collection of personal data. All experiments were conducted on existing benchmark datasets, following their established evaluation protocols. The proposed method is intended for general forecasting research, and we are not aware of any direct ethical risk arising from the method itself. We do not release any new dataset containing sensitive information or personally identifiable information.

REPRODUCIBILITY STATEMENT

The experimental setup is described in Section 4, and further dataset and implementation details are provided in Appendix A. The accompanying codebase contains the complete code of GRAM, together with all hyper-parameter settings, scripts and experimental configurations, and is available at `https://anonymous.4open.science/r/GRAM-B30A/`. All experimental results reported in this paper come from actual runs and faithfully reflect the corresponding experimental outcomes.

REFERENCES

Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *IEEE International Conference on Data Engineering*, pp. 3–14, 1995.

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Sundar Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21:116:1–116:6, 2020.

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. *Transactions on Machine Learning Research*, 2024, 2024.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, volume 162, pp. 2206–2240, 2022.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th edition, 2015.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramírez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *AAAI Conference on Artificial Intelligence*, pp. 6989–6997, 2023.

Si-An Chen, Chun-Liang Li, Sercan Ö. Arik, Nathanael C. Yoder, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023, 2023.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, volume 235, pp. 10148–10167, 2024.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Advances in Neural Information Processing Systems*, 2024.

Valentin Flunkert, David Salinas, and Jan Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *arXiv preprint arXiv:1704.04110*, 2017.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Azul Garza and Max Mergenthaler Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Harry R. Glahn and Dale A. Lowry. The use of model output statistics (MOS) in objective weather forecasting. *Journal of Applied Meteorology*, 11(8):1203–1211, 1972.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, volume 235, pp. 16115–16152, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, volume 70, pp. 1321–1330, 2017.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan Ö. Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *International Conference on Machine Learning*, volume 267, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, 2020.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6555–6565, 2024.

Bryan Lim, Sercan Ömer Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*, 2019.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022a.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024a.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, 2022b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024b.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *International Conference on Machine Learning*, volume 235, pp. 32369–32399, 2024c.

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 *International Conference on Machine Learning*, volume 267, 2025.

Abdullah Mueen, Eamonn J. Keogh, Qiang Zhu, Sydney Cash, and M. Brandon Westover. Exact discovery of time series motifs. In *SIAM International Conference on Data Mining*, pp. 473–484, 2009.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Y. Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. In *Advances in Neural Information Processing Systems*, 2025.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17 (9):2363–2377, 2024.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Bilos, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, Sahil Garg, Alexandre Drouin, Nicolas Chapados, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

D. Sculley. Web-scale k-means clustering. In *International Conference on World Wide Web*, pp. 1177–1178, 2010.

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 *International Conference on Learning Representations*, 2025.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, volume 235, pp. 53140–53164, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. *arXiv preprint arXiv:2412.16643*, 2024.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Advances in Neural Information Processing Systems*, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, pp. 11121–11128, 2023.

Huanyu Zhang, Chang Xu, Yifan Zhang, Zhang Zhang, Liang Wang, and Jiang Bian. Timeraf: Retrieval-augmented foundation model for zero-shot time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5654–5665, 2025.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence*, pp. 11106–11115, 2021.

Shiqiao Zhou, Holger Schöner, Zipeng Wu, Edouard Fouché, IAG Wilson, and Shuo Wang. Stationarity-aware retrieval-augmented time series forecasting. *arXiv preprint arXiv:2606.04135*, 2026.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, volume 162, pp. 27268–27286, 2022.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems*, 2023.

Table 4: Benchmark characteristics and splitting protocol. The proportions are train/validation/test, are the published ones and are chronological. The memory-construction and calibration portions are cut from the training portion, and the reported numbers are computed on the test portion.

