ROAR: RETRIEVAL OPPORTUNITY-AWARE REFINEMENT FOR ZERO-SHOT TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval augmentation provides time series forecasters with historical continuations, yet even candidates that outperform the base forecast may fail to improve the final prediction. We propose ROAR, a Retrieval Opportunity-Aware Refinement framework for zero-shot time series forecasting. To better exploit these improvement opportunities, its training objective allocates additional emphasis across queries based on base-forecast difficulty and the relative improvement offered by retrieved candidates. Using this objective, ROAR first learns to aggregate aligned historical candidates and uses a learned gate to control their correction strength against a fixed base forecaster. It then jointly calibrates the forecasting module and gate to coordinate their contributions, while anchoring the combined prediction to the first-stage refined output. We derive exact decompositions of refinement gains and the opportunity-weighted training loss, and characterize additional gains from joint calibration under a local linearization. Experiments on seven benchmarks show that ROAR achieves the lowest average MSE among the compared methods. Further evaluations demonstrate average forecasting improvements across multiple backbone families and retrieval-augmented forecasters.

1 INTRODUCTION

Time series forecasting predicts future observations from historical data and supports applications including weather prediction (Sun et al., 2022; Sapankevych & Sankar, 2009), energy consumption modeling (Tzelepi et al., 2023), healthcare (Jin et al., 2018; Tomar et al., 2025), and financial analysis (Qiu et al., 2025a;b). Recently, Time Series Foundation Models (TSFMs) have demonstrated promising zero-shot forecasting capabilities by learning temporal patterns from large and diverse pretraining corpora (Das et al., 2024; Ansari et al., 2024). Nevertheless, their performance can deteriorate when the patterns encountered at inference differ from those represented during training (Kim et al., 2025; Zhou et al., 2023; Wang et al., 2024).

Retrieval augmentation complements these learned patterns with historical examples relevant to the current input (Lewis et al., 2020; Zhao et al., 2026; Wu et al., 2026). Each historical example contains an observed context and its subsequent trajectory, providing a candidate continuation for the current query. Previous studies have explored adaptive combinations of retrieved forecasts and model outputs, as well as corrections based on retrieved historical residuals (Du et al., 2026; He et al., 2026). These developments motivate a key question: *how can useful historical candidates be translated into improvements over an existing forecast?*

To investigate this question, we conduct an empirical analysis of candidate usefulness and realized retrieval gains. We examine TS-RAG (Ning et al., 2025) and Cross-RAG (Lee et al., 2026) using Chronos-Bolt as a common frozen backbone, and evaluate each context-aligned historical continuation as an individual candidate forecast. Fig. 1 shows that at least one candidate has lower MSE than Chronos-Bolt for 46% and 35% of queries, respectively. Yet for 17% and 7% of all queries, respectively, the retrieval-augmented forecast has higher MSE than Chronos-Bolt despite the availability of a better candidate. The trajectory examples in Fig. 1(c,d) illustrate successful correction and a missed opportunity to improve the forecast. These findings reveal a gap between the availability of useful historical evidence and its effective integration into the final forecast. Further details appear in Appendix A.

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.

Motivated by this gap, we propose a **R**etrieval **O**pportunity-**A**ware **R**efinement framework (**ROAR**) that learns retrieval-based forecast corrections on source data for zero-shot transfer. Its opportunity-aware objective allocates additional training emphasis across queries according to base-forecast difficulty and the relative improvement offered by retrieved candidates. ROAR uses this objective to learn candidate aggregation and correction strength while keeping the base forecaster fixed. Reference-guided joint calibration then updates the trainable forecasting module and gate, anchoring their combined prediction to the initially refined forecast.

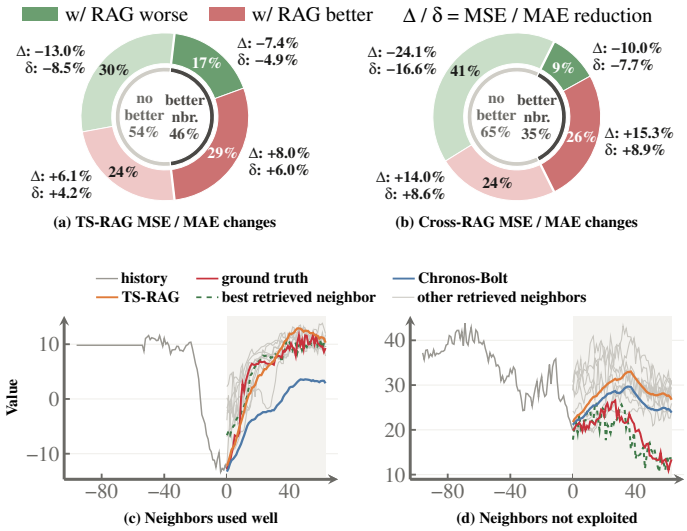


Figure 1: **Useful candidates do not always improve forecasts.** (a, b) MSE changes of TS-RAG and Cross-RAG relative to Chronos-Bolt, split by whether a better candidate exists. (c, d) TS-RAG examples of a successful correction and a missed opportunity.

Our contributions are summarized as follows:

- We provide an empirical diagnosis of retrieval-augmented forecasting, revealing that useful historical candidates can coexist with degraded forecasts.
- We introduce ROAR, which allocates training emphasis across queries according to candidate improvement opportunities and base-forecast difficulty, and coordinates the base prediction with learned corrections through reference-guided joint calibration.
- We derive exact decompositions of refinement gains and the opportunity-weighted forecasting loss. Under local linearization, we characterize the MSE advantage of joint over single-component calibration at their reference-regularized optima.
- We evaluate ROAR on seven benchmarks in zero-shot forecasting, demonstrating the lowest average MSE among the compared methods and improvements across multiple backbone families and retrieval-augmented forecasters.

2 Related Work

Time Series Foundation Models. Time Series Foundation Models (TSFMs) learn transferable temporal patterns from large, heterogeneous corpora, supporting forecasting with limited or no task-specific adaptation (Liang et al., 2024). Chronos (Ansari et al., 2024) discretizes numerical observations into tokens and adopts language modeling objectives, whereas TimesFM (Das et al., 2024) operates on continuous-valued patches with a decoder-only architecture. Moirai (Woo et al., 2024) addresses heterogeneity in frequency, dimensionality, and marginal distributions. Other developments include mixture-of-experts modeling in Moirai-MoE (Liu et al., 2025b), masked modeling in MOMENT (Goswami et al., 2024), generative probabilistic forecasting in Sundial (Liu et al., 2025c), and scalable pretraining in Timer-S1 (Liu et al., 2026).