Dataset	Domain	Frequency	Channels	Split (train/val/test)
ETTh1	Electricity transformer	Hourly	7	6:2:2
ETTh2	Electricity transformer	Hourly	7	6:2:2
ETTm1	Electricity transformer	15 minutes	7	6:2:2
ETTm2	Electricity transformer	15 minutes	7	6:2:2
Electricity	Household consumption	Hourly	321	7:1:2
Weather	Meteorology	10 minutes	21	7:1:2
Exchange Rate	Foreign exchange	Daily	8	7:1:2

A Experimental Details

A.1 Datasets and evaluation protocol

Table 4 lists the seven benchmarks and their split proportions. All of them are used exactly as published, with their canonical chronological splits Lai et al. (2018); Wu et al. (2021); Nie et al. (2023); Zeng et al. (2023); Liu et al. (2024b); Wang et al. (2024); Qiu et al. (2024); Godahewa et al. (2021); Aksu et al. (2024); Alexandrov et al. (2020), and both the memory-construction and the calibration portions are cut from the training portion, separated from each other and from the published test portion by a gap equal to the forecast horizon, so that no future observation of one portion can enter a window of the next; the test portion is never touched before evaluation. A separate memory is built for every dataset, horizon and TSFM, while the retrieval encoder and the graph fusion module are shared across all of them, and each channel is forecast independently.

A.2 Metrics

We report mean squared error (MSE) and mean absolute error (MAE), the two metrics used by previous studies on these benchmarks. Both are computed per channel against the un-normalised ground truth and then averaged over channels, and results are reported per dataset and horizon; when a table reports a single number for a dataset, as Table 1 and Table 7 do, it is the mean of the four horizon values of that dataset.

A.3 Memory construction and read-out settings

The offline phase has no optimisation: one pass over the memory-construction portion of a target cell writes the record of Section 3.2.2, whose fields are the ones Eq. equation 2 to Eq. equation 4 define. Its principal settings are $C = 128$ prototypes, a recent level of $S = 24$ observations, a member weight of $|\mathrm{exc}|$, a support floor of 8 for bias edges and $L = 2$ neighbours per relation. Online, the $K = 3$ closest prototypes seed a subgraph of about 21 nodes, each expanded by three neighbours per relation. The read-out has width 96, two message-passing layers, four attention heads and dropout 0.2, and enters the correction through the anchor weight $\mu = 0.35$ and the residual weight $\nu = 0.15$.

The read-out is fitted once, for 1500 steps with AdamW and a cosine schedule and three seeds, on the LOTSA corpus Woo et al. (2024), a multi-domain pretraining corpus unrelated to the seven evaluation benchmarks, and is then frozen; no series, forecast or label of an evaluation dataset takes part in that fitting, the frozen TSFM and the retrieval encoder are never updated, and the memory of a target cell is built only from the observed training portion of that cell. GRAM is therefore a zero-shot method on the evaluation benchmarks: *zero-shot* throughout this paper means that the frozen TSFM is used as published, the correction is fitted on an unrelated corpus, and no evaluation-domain data of any kind enters training. All measurements are taken on a single H200 with batches of 4096 windows, and Table 8 reports the resulting cost of 7.7–12.2 microseconds per window.

A.4 Retrieval and the graph fusion module

At inference the query embedding is compared with the C centroids and the subgraph of Section 3.3.1 is assembled from the seeds, the corpus prior and the anchor node: $K = 3$ seeds, each

expanded by three neighbours per relation, which yields 21 nodes on average. The record of a node is the five-tuple of Section 3.2.2, extended by the log support relative to the median prototype, the coherence, the retrieval similarity and the support flag; the query feature is the scale-normalised excursion of Eq. equation 1 followed by its log scale, its standard deviation and its maximum. The graph fusion module keeps the three blocks of Section 3.3.2 at width 96, 4 attention heads and dropout 0.2 Srivastava et al. (2014). Eq. equation 8 is implemented as written: each term is an attention-weighted average over the subgraph, the four gains $\kappa_i = \mathrm{softplus}(\cdot)$ start at $\log 2$, the bias b starts at 2, and a subgraph whose nodes disagree about the scaling cannot be capped permissively. The deployed ratio is the one of Eq. equation 11, with the anchor weight $\mu = 0.35$ and the residual weight $\nu = 0.15$; the gain head's last layer starts at zero so that $g \equiv 1$ and training begins from the fixed terms plus the vote, while the gate begins permissive and is only trained down where the evidence is thin. The head is fitted once on the LOTSA corpus Woo et al. (2024), whose series are split into memory and calibration portions with a fixed seed, and is then frozen for all target cells; the frozen TSFM and the retrieval encoder are never updated, and no label of an evaluation domain is used at any point. The fitting objective is the error of the corrected forecast—a squared and an absolute term—plus a penalty that keeps the returned amplitude close to the pooled retrieval answer.