Retrieval-Augmented Forecasting. Retrieval augmentation supplies forecasters with historical examples relevant to the current input. RAF (Tire et al., 2026) studies retrieval augmentation for pretrained forecasters, while RAFT (Han et al., 2025) incorporates historical sequences and their continuations into a multi-resolution forecasting framework. TimeRAF (Zhang et al., 2025) supervises candidate ranking with forecasting feedback and integrates retrieved representations through channel prompting. TS-RAG (Ning et al., 2025) adaptively combines query and retrieval information, while Cross-RAG (Lee et al., 2026) uses cross-attention between the query and retrieved representations to incorporate relevant information and reduce interference from irrelevant neighbors.

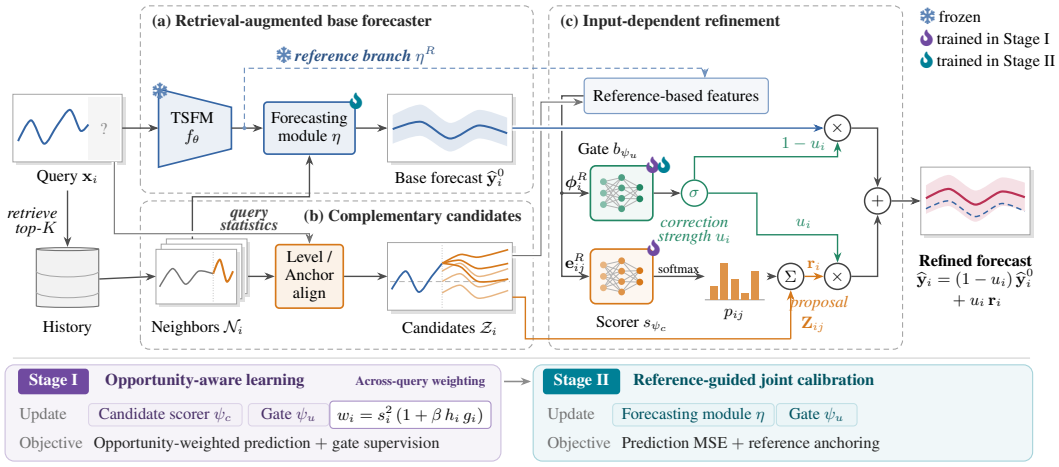


Figure 2: Overview of ROAR. (a–c) Aligned historical continuations refine the base forecast through learned candidate aggregation and input-dependent gating. The TSFM backbone remains frozen, and the scorer and gate use frozen reference features. Stage I applies opportunity-aware weights w_i across queries, emphasizing difficult queries with useful candidates. Stage II jointly calibrates the forecasting module and gate, anchoring their combined prediction to the Stage-I refined output.

Forecast Refinement and Example Reweighting. Retrieved evidence also supports refinement in prediction space. PFRP (Du et al., 2026) adaptively combines predictions from a global historical memory with local model forecasts. RATL (He et al., 2026) freezes the base forecaster and learns to combine retrieved historical residuals, training its router with prediction loss and oracle supervision derived from candidate correction errors. Both methods are developed and evaluated under dataset-specific supervised training settings. Example reweighting provides a complementary approach to allocating training effort. Focal loss downweights well-classified instances (Lin et al., 2017), while meta-learning methods learn example weights using validation objectives (Ren et al., 2018).

In contrast, our method combines prediction-space refinement with opportunity-aware reweighting across source-training queries for zero-shot transfer.

3 Our Method

We introduce a **R**etrieval **O**pportunity-**A**ware **R**efinement framework (**ROAR**) that refines a base forecast using retrieved historical continuations. It allocates additional training emphasis across queries based on base-forecast difficulty and the relative improvement offered by retrieved candidates. Fig. 2 provides an overview of ROAR's prediction framework and two-stage training procedure. Stage I keeps the base forecaster fixed and learns candidate aggregation and correction strength using opportunity-aware weighting. Stage II retains the learned aggregation and jointly calibrates the trainable forecasting module and gate, using a reference penalty to anchor their combined prediction to the Stage-I refined output.

Problem Setup. For a single channel, let $\mathbf{x}_i \in \mathbb{R}^L$ denote an observed context and $\mathbf{y}_i \in \mathbb{R}^H$ its future. Each query retrieves K historical examples $\mathcal{N}_i = \{(\mathbf{x}_{ij}^r, \mathbf{y}_{ij}^r)\}_{j=1}^K$, consisting of contexts and their observed continuations. Retrieval uses only continuations available at the forecast origin and excludes each source-training query's own target segment. The retrieval-augmented base forecaster combines a frozen foundation backbone with parameters θ and a trainable forecasting module with parameters η. We denote its point forecast by

$$\widehat{\mathbf{y}}_i^0(\eta) = f_{\theta,\eta}(\mathbf{x}_i, \mathcal{N}_i). \tag{1}$$

Let η^R denote the forecasting module's source-trained initialization before refinement and $\widehat{\mathbf{y}}_i^{0,R} = \widehat{\mathbf{y}}_i^0(\eta^R)$ its point forecast. The parameters θ remain frozen throughout. All trainable modules are optimized on source data and applied to target datasets without parameter updates. We denote the per-query forecasting loss by ℓ_i. Both training stages evaluate this loss on the original scale.

3.1 Overall Framework: Refinement with Historical Candidates

We first describe the prediction framework shared by both training stages. ROAR aligns retrieved continuations to the query and aggregates the resulting candidates into a proposal. A learned gate determines how far to adjust the base forecast toward this proposal. The two-stage optimization of this framework is described in Sections 3.2 and 3.3.

Context-Aligned Historical Candidates. Retrieved continuations can differ from the query in level and scale. We construct two aligned versions of each continuation, using the contextual mean and the latest observation as alternative reference levels. Let μ_i and $s_i > 0$ denote the temporal mean and stabilized standard deviation of $\mathbf{x}_i$. The corresponding statistics of the retrieved contexts are μ^r_{ij} and $s^r_{ij} > 0$ (where i indexes queries and j indexes neighbors retrieved for each query). The candidates are

$$\mathbf{Z}^{\text{lev}}_{ij} = \mu_i + s_i \frac{\mathbf{y}^r_{ij} - \mu^r_{ij}}{s^r_{ij}}, \qquad \mathbf{Z}^{\text{anc}}_{ij} = x_{i,L} + s_i \frac{\mathbf{y}^r_{ij} - x^r_{ij,L}}{s^r_{ij}}, \tag{2}$$

where $x_{i,L}$ and $x^r_{ij,L}$ are the last observed values of the query and retrieved contexts. Level alignment transfers deviations from the historical mean, while anchor alignment transfers changes relative to the last historical observation. We retain both aligned versions of each retrieved continuation, yielding $J = 2K$ candidates indexed by k: $\mathcal{Z}_i = \{\mathbf{Z}_{ik}\}_{k=1}^{J}$. Their values depend only on observed contexts and historical continuations.