A.5 Baselines and fine-tuning protocol

The baseline throughout is the frozen TSFM itself, whose zero-shot forecast is compared with the same forecast corrected by GRAM, on the same dataset and horizon and with the same metric. The parameter-efficient baselines are LoRA adapters optimised Kingma & Ba (2015) with each TSFM's native training loss on chronological training windows; the adapter strength and the checkpoint are selected on a later chronological tail of the training split, and the selected model is frozen before the memory is built, so that the fine-tuned setting follows the same contract as the zero-shot one.

A.6 Protocol for the diagnostics

The diagnostics of Figure 2 cover all 112 settings. To keep the pass cheap, each cell contributes at most 2000 test windows drawn uniformly over the test portion, and the forecast steps are thinned to 96 uniformly spaced leads before horizons are pooled on the relative axis h/H; alignments, spreads and counts are computed on that thinned grid, and the amplitude of a window is its $|\mathrm{exc}|$-weighted average. The random-pairing baseline permutes the retrieved evidence across test windows and averages five permutations. The quality of the retrieved evidence is measured by the cosine between the correction it implies and the realised error, which the main text calls its alignment.

A.7 Compute

Building the memory is a single pass over the target history: $O(n \cdot C)$ comparisons for the assignment, then one pass per prototype to aggregate its members. A query costs one comparison against the centroids, the assembly of a 21-node subgraph, and the forward pass of the graph fusion module, whose parameters are three orders of magnitude below the frozen TSFM's. The memory itself stores C records of H amplitude values plus five scalars, that is $128 \times H$ floating point numbers per dataset, horizon and TSFM, which is negligible next to the frozen model.

B Experimental Results

B.1 Complete results over the 112 settings

Table 5 lists every dataset, horizon and TSFM, and reads as the settings behind Table 1: both metrics improve in 107 of the 112 settings, the largest reductions reach 34.5% of MSE and 16.7% of MAE (Moirai on Electricity at $H = 720$, from 1.476 to 0.966 MSE), fourteen settings remove at least a tenth of the MSE and ten at least a fifth, and the gains grow with the horizon and are largest where the frozen model is weakest, which the four column blocks make visible: the Moirai block is the largest of the four and the TimesFM block the smallest.

Figure 3 is the window-level view at $H = 96$: four frozen models in the rows, three datasets in the columns, and one test window per column. The windows are picked for the two properties a case

Table 5: Complete zero-shot results behind Table 1, every dataset and horizon. For each TSFM, *Ori* is the bare frozen model and +GRAM the corrected forecast; MSE is compared with MSE and MAE with MAE, and **red** marks the better of the two.