Candidate Aggregation and Correction Gating. A candidate scorer determines the contribution of each historical forecast, while a separate gate controls correction strength. For candidate j of query i, we construct a feature vector $\mathbf{e}^R_{ij}$ from its retrieval distance, normalized deviation from the reference base forecast, and alignment type. The gate uses query-level features $\boldsymbol{\phi}^R_i$ summarizing reference-forecast uncertainty, retrieval distance, and candidate dispersion. These features are computed from the frozen reference branch and retrieved history. The aggregation weights and correction strength are

$$p_{ij} = \frac{\exp\big(s_{\psi_c}(\mathbf{e}^R_{ij})\big)}{\sum_{k=1}^{J} \exp\big(s_{\psi_c}(\mathbf{e}^R_{ik})\big)}, \qquad u_i = \sigma\big(b_{\psi_u}(\boldsymbol{\phi}^R_i)\big)\,, \tag{3}$$

where ψ_c and ψ_u parameterize the scorer and gate, respectively, and σ is the sigmoid function. The aggregated proposal is $\mathbf{r}_i = \sum_{k=1}^{J} p_{ik}\mathbf{Z}_{ik}$. We adjust the base forecast toward this proposal:

$$\boxed{\widehat{\mathbf{y}}_i = \widehat{\mathbf{y}}^0_i(\eta) + u_i\left(\mathbf{r}_i - \widehat{\mathbf{y}}^0_i(\eta)\right).} \tag{4}$$

The resulting forecast is a convex combination of the base prediction and historical candidates, with weights $1 - u_i$ and $u_i p_{ij}$.

In our quantile-output implementation, the base forecaster produces $\mathbf{q}^{0,\tau}_i(\eta)$ at quantile level τ. We use the median ($\tau = 0.5$) as the base point forecast $\widehat{\mathbf{y}}^0_i(\eta)$ and apply the same refinement correction to every quantile:

$$\widehat{\mathbf{q}}^{\tau}_i = \mathbf{q}^{0,\tau}_i(\eta) + \widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}^0_i(\eta). \tag{5}$$

Candidate aggregation operates within each query. The opportunity-aware objective below determines the additional training emphasis assigned across queries.

3.2 Stage I: Opportunity-Aware Learning

A large base error indicates room for improvement, but does not establish that an individual retrieved candidate offers a better forecast. Stage I therefore learns the scorer and gate with $\eta = \eta^R$ fixed, assigning additional training emphasis to queries that are both difficult and supported by beneficial historical candidates.

Query-Level Improvement Opportunity. Let Ω_i be the set of valid forecasting entries for query i and $n_i = \max\{|\Omega_i|, 1\}$. For a point-forecast vector $\mathbf{v} = (v_1, \ldots, v_H)^\top$, define its error as

$$\delta_i(\mathbf{v}) = \frac{1}{n_i} \sum_{t \in \Omega_i} (v_t - y_{i,t})^2. \tag{6}$$

Write $\delta_i^0 = \delta_i(\widehat{\mathbf{y}}_i^{0,R})$ for the reference base error and $\delta_i^{r,\min} = \min_{1\le k\le J} \delta_i(\mathbf{Z}_{ik})$ for the best individual candidate error. A threshold t_h on the normalized reference error identifies difficult queries: $h_i = \mathbb{1}[\delta_i^0/s_i^2 > t_h]$. The opportunity score and training weight are

$$g_i = \frac{[\delta_i^0 - \delta_i^{r,\min}]_+}{\delta_i^0 + \epsilon s_i^2}, \qquad w_i = s_i^2(1 + \beta h_i g_i), \tag{7}$$

where $[a]_+ = \max(a, 0)$, $\epsilon > 0$, and $\beta \ge 0$. The score g_i measures the relative error reduction available from the best individual candidate and is zero when no candidate improves on the base forecast. The factor s_i^2 sets the baseline weighting of the raw-scale forecasting loss, and $1 + \beta h_i g_i$ supplies the opportunity-dependent multiplier.

Learning the Refined Forecast. For a source-training mini-batch $\mathcal{B}$, we optimize the opportunity-weighted forecasting loss with auxiliary gate supervision:

$$\min_{\psi_c,\psi_u} \mathcal{L}_{\text{init}} = \frac{\sum_{i\in\mathcal{B}} \text{sg}(w_i)\ell_i}{\sum_{i\in\mathcal{B}} \text{sg}(w_i)} + \frac{\lambda_{\text{gate}}}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \text{BCE}(u_i, \text{sg}(z_i)) \,. \tag{8}$$

In our implementation, ℓ_i is twice the mean pinball loss over quantile levels and valid forecast entries, evaluated on the refined quantiles in Eq. 5. Here, BCE is binary cross-entropy, $\lambda_{\text{gate}} \ge 0$ controls the auxiliary term, and sg denotes stop-gradient. The normalized weight $w_i / \sum_{k\in\mathcal{B}} w_k$ determines query i's forecasting-loss contribution to learning both candidate aggregation and correction strength. The auxiliary target compares the current aggregated proposal ($u_i = 1$) with the reference base forecast ($u_i = 0$): $z_i = \mathbb{1}\left[\delta_i(\mathbf{r}_i) < \delta_i^0\right]$.

The label supervises a preference between the two endpoints. The combined objective determines the continuous correction strength. The score g_i instead evaluates the best individual candidate. Opportunity weights and preference labels are constructed from source-training outcomes and are used only during training.

3.3 Stage II: Reference-Guided Joint Calibration

Stage I fixes the base forecast to maintain a stable criterion for candidate opportunity while learning the refiner. Stage II jointly adapts the forecasting module and gate under a shared prediction objective while retaining the learned retrieval proposal. The Stage-I refined prediction provides a common reference for these updates.

The Stage-I parameters $(\eta^R, \psi_c^R, \psi_u^R)$ define the reference proposal $\mathbf{r}_i^R$, gate u_i^R, and refined prediction

$$\widehat{\mathbf{y}}_i^R = (1 - u_i^R)\widehat{\mathbf{y}}_i^{0,R} + u_i^R \mathbf{r}_i^R. \tag{9}$$

We freeze the reference model and scorer ψ_c^R. The aggregation weights p_{ij}^R, proposal $\mathbf{r}_i^R$, and feature vectors $\boldsymbol{\phi}_i^R$ therefore remain fixed for each query throughout calibration. The gate parameters ψ_u remain trainable on these query-specific features. Starting from $(\eta, \psi_u) = (\eta^R, \psi_u^R)$, we jointly update η and ψ_u, yielding

$$\widehat{\mathbf{y}}_i = (1 - u_i)\widehat{\mathbf{y}}_i^0(\eta) + u_i \mathbf{r}_i^R. \tag{10}$$

The objective combines a uniformly averaged forecasting loss with a squared penalty on changes to the reference prediction:

$$\min_{\eta,\psi_u} \mathcal{L}_{\text{cal}} = \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \left[\ell_i + \lambda_{\text{ref}} \frac{\left\|\widehat{\mathbf{y}}_i - \text{sg}(\widehat{\mathbf{y}}_i^R)\right\|_{\Omega_i}^2}{n_i} \right]. \tag{11}$$

Here, $\ell_i = \delta_i(\widehat{\mathbf{y}}_i)$ is point-forecast MSE on the original scale. The norm is restricted to Ω_i, and $\lambda_{\text{ref}} \ge 0$ controls the reference penalty. Thus, we jointly calibrate the forecasting module and gate while regularizing their combined prediction toward the Stage-I refined output.