Models		Sundial				Chronos				TimesFM				Moirai			
		Ori		+GRAM (Ours)		Ori		+GRAM (Ours)		Ori		+GRAM (Ours)		Ori		+GRAM (Ours)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.356	0.383	**0.352**	**0.381**	0.385	0.380	**0.376**	**0.378**	0.384	0.382	**0.380**	**0.381**	0.426	0.411	**0.416**	**0.408**
	192	0.390	0.406	**0.384**	**0.403**	0.442	0.412	**0.427**	**0.409**	0.428	0.408	**0.421**	**0.406**	0.467	0.433	**0.454**	**0.429**
	336	0.403	0.416	**0.400**	**0.415**	0.481	0.433	**0.467**	**0.430**	0.446	0.419	**0.441**	**0.418**	0.478	0.446	**0.463**	**0.443**
	720	0.438	0.444	**0.420**	**0.433**	0.483	0.449	**0.475**	**0.439**	**0.428**	0.427	0.428	**0.423**	1.112	0.725	**0.787**	**0.612**
ETTh2	96	0.279	0.334	**0.274**	**0.329**	0.290	0.325	**0.286**	**0.324**	0.306	0.338	**0.301**	**0.335**	0.332	0.361	**0.317**	**0.352**
	192	0.342	0.376	**0.340**	**0.373**	0.360	0.370	**0.358**	**0.369**	0.376	0.384	**0.372**	**0.381**	0.413	0.411	**0.395**	**0.401**
	336	**0.361**	0.395	0.363	**0.395**	0.403	**0.403**	**0.403**	0.403	0.401	0.409	**0.399**	**0.408**	0.445	0.438	**0.426**	**0.429**
	720	0.392	0.426	**0.379**	**0.422**	0.414	0.421	**0.401**	**0.412**	0.405	0.422	**0.403**	**0.416**	0.615	0.541	**0.482**	**0.478**
ETTm1	96	0.292	0.325	**0.288**	**0.325**	0.332	0.333	**0.323**	**0.332**	0.313	0.329	**0.310**	**0.329**	0.479	0.417	**0.457**	**0.410**
	192	0.342	0.357	**0.336**	**0.356**	0.387	0.366	**0.377**	**0.364**	0.361	0.359	**0.357**	**0.359**	0.517	0.440	**0.491**	**0.432**
	336	0.378	0.380	**0.371**	**0.379**	0.436	0.393	**0.426**	**0.392**	0.394	0.382	**0.390**	**0.382**	0.651	0.472	**0.596**	**0.464**
	720	0.451	0.418	**0.443**	**0.408**	0.528	0.439	**0.518**	**0.428**	0.456	0.418	**0.454**	**0.410**	1.314	0.763	**0.925**	**0.633**
ETTm2	96	0.176	0.255	**0.168**	**0.249**	0.178	0.245	**0.175**	**0.243**	0.179	0.246	**0.176**	**0.244**	0.232	0.291	**0.214**	**0.281**
	192	0.239	0.298	**0.228**	**0.290**	0.251	0.294	**0.248**	**0.291**	0.249	0.293	**0.245**	**0.291**	0.333	0.344	**0.302**	**0.330**
	336	0.297	0.336	**0.284**	**0.327**	0.317	0.336	**0.313**	**0.334**	0.314	0.336	**0.308**	**0.332**	0.448	0.402	**0.393**	**0.381**
	720	0.388	0.393	**0.374**	**0.384**	0.416	0.395	**0.413**	**0.393**	0.420	0.396	**0.409**	**0.391**	0.601	0.514	**0.466**	**0.449**
ECL	96	0.118	0.213	**0.117**	**0.213**	0.124	0.211	**0.124**	**0.211**	0.133	0.220	**0.132**	**0.220**	0.180	0.266	**0.179**	**0.266**
	192	0.135	0.231	**0.134**	**0.230**	0.143	0.229	**0.143**	**0.229**	0.150	0.236	**0.149**	**0.236**	0.202	0.281	**0.200**	**0.280**
	336	0.150	0.247	**0.150**	**0.247**	0.161	0.248	**0.161**	**0.248**	0.167	0.253	**0.167**	**0.253**	0.223	0.296	**0.220**	**0.295**
	720	0.188	0.280	**0.188**	**0.280**	0.207	0.288	**0.206**	**0.288**	0.206	0.287	**0.206**	**0.287**	1.476	0.942	**0.966**	**0.785**
Weather	96	0.160	0.207	**0.156**	**0.203**	0.178	0.210	**0.175**	**0.209**	0.163	0.194	**0.161**	**0.193**	0.240	0.231	**0.226**	**0.229**
	192	0.213	0.255	**0.203**	**0.247**	0.224	0.254	**0.218**	**0.249**	0.216	0.242	**0.208**	**0.239**	0.351	0.281	**0.287**	**0.272**
	336	0.267	0.295	**0.256**	**0.287**	0.284	0.295	**0.276**	**0.291**	0.289	0.289	**0.275**	**0.284**	0.357	0.340	**0.321**	**0.324**
	720	0.336	0.344	**0.328**	**0.337**	0.367	0.347	**0.358**	**0.343**	0.385	0.345	**0.366**	**0.340**	0.549	0.479	**0.426**	**0.416**
Exchange	96	0.132	0.249	**0.104**	**0.223**	0.099	0.219	**0.093**	**0.211**	0.089	0.208	**0.087**	**0.205**	0.093	0.213	**0.089**	**0.208**
	192	0.291	0.380	**0.204**	**0.320**	0.194	0.314	**0.181**	**0.302**	0.191	0.312	**0.182**	**0.305**	0.198	0.314	**0.185**	**0.304**
	336	0.544	0.535	**0.389**	**0.454**	0.342	0.424	**0.325**	**0.412**	0.351	0.431	**0.331**	**0.418**	0.368	0.434	**0.342**	**0.420**
	720	1.265	0.830	**1.059**	**0.765**	0.847	0.690	**0.832**	**0.684**	0.984	0.731	**0.916**	**0.712**	1.744	0.993	**1.236**	**0.841**