3.4 Theoretical Analysis

We analyze how candidate opportunity translates into forecasting gains and guides training allocation across queries. We then characterize the additional corrections available through joint calibration under a local linearization on source data.

From Candidate Opportunity to Forecast Improvement. For Stage I with the base forecaster fixed, define the weighted candidate regret

$$R_i^{\mathrm{agg}} = \sum_{k=1}^{J} p_{ik}\delta_i(\mathbf{Z}_{ik}) - \delta_i^{r,\min} \geq 0. \tag{12}$$

For any query with $g_i > 0$, the point-forecast MSE reduction admits the exact decomposition

$$\delta_i^0 - \delta_i(\widehat{\mathbf{y}}_i) = u_i\big[(\delta_i^0 + \epsilon s_i^2)g_i - R_i^{\mathrm{agg}}\big] + \frac{u_i}{n_i}\sum_{k=1}^{J} p_{ik}\|\mathbf{Z}_{ik} - \mathbf{r}_i\|_{\Omega_i}^2 + \frac{u_i(1-u_i)}{n_i}\|\mathbf{r}_i - \widehat{\mathbf{y}}_i^{0,R}\|_{\Omega_i}^2. \tag{13}$$

The two squared-norm terms quantify nonnegative reductions from convex mixing relative to the corresponding weighted component errors. Consequently, $u_i > 0$ and $R_i^{\mathrm{agg}}/(\delta_i^0 + \epsilon s_i^2) < g_i$ are sufficient for a strict improvement over the base forecast. The proof is given in Appendix B.1.

Training Allocation Across Queries. For a source mini-batch $\mathcal{B}$, define the baseline scale weights $q_i = s_i^2/\sum_{k\in\mathcal{B}} s_k^2$, the difficulty-filtered opportunity $a_i = h_i g_i$, and its weighted mean $m = \sum_{i\in\mathcal{B}} q_i a_i$. When $m > 0$, let $q_i^{\mathrm{opp}} = q_i a_i/m$. The forecasting term in Eq. 8 admits the exact decomposition

$$\frac{\sum_{i\in\mathcal{B}} w_i \ell_i}{\sum_{i\in\mathcal{B}} w_i} = (1-\alpha)\sum_{i\in\mathcal{B}} q_i \ell_i + \alpha \sum_{i\in\mathcal{B}} q_i^{\mathrm{opp}} \ell_i, \qquad \alpha = \frac{\beta m}{1+\beta m}. \tag{14}$$

Thus opportunity weighting combines the baseline loss with an additional component supported on difficult queries whose candidates improve on the base forecast. All queries retain positive forecasting-loss weights. When $m = 0$, the forecasting term reduces to the scale-weighted baseline. The derivation is given in Appendix B.2.

Additional Corrections from Joint Calibration. For squared-error calibration on a fixed source set, assume the prediction map is differentiable at the Stage-I reference. Let $\mathbf{e}$ be the stacked difference between the targets and the Stage-I refined predictions, restricted to valid entries and scaled so that $\|\mathbf{e}\|^2$ equals the source-averaged MSE. Let $\mathcal{V}_\eta$ and $\mathcal{V}_u$ be the ranges of the Jacobians of the correspondingly scaled prediction map with respect to the forecasting-module and gate parameters, evaluated at the reference. Linearized joint updates span $\mathcal{V}_{\mathrm{joint}} = \mathcal{V}_\eta + \mathcal{V}_u$. For an output correction $\mathbf{\Delta} \in \mathcal{V}$, the linearized objective is $\|\mathbf{e} - \mathbf{\Delta}\|^2 + \lambda\|\mathbf{\Delta}\|^2$, where $\lambda = \lambda_{\mathrm{ref}} \geq 0$. Writing $\mathbf{P}_\mathcal{V}$ for orthogonal projection onto $\mathcal{V}$, its optimal correction and the resulting forecasting MSE reduction are

$$\mathbf{\Delta}_\mathcal{V}^\star = \frac{\mathbf{P}_\mathcal{V}\mathbf{e}}{1+\lambda}, \qquad G(\mathcal{V}) = \|\mathbf{e}\|^2 - \|\mathbf{e} - \mathbf{\Delta}_\mathcal{V}^\star\|^2 = \frac{1+2\lambda}{(1+\lambda)^2}\|\mathbf{P}_\mathcal{V}\mathbf{e}\|^2. \tag{15}$$

Comparing the optima of the linearized objectives at the same reference and penalty strength, for either $\mathcal{U} \in \{\mathcal{V}_\eta, \mathcal{V}_u\}$,

$$G(\mathcal{V}_{\mathrm{joint}}) - G(\mathcal{U}) = \frac{1+2\lambda}{(1+\lambda)^2}\left\|(\mathbf{P}_{\mathcal{V}_{\mathrm{joint}}} - \mathbf{P}_\mathcal{U})\mathbf{e}\right\|^2 \geq 0. \tag{16}$$

The gain over $\mathcal{U}$ is strict precisely when the reference residual has a nonzero component in $\mathcal{V}_{\mathrm{joint}} \cap \mathcal{U}^\perp$. The derivation and an error analysis for nonlinear updates are provided in Appendices B.3 and B.4.

4 Experiments

4.1 Experimental Settings

Pretraining and Evaluation Datasets For pretraining, we adopt the Chronos pretraining dataset (Ansari et al., 2024) where 50M data points are uniformly sampled following (Ning et al., 2025; Lee et al., 2026) and we adopt the same in-domain retrieval knowledge base following (Ning et al., 2025) which is constructed by a subset of 5M data points. We partition both the pretraining dataset and the retrieval knowledge base into segments using a fixed context window, yielding approximately 26 million pretraining pairs and 2.8 million knowledge base entries.

Table 1: **Zero-shot forecasting results across TSFMs.** Best results are in **boldface**, and the second-best results are underlined.