study is meant to show at once: before the correction every model is visibly off the label, and after it the corrected forecast sits close to the label. Every panel removes between 25% and 63% of the squared error, and in every panel the corrected forecast is closer to the label than the frozen one at most forecast steps.

B.2 PARAMETER-EFFICIENT ADAPTATION

Training LoRA adapters on the target history shortens the distance the correction has to cover, and GRAM still helps afterwards: averaged over the four horizons, both metrics improve on all seven datasets under Moirai and on six of them under Chronos and Sundial, and the reduction follows the pattern of the zero-shot setting, largest where the adapted model is weakest. TimesFM is the exception, at -0.68% MSE and -0.29% MAE: its adapted checkpoints already track the target so closely that the gate suppresses the correction, which is the intended behaviour rather than a failure. The largest single gain is fine-tuned Moirai on ETTh1 at $H = 720$, where the MSE falls from 1.077 to 0.753, a 30.1% reduction. Table 6 gives the same comparison for every dataset and horizon.

B.3 COMPONENT ABLATIONS

Table 3 removes one component at a time at $H = 96$, averaged over the four TSFMs and reported as the relative improvement over the frozen model in each dataset. The complete method is the best of the six rows in all 14 dataset–metric columns. Removing either relation of the prototype graph lowers both averages (3.47% and 3.32% MSE against 3.58% for the full method), which is the performance side of the structural result in Figure 2(b): the members of a prototype differ more than the prototypes differ from each other, so evidence has to be pooled across related prototypes. Removing the anchor also lowers both averages, because the head then loses the retrieval baseline it can fall back on. Replacing the amplitude curves by mean residual curves removes most of the benefit, leaving 0.29% MSE and -0.91% MAE, since borrowed residuals track the errors of their own windows rather than those of the query. Replacing the recent level of the query by the naive ratio between future and forecast performs worse than the frozen model on average (-0.90% MSE, -2.91% MAE), and the columns it wins come at the cost of the other metric: it leads the Weather column on MSE while losing 2.40% MAE there.