Methods	ROAR (Ours)		Chronos-bolt (TMLR 2024)		MOMENT (ICML 2024)		TTM (NeurIPS 2024)		Moirai (ICML 2024)		TimesFM (ICML 2024)		Chronos (TMLR 2024)		Time-MoE (ICLR 2025)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.339**	**0.363**	0.362	0.365	0.392	0.411	0.362	0.371	0.369	0.384	0.425	0.383	0.422	0.381	0.362	0.367
ETTh2	**0.235**	**0.295**	0.252	0.299	0.274	0.333	0.253	0.303	0.255	0.305	0.289	0.323	0.266	0.314	0.252	0.322
ETTm1	**0.274**	**0.309**	0.311	0.319	0.351	0.383	0.315	0.325	0.540	0.432	0.332	0.333	0.394	0.370	0.321	0.334
ETTm2	**0.141**	**0.222**	0.149	0.224	0.170	0.258	0.151	0.241	0.196	0.269	0.170	0.255	0.166	0.252	0.157	0.254
Weather	**0.141**	**0.178**	0.153	0.183	0.180	0.238	0.154	0.189	0.171	0.191	—	—	0.190	0.211	0.149	0.184
Electricity	**0.112**	0.205	0.113	**0.200**	0.197	0.303	0.172	0.264	0.183	0.281	—	—	0.146	0.224	0.114	0.203
Exchange	**0.066**	0.176	0.067	0.178	0.098	0.206	**0.066**	0.173	**0.066**	**0.172**	0.070	0.180	0.083	0.188	0.085	0.206
Average	**0.187**	**0.250**	0.201	0.253	0.237	0.304	0.210	0.267	0.254	0.291	—	—	0.238	0.277	0.206	0.267

Table 2: **Zero-shot forecasting results across RAG-based TSF methods.** Best results are in **boldface**, and the second-best results are underlined.

Methods	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Electricity		Exchange		Average	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$\text{PFRP}_{\text{Chronos-bolt}}$	0.361	0.365	0.249	0.299	0.311	0.319	0.149	0.224	0.153	0.184	0.114	0.201	0.068	0.179	0.201	0.253
$\text{RATL}_{\text{Chronos-bolt}}$	0.362	0.365	0.251	0.299	0.310	0.318	0.148	0.224	0.152	0.182	0.113	0.200	0.067	0.178	0.201	0.252
$\text{RAF}_{\text{Chronos-bolt}}$	0.366	0.371	0.252	0.305	0.306	0.329	0.148	0.228	0.178	0.206	0.118	0.211	**0.063**	**0.172**	0.204	0.260
TS-RAG	0.359	0.364	0.248	0.299	0.300	0.313	0.147	0.222	0.150	0.181	0.113	**0.200**	0.067	0.176	0.198	0.251
Cross-RAG	0.344	0.371	0.244	0.302	0.290	0.319	0.146	0.226	0.145	0.180	0.113	0.205	0.070	0.183	0.193	0.255
ROAR (Ours)	**0.339**	**0.363**	**0.235**	**0.295**	**0.274**	**0.309**	**0.141**	**0.222**	**0.141**	**0.178**	**0.112**	0.205	0.066	0.176	**0.187**	**0.250**

We adopt 7 benchmark time series forecasting datasets for zero-shot forecasting evaluation, including ETTh1, ETTh2, ETTm1, ETTm2 (Zhou et al., 2021), Weather, Electricity (Wu et al., 2023), and Exchange Rate (Lai et al., 2018). We conduct zero-shot evaluation on the test set of each dataset. Details of each dataset are presented in Table 9 in Appendix C.

Baselines and Implementation. We compare ROAR with 7 TSFMs: Chronos (Ansari et al., 2024), Chronos-Bolt (Ansari et al., 2024), MOMENT (Goswami et al., 2024), TTM (Ekambaram et al., 2024), Moirai (Woo et al., 2024), TimesFM (Das et al., 2024), and Time-MoE (Shi et al., 2025). We also compare with 5 RAG-based TSF methods: RAF (Tire et al., 2026), TS-RAG (Ning et al., 2025), Cross-RAG (Lee et al., 2026), PFRP (Du et al., 2026), and RATL (He et al., 2026). PFRP and RATL are adapted to the zero-shot setting. All RAG-based methods, including ROAR, use the frozen Chronos-Bolt backbone and share the same training and inference protocols. Following Ning et al. (2025), the input length and forecasting horizon are set as 512 and 64, and the pre-training corpus and retrieval settings follow TS-RAG with $K = 10$ retrieved examples. After training the forecasting module, ROAR is trained for 2,500 steps in Stage I and 2,000 steps in Stage II. We set $\beta = 0.5$, $\lambda_{\text{gate}} = 1$, and $\lambda_{\text{ref}} = 15$. All methods are evaluated with MSE and MAE. We report the averaged scores of 5 independent runs for our ROAR and all reproduced methods. Further implementation details and hyper-parameters are provided in Appendix C.

4.2 Zero-shot Forecasting Performance

We compare ROAR with 7 TSFMs and 5 RAG-based TSF methods. Baseline results come from reproductions using the official source code and from the results reported in (Lee et al., 2026).

Comparison with TSFMs. As shown in Table 1, ROAR achieves the best performance in most settings among all TSFMs. Compared with its Chronos-Bolt backbone, the second-best model on average, ROAR reduces the average MSE from 0.201 to 0.187 (7.0%). The largest reduction is on ETTm1, from 0.311 to 0.274 (11.9%). On Electricity and Exchange, ROAR is slightly behind the best MAE (0.205 vs. 0.200 and 0.176 vs. 0.172) while keeping the best MSE. This is consistent with Stage II being calibrated with point-forecast MSE. These results show that retrieved historical continuations add useful information that pretrained backbones alone do not provide.

Figure 3: **ROAR as a plug-in across retrieval-augmented methods.** Relative MSE reduction of each method equipped with ROAR (higher is better).

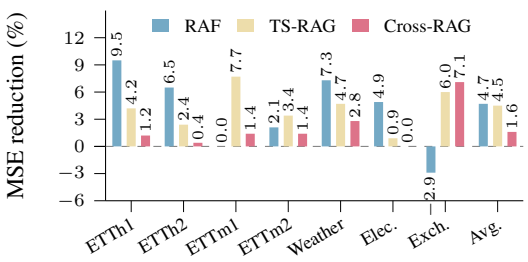


Table 3: **ROAR across TSFM backbones.** $\Delta\%$: relative change of MSE / MAE over the backbone; deeper shading means larger MSE reduction.