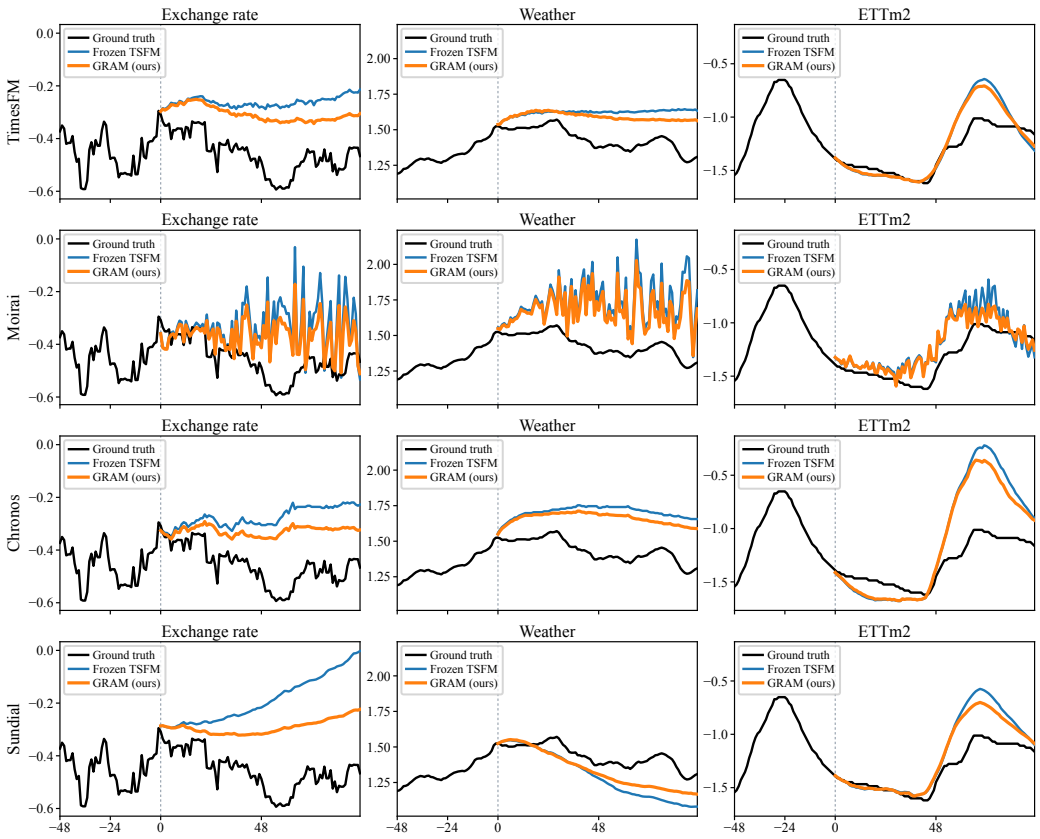

Figure 3: Four frozen models (one per row, named on the left: TimesFM, Moirai, Chronos, Sundial) on three datasets (one per column) at $H = 96$. Each panel is a single test window — the same window for the rows of a column — with the last 48 steps of the series before the dotted line and the 96-step forecast after it; the two forecasts are drawn only inside the forecast window, and all values are standardised. The windows are chosen so that every model is clearly off before the correction (frozen RMSE between 18% and 55% of the panel's own vertical range) and close after it (residual at most 42% of that range, +25% to +63% of the squared error); among the windows that qualify, the one drawn is the window whose worst-case residual is the smallest. The Moirai row is the jitteriest of the four, which is the model's own behaviour on these series rather than the correction's.

The two structural variants are close on the average, which is expected: removing one relation of degree two leaves the other one to supply most of the neighbourhood, and the per-dataset columns show where each relation carries the larger share, as on ETTh2 and ETTm2 for the shape relation.

B.4 Comparison with the state-of-the-art retrieval-augmented method

The related family closest to ours reuses the ground-truth futures of the historical windows that are most similar to a query. TS-RAG Ning et al. (2025), the state of the art of that family, rewrites the query with the futures of its nearest neighbours, and Table 7 reports it together with GRAM on the seven datasets of Table 1; every entry is produced by the implementation of the method it belongs to with the frozen model that method is built for, and each is averaged over the four horizons of the dataset.

The two averages are close—MSE is 0.3115 for GRAM against 0.3151 for TS-RAG, and MAE 0.3463 against 0.3409, both within two percent—so the difference lies in what the two carry across, and there GRAM is the better of the two on six of the seven datasets in MSE and on five in MAE, and over the 28 individual settings it is the better one in 24 of them in MSE and in 15 in MAE. TS-RAG is ahead on Exchange, the series closest to a random walk, which is the one setting where reusing

Table 6: Complete results behind Table 2, every dataset and horizon. For each TSFM, *FT* is the LoRA fine-tuned model and +GRAM the same fine-tuned model corrected by GRAM; MSE is compared with MSE and MAE with MAE, and **red** marks the better of the two.