Data	Moirai2			TimesFM2.5		
	Base	+ROAR	$\Delta\%$	Base	+ROAR	$\Delta\%$
ETTh1	.446/.451	**.412/.428**	−7.5/−5.2	.404/.421	**.390/.411**	−3.4/−2.3
ETTh2	.274/.342	**.258/.331**	−6.1/−3.1	.268/.336	**.252/.326**	−5.9/−3.0
ETTm1	.379/.398	**.352/.380**	−7.3/−4.5	.348/.376	**.332/.367**	−4.5/−2.5
ETTm2	.189/.275	**.177/.266**	−6.5/−3.2	.187/.272	**.175/.264**	−6.5/−3.0
Weather	.157/.205	**.154/.202**	−1.8/−1.4	.156/.202	**.153/.201**	−1.8/−0.6
Elec.	.244/.351	**.194/.304**	−20.5/−13.6	.184/.293	**.168/.275**	−9.0/−6.3
Exch.	.092/.215	**.085/.207**	−8.6/−3.9	.094/.216	**.088/.209**	−5.8/−3.2
Avg.	.255/.320	**.233/.303**	−8.5/−5.4	.234/.303	**.223/.293**	−5.0/−3.0

Comparison with RAG-based TSF methods. We present the comparison with RAF, TS-RAG, Cross-RAG, PFRP, and RATL in Table 2. PFRP and RATL were originally trained per dataset. We adapt them to the zero-shot setting by training once on the same pre-training corpus over a frozen Chronos-Bolt backbone. This protocol is same for all methods. We can find that, ROAR achieves the best MSE on six of the seven datasets and the best average MSE and MAE. Compared with Cross-RAG, the second-best method in average MSE, ROAR reduces the average MSE from 0.193 to 0.187 and the average MAE from 0.255 to 0.250. The largest reduction is on ETTm1 (0.290 to 0.274). Existing methods learn how to use candidates within each query, but weight all training queries equally regardless of whether their candidates can improve the base forecast. ROAR instead emphasizes training queries where retrieval offers an opportunity and learns how strongly to correct toward it, making its use of retrieved information more effective.

4.3 Effectiveness Examination of ROAR

Ablation Study We examine each training stage of ROAR by removing them one at a time (Table 4). "w/o Stage II" uses the Stage-I prediction $\widehat{\mathbf{y}}_i^R$ without joint calibration, and "w/o Stage I & II" reduces to the base forecaster $\widehat{\mathbf{y}}_i^{0,R}$; Stage I is not removed alone since Stage II builds on it. All variants share the same frozen backbone, retrieval, and zero-shot protocol. Removing either stage degrades performance. Stage I contributes most, and removing it raises the average MSE from 0.1884 to 0.1950, showing that opportunity-aware refinement turns retrieved candidates into useful corrections. Stage II further lowers MSE on every dataset, with smaller and mixed effects on MAE since it is trained with MSE. Overall, Stage I learns where and how much to trust retrieved candidates, and Stage II calibrates the base forecast around this refinement.

Table 4: **Ablation of ROAR's two stages** (Chronos-Bolt, zero-shot). Subscripts: absolute change caused by removing one more stage, i.e., w/o Stage II vs. ROAR, and w/o Stage I & II vs. w/o Stage II.

Data	ROAR		w/o Stage II		w/o Stage I & II	
	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.3385**	**0.3626**	$0.3432_{+0.0047}$	$0.3645_{+0.0019}$	$0.3527_{+0.0095}$	$0.3645_{\pm0.0000}$
ETTh2	**0.2351**	0.2953	$0.2361_{+0.0010}$	$\mathbf{0.2952}_{-0.0001}$	$0.2413_{+0.0052}$	$0.2959_{+0.0007}$
ETTm1	**0.2735**	**0.3087**	$0.2778_{+0.0043}$	$0.3092_{+0.0005}$	$0.2949_{+0.0171}$	$0.3132_{+0.0040}$
ETTm2	**0.1408**	**0.2221**	$0.1418_{+0.0010}$	$0.2222_{+0.0001}$	$0.1459_{+0.0041}$	$0.2230_{+0.0008}$
Weather	**0.1405**	0.1779	$0.1418_{+0.0013}$	$\mathbf{0.1776}_{-0.0003}$	$0.1476_{+0.0058}$	$0.1809_{+0.0033}$
Elec.	**0.1119**	0.2047	$0.1121_{+0.0002}$	$0.2034_{-0.0013}$	$0.1133_{+0.0012}$	$\mathbf{0.2028}_{-0.0006}$
Exch.	**0.0660**	**0.1761**	$0.0662_{+0.0002}$	$0.1765_{+0.0004}$	$0.0692_{+0.0030}$	$0.1803_{+0.0038}$
Avg.	**0.1866**	**0.2496**	$0.1884_{+0.0018}$	$0.2498_{+0.0002}$	$0.1950_{+0.0066}$	$0.2515_{+0.0017}$

Plugging ROAR into RAG-based TSF Methods We test whether ROAR works as a plug-in refiner for existing retrieval-augmented forecasters. We apply it to three methods: RAF, TS-RAG, and Cross-RAG. Each method keeps its own backbone, retrieval pipeline, and hyper-parameters, and its forecast serves as the base forecast $\widehat{\mathbf{y}}_i^0$ that ROAR refines. Only the ROAR modules are added. Fig. 3 reports the relative MSE reduction. ROAR lowers the average MSE by 4.7%, 4.5%, and 1.6%, respectively. It improves 18 of the 21 Base–dataset pairs and degrades only RAF on Exchange. Cross-RAG gains the least. A likely reason is that its cross-attention fusion already uses the retrieved series more effectively, so less improvement is left for refinement. Overall, ROAR is compatible with different retrieval-augmented designs and consistently improves.

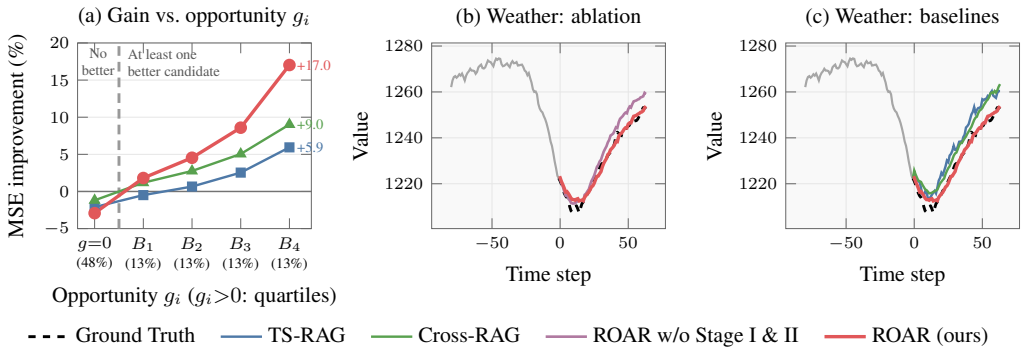


Figure 4: (a) MSE improvement over Chronos-Bolt over the best-candidate opportunity g_i. (b) Forecasting visualization between ROAR and the ablative version "ROAR w/o Stage-I & II". (c) Forecasting visualization between ROAR and RAG-based TSF methods TS-RAG and Cross-RAG.

Generalization across Backbones To examine whether ROAR generalizes beyond different backbones, we replace the backbone with Moirai2.0 Liu et al. (2025a) and TimesFM2.5 Das et al. (2024), and keep the same retrieval procedure and zero-shot protocol. Table 3 shows that ROAR improves both backbones on all seven datasets and on both metrics. The average MSE falls by 8.5% for Moirai2.0 and 5.0% for TimesFM2.5. The largest gain is on Electricity (20.5% with Moirai2.0), whose recurring daily and weekly patterns make retrieved continuations especially informative. Gains are smaller on Weather (1.8%), where historical segments offer less transferable information. Overall, ROAR is backbone-agnostic and complements the representations learned by different TSFMs.

4.4 Visualizations

Retrieval Opportunity Gains Visualization We examine whether ROAR's gains concentrate on queries with beneficial candidates. We compute g_i (Eq. 7) with Chronos-Bolt as a common base over the top-10 level-aligned candidates shared by TS-RAG and ROAR, and bin queries into $g_i = 0$ and within-dataset quartiles B_1–B_4 of $g_i > 0$. To avoid selecting bins on the evaluated errors, g_i and the improvement are computed on opposite halves of the horizon, averaged over both directions; $g_i > 0$ thus indicates a better candidate on the binning half. As shown in Fig. 4(a), gains of all methods grow with g_i, and ROAR grows fastest: it reaches $+17.0\%$ in B_4 versus $+5.9\%$ (TS-RAG) and $+9.0\%$ (Cross-RAG), and leads in every $g_i > 0$ bin. When no candidate is better, all methods lose slightly (-1.2% to -2.9%). Thus, ROAR converts retrieval opportunity into forecast improvement more effectively than existing retrieval-augmented forecasters.

Forecasting Visualizations We present the forecasting visualizations between ROAR and the ablative version and RAG-based TSF methods in Fig. 4(b,c). We can find that our ROAR consistently outperforms both the ablative version and the baseline methods in terms of trend fidelity and continuity with the history, directly indicating the effectiveness of our proposal. More visualizations are provided in Appendix C.4.

5 Conclusion

We presented ROAR, a framework that learns to refine time series forecasts using retrieved historical continuations. ROAR allocates additional training emphasis to difficult source queries with beneficial candidates, then uses reference-guided joint calibration to coordinate the base forecast and learned correction. Our analyses characterize realized refinement gains, the training allocation induced by opportunity weighting, and conditions for additional gains from joint calibration under a local linearization. Experiments on seven benchmarks demonstrate improved average forecasting accuracy across three backbone families and multiple retrieval-augmented hosts without target-specific parameter updates. Stage ablations further show that refinement and joint calibration contribute complementary MSE improvements.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and polishing of the manuscript. We have reviewed all AI-assisted revisions to ensure that they accurately reflect our intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies retrieval-augmented time series forecasting using publicly available benchmark datasets. It does not involve human-subject experiments or the collection of personal data. We do not identify any specific ethical concerns arising from this study.

REPRODUCIBILITY STATEMENT

We provide anonymous source code to support reproduction of our experiments in `https://anonymous.4open.science/r/ROAR-40D2`.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

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

Dazhao Du, Tao Han, and Song Guo. Predicting the future by retrieving the past. In *AAAI Conference on Artificial Intelligence*, volume 40, pp. 20896–20904, 2026.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H 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 *Neural Information Processing Systems*, volume 37, pp. 74147–74181, 2024.

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*, pp. 16115–16152, 2024.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *International Conference on Machine Learning*, pp. 21774–21797, 2025.

Yuchen He, Yueyang Cang, Zhiyuan Ning, Ningyu Wang, and Li Shi. RATL: Learning from retrieved residuals for robust multivariate time-series forecasting. *arXiv preprint arXiv:2609.03937*, 2026.

Bo Jin, Haoyu Yang, Leilei Sun, Chuanren Liu, Yue Qu, and Jianing Tong. A treatment engine by predicting next-period prescriptions. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1608–1616, 2018.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artificial Intelligence Review*, 58(7):216, 2025.

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 & Development in Information Retrieval*, pp. 95–104, 2018.

Seunghan Lee, Jaehoon Lee, Jun Seo, Sungdong Yoo, Minjae Kim, Tae Yoon Lim, Dongwan Kang, Hwanil Choi, SoonYoung Lee, and Wonbin Ahn. Not all retrievals are useful: Cross-attention for input-aware RAG in time series forecasting. *arXiv preprint arXiv:2603.14709*, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Neural Information Processing Systems*, volume 33, pp. 9459–9474, 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.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, pp. 2980–2988, 2017.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025a.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, et al. Moirai-MoE: Empowering time series foundation models with sparse mixture of experts. In *International Conference on Machine Learning*, pp. 38940–38962, 2025b.

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*, pp. 39295–39317, 2025c.

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. *arXiv preprint arXiv:2603.04791*, 2026.

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 *Neural Information Processing Systems*, 2025.

Xiangfei Qiu, Xiuwen Li, Ruiyang Pang, Zhicheng Pan, Xingjian Wu, Liu Yang, Jilin Hu, Yang Shu, Xuesong Lu, Chengcheng Yang, et al. EasyTime: Time series forecasting made easy. In *International Conference on Data Engineering*, pp. 4564–4567, 2025a.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *Neural Information Processing Systems*, 2025b.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343, 2018.

Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009.

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.

Yanru Sun, Zongxia Xie, Yanhong Chen, and Qinghua Hu. Accurate solar wind speed prediction with multimodality information. *Space: Science & Technology*, 2022.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, 2026.

Shivani Tomar, Seshu Tirupathi, Radu Marinescu, Elizabeth M Daly, and Ivana Dusparic. AT4TS: Autotune for time series foundation models. *Transactions on Machine Learning Research*, 2025.

Maria Tzelepi, Charalampos Symeonidis, Paraskevi Nousi, Efstratios Kakaletsis, Theodoros Manousis, Pavlos Tosidis, Nikos Nikolaidis, and Anastasios Tefas. Deep learning for energy time-series analysis and forecasting. *arXiv preprint arXiv:2306.09129*, 2023.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024.

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*, 2024.

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.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. Retrieval-augmented generation for natural language processing: A survey. *Artificial Intelligence Review*, 2026.

Huanyu Zhang, Chang Xu, Yi-Fan 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*, 2025.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. Retrieval-augmented generation for AI-generated content: A survey. *Data Science and Engineering*, pp. 1–29, 2026.

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*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Neural Information Processing Systems*, volume 36, pp. 43322–43355, 2023.

APPENDICES: ROAR: RETRIEVAL OPPORTUNITY-AWARE REFINEMENT FOR ZERO-SHOT TIME SERIES FORECASTING

A PRELIMINARY STUDY: ARE RETRIEVED NEIGHBORS FULLY EXPLOITED?