Models		Sundial				Chronos				TimesFM				Moirai			
		FT		+GRAM (Ours)		FT		+GRAM (Ours)		FT		+GRAM (Ours)		FT		+GRAM (Ours)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.368**	**0.387**	0.371	0.388	0.411	**0.387**	**0.400**	0.387	0.414	0.411	**0.414**	**0.411**	0.410	0.403	**0.402**	**0.401**
	192	0.394	0.408	**0.393**	**0.407**	0.461	0.416	**0.441**	**0.413**	0.518	0.451	**0.518**	**0.451**	0.437	0.420	**0.424**	**0.416**
	336	**0.407**	**0.418**	0.410	0.418	0.488	0.434	**0.473**	**0.432**	0.602	0.479	**0.602**	**0.479**	0.454	0.433	**0.442**	**0.431**
	720	**0.444**	**0.449**	0.470	0.455	**0.483**	**0.448**	0.492	0.455	0.421	0.409	**0.421**	**0.409**	1.077	0.715	**0.753**	**0.599**
ETTh2	96	0.267	0.323	**0.263**	**0.320**	0.281	0.320	**0.274**	**0.317**	**0.287**	**0.332**	0.288	0.332	0.311	0.343	**0.303**	**0.339**
	192	0.336	0.369	**0.334**	**0.367**	0.361	0.372	**0.357**	**0.371**	0.371	0.385	**0.371**	**0.384**	0.395	0.394	**0.389**	**0.391**
	336	**0.362**	**0.393**	0.364	0.393	0.415	**0.411**	**0.412**	0.411	0.397	0.409	**0.396**	**0.408**	0.409	0.413	**0.404**	**0.411**
	720	0.386	0.420	**0.382**	**0.417**	0.445	0.434	**0.437**	**0.433**	0.408	0.426	**0.407**	**0.425**	0.612	0.538	**0.480**	**0.476**
ETTm1	96	0.267	0.311	**0.262**	**0.311**	0.259	0.302	**0.255**	**0.301**	0.222	0.303	**0.220**	**0.302**	0.303	0.333	**0.296**	**0.330**
	192	0.326	0.349	**0.316**	**0.347**	0.319	0.338	**0.312**	**0.336**	0.241	0.326	**0.234**	**0.323**	0.365	0.367	**0.355**	**0.365**
	336	0.368	0.374	**0.359**	**0.373**	0.367	0.366	**0.359**	**0.364**	**0.262**	0.355	0.263	**0.355**	0.404	0.391	**0.392**	**0.388**
	720	0.444	0.415	**0.436**	**0.413**	0.449	0.409	**0.437**	**0.407**	**0.491**	**0.434**	0.497	0.436	1.193	0.714	**0.865**	**0.607**
ETTm2	96	0.172	0.252	**0.165**	**0.245**	0.167	0.235	**0.166**	**0.234**	**0.165**	0.236	0.165	**0.236**	0.194	0.257	**0.185**	**0.253**
	192	0.238	0.297	**0.227**	**0.289**	0.237	0.283	**0.237**	**0.282**	0.232	0.282	**0.231**	**0.282**	0.251	0.298	**0.243**	**0.293**
	336	0.297	0.335	**0.283**	**0.326**	0.300	0.326	**0.300**	**0.325**	0.291	0.322	**0.290**	**0.321**	0.311	0.336	**0.301**	**0.332**
	720	0.388	0.394	**0.374**	**0.385**	0.397	0.389	**0.395**	**0.387**	0.386	0.381	**0.385**	**0.380**	0.518	0.475	**0.444**	**0.437**
Weather	96	0.154	0.202	**0.148**	**0.196**	0.142	0.175	**0.140**	**0.174**	0.142	0.179	**0.141**	**0.178**	0.158	0.188	**0.155**	**0.187**
	192	0.211	0.254	**0.200**	**0.245**	0.191	0.221	**0.188**	**0.220**	0.190	0.225	**0.190**	**0.225**	0.211	0.236	**0.207**	**0.234**
	336	0.275	0.301	**0.256**	**0.288**	0.261	0.267	**0.250**	**0.264**	0.249	0.270	**0.248**	**0.269**	0.281	0.289	**0.273**	**0.285**
	720	0.356	0.359	**0.335**	**0.346**	0.351	0.325	**0.336**	**0.322**	0.334	0.327	**0.332**	**0.326**	0.469	0.439	**0.390**	**0.395**
Exchange	96	0.105	0.225	**0.096**	**0.215**	0.101	0.219	**0.093**	**0.208**	**0.103**	**0.224**	0.109	0.231	0.093	0.212	**0.090**	**0.208**
	192	0.227	0.335	**0.198**	**0.311**	0.205	0.319	**0.193**	**0.306**	**0.196**	**0.315**	0.207	0.323	0.197	0.313	**0.187**	**0.305**
	336	0.442	0.481	**0.374**	**0.440**	0.369	0.433	**0.351**	**0.420**	**0.353**	**0.429**	0.377	0.444	0.377	0.439	**0.354**	**0.424**
	720	1.211	0.808	**1.059**	**0.761**	0.908	0.710	**0.864**	**0.695**	**0.950**	**0.723**	1.011	0.741	1.728	0.983	**1.251**	**0.845**
ECL	96	0.116	0.212	**0.116**	**0.211**	0.122	**0.209**	**0.122**	0.209	0.127	0.216	**0.127**	**0.216**	0.171	**0.263**	**0.171**	0.263
	192	0.133	0.230	**0.133**	**0.229**	0.141	0.227	**0.141**	**0.227**	0.144	0.232	**0.144**	**0.232**	0.189	**0.277**	**0.188**	0.277
	336	0.149	0.247	**0.148**	**0.246**	0.160	0.247	**0.160**	**0.247**	0.162	0.251	**0.162**	**0.250**	0.205	0.292	**0.204**	**0.292**
	720	0.188	0.280	**0.187**	**0.280**	0.205	0.287	**0.205**	**0.287**	0.203	0.287	**0.202**	**0.286**	1.301	0.888	**0.941**	**0.774**

Table 7: Comparison with the retrieval-augmented method that reuses the ground-truth futures of similar historical windows, TS-RAG Ning et al. (2025), the state of the art of that family. Every entry is the average over the four horizons of the dataset; the better of the two methods is **red** in each column.

		ETTh1	ETTh2	ETTm1	ETTm2	Weather	Exchange	ECL
TS-RAG	MSE	0.434	0.348	0.395	0.282	0.258	**0.332**	0.157
	MAE	0.413	**0.375**	0.374	0.314	0.274	**0.392**	0.244
GRAM (Ours)	MSE	**0.394**	**0.341**	**0.360**	**0.264**	**0.236**	0.439	**0.147**
	MAE	**0.411**	0.380	**0.369**	**0.313**	**0.269**	0.440	**0.242**

futures is the stronger choice. A reused future brings the recurring bias together with the sample-specific part that the decomposition of Section 3 argues is not transferable, so its value depends on how much of that part is shared between the retrieved window and the query; the dimensionless ratio that GRAM transfers does not, which is why its placement is the more even one across settings.

B.5 EFFICIENCY AND MEMORY BUDGET

The read-out is the only trained part and stays below a million parameters at every horizon, so the correction never dominates the cost of a backbone forward pass: the frozen backbones carry two to three orders of magnitude more parameters (128M–2B against 0.3–0.6M). What has to be stored per dataset, TSFM and horizon is the prototype memory—$C = 128$ records of H amplitudes and H spreads plus a handful of scalars, the centroid of each record being also the key it is retrieved by. Retrieval itself is a comparison against C centroids followed by the assembly of a 21-node subgraph, and the head is a small graph network run once per window, so the whole correction costs between 7.7 and 12.2 microseconds per window on an H200.

B.6 LIMITATIONS AND FUTURE WORK

The offline phase is a single pass over the target history, and the read-out is fitted on the memory-construction and calibration portions before the test stream starts, so both costs are one-off; amor-

Table 8: Efficiency of the correction, measured on an H200 with batches of 4096 windows. The read-out is the only trained part; the memory is what has to be stored next to the frozen backbone. Retrieval is a comparison against $C = 128$ centroids and stays below $0.01\,\mu$s per window, so the read-out dominates the measured cost.

	$H = 96$	$H = 192$	$H = 336$	$H = 720$
read-out parameters (M)	0.26	0.32	0.40	0.63
prototype records (KB)	98	194	338	722
read-out cost (μs / window)	7.7	8.7	9.4	12.2

tising one read-out across the four horizons instead of fitting each of them separately is a natural next step. The prototype library is built from the target series itself, and the retrieval pipeline itself is corpus-agnostic, so a memory that a frozen backbone can arrive with—built on one corpus and reused on another—is the direction we consider most promising. Our study covers seven long-standing benchmarks, four foundation models and the four standard horizons, all as deterministic point forecasts; probabilistic heads, other model families, and higher-frequency or geographically distributed series are left to future work.