This section examines whether existing retrieval-augmented forecasters fully exploit the information carried by retrieved neighbors. We study three questions: (**Q1**) Whether retrieval contributes positively and whether the contribution stems from the retrieved content; (**Q2**) How much of the improvement offered by the retrieved candidates is realized; and (**Q3**) Whether retrieval similarity indicates which candidates are useful. The overall finding is that retrieval is beneficial, but existing methods realize only a small part of its potential: the value of retrieved neighbors is highly uneven across samples, and similarity cannot reveal it.

A.1 EXPERIMENTAL SETUP

Models and data. We conduct the examination on two state-of-the-art retrieval-augmented zero-shot forecasting methods, TS-RAG (Ning et al., 2025) and Cross-RAG (Lee et al., 2026), both reproduced with their official configurations. We use the benchmarks of the main experiments and follow the official evaluation procedure of both methods. Following their official code, each variable retrieves only from its own training split of the target dataset. Unless stated otherwise, results are equal-weight averages over ETTh1, ETTh2, ETTm1, ETTm2, Weather, Exchange, and Electricity.

Candidate forecasts and sample groups. For a test sample i with retrieved neighbors $\{(\mathbf{x}_{ij}^r, \mathbf{y}_{ij}^r)\}_{j=1}^K$, we take the level-aligned continuation $\mathbf{Z}_{ij}^{\text{lev}}$ of Eq. (2) as the j-th candidate forecast. Let ℓ_i^0 denote the MSE of the base forecast and $\ell_i^{r,\min} = \min_j \ell_i(\mathbf{Z}_{ij}^{\text{lev}})$ that of the best candidate. We compute the opportunity score $g_i = [\ell_i^0 - \ell_i^{r,\min}]_+ / \ell_i^0$, which has the form of Eq. (8), and split the samples into $\mathcal{G}^+ = \{i : g_i > 0\}$, where at least one candidate outperforms the base forecast, and $\mathcal{G}^- = \{i : g_i = 0\}$, where none does.

Metrics. MSE and MAE follow the official evaluation. For a set of samples, the improvement of a forecast over the base forecast is $\Delta = (e_0 - e)/e_0$, where e_0 and e are the mean errors of the base and the compared forecast on that set. Positive values indicate lower error. The groups $\mathcal{G}^{\pm}$ and the best candidate are always defined by MSE; MAE is reported on the same samples and candidates.

A.2 DOES RETRIEVAL CONTRIBUTE POSITIVELY?

Method. We use two controls. *Candidate quality:* for each test sample, we draw K training samples uniformly from its retrieval library (three draws) and compare the share of samples that have a candidate better than the base forecast with that obtained from the retrieved top-K neighbors. *Neighbor shuffling:* we replace the neighbor set of each test sample with that of a randomly chosen test sample of the same variable, keeping the model and all other inputs fixed, and measure the overall improvement of each method.

Results. Table 5 reports both controls. TS-RAG's retrieved candidates outperform random ones on all seven datasets. Cross-RAG's candidates outperform random ones on the ETT datasets and Electricity but fall behind on Weather and Exchange, so on average they are on par with random draws. After shuffling, the MSE improvement of both methods turns negative and MAE degrades further. With the original neighbors, Cross-RAG improves MSE but leaves MAE essentially unchanged.

Conclusion. Retrieval contributes positively, and the contribution originates from the retrieved content rather than from the model architecture.

A.3 IS THE RETRIEVAL GAIN FULLY REALIZED?

Method. On $\mathcal{G}^+$, we measure the improvement offered by the best candidate and define the *realization rate* as the error reduction achieved by the retrieval-augmented method divided by that

Table 5: Contribution of retrieval. Top: share of samples with a candidate better than the base forecast. Bottom: overall improvement Δ over the base forecast with the original and the shuffled neighbor sets (seed 2021). Positive values indicate lower error.

	TS-RAG		Cross-RAG	
	MSE	MAE	MSE	MAE
Share of samples with a better candidate				
Retrieved top-K	46.3%		35.3%	
Random draws from the library	30.6%		35.1%	
Overall improvement Δ				
Original neighbors	$+1.4\%$	$+0.7\%$	$+3.3\%$	-0.4%
Shuffled neighbors	-0.7%	-1.2%	-12.9%	-14.0%

offered by the best candidate. Within $\mathcal{G}^+$ and $\mathcal{G}^-$, we compare the retrieval-augmented and base forecasts sample by sample, and we measure the share of the total available improvement contributed by the top 10% of samples. Because $\mathcal{G}^\pm$ are defined relative to the base forecast, grouping and evaluating on the same horizon steps share the same noise and induce a regression-to-the-mean effect. We therefore also define the groups on one half of the horizon (steps 1–32 or 33–64) and evaluate on the other half, averaging both directions.

Results. Table 6 summarizes the results (three seeds, averaged). The best candidate reduces MSE by about half and MAE by about a third on $\mathcal{G}^+$, yet TS-RAG and Cross-RAG realize only 5.1% and 20.0% of the MSE reduction (5.9% and 15.8% for MAE). Even when a better candidate exists, retrieval worsens 38% and 27% of the forecasts in MSE. The available improvement is highly concentrated: the top 10% of samples account for 80% and 85% of it. Table 7 shows that, across halves, the degradation on $\mathcal{G}^-$ shrinks by about two thirds to three quarters in MSE and by about half to two thirds in MAE. A large part of this degradation thus reflects regression to the mean, and we report it only as a descriptive result.

Table 6: Realization of the available improvement (three seeds, averaged). Δ denotes the improvement over the base forecast; positive values indicate lower error.

	TS-RAG		Cross-RAG	
	MSE	MAE	MSE	MAE
Share of $\mathcal{G}^+$	46.3%		35.3%	
Δ of the best candidate on $\mathcal{G}^+$	$+51.2\%$	$+31.1\%$	$+46.1\%$	$+27.2\%$
Realization rate	5.1%	5.9%	20.0%	15.8%
Worsened samples in $\mathcal{G}^+$	38%	40%	27%	34%
Worsened samples in $\mathcal{G}^-$	55%	56%	64%	66%
Share of available gain from the top 10% of samples	80%	66%	85%	75%

Table 7: Split-horizon control on $\mathcal{G}^-$. Improvement Δ of the retrieval-augmented forecast when the groups are defined and evaluated on the same half of the horizon or on different halves (three seeds, averaged).

	TS-RAG		Cross-RAG	
Grouping and evaluation	MSE	MAE	MSE	MAE
Same half	-7.5%	-4.5%	-13.8%	-9.6%
Different halves	-2.4%	-1.6%	-3.5%	-5.1%

Conclusion. Existing methods realize only a small part of the improvement available from retrieval, and this improvement is concentrated in a small share of samples.

A.4 Can Retrieval Similarity Identify Useful Candidates?

Method. We examine how often the most similar candidate is the best one, compared with random choice, and the Spearman correlation between the retrieval ranking and the candidate MSE ranking. We further compute the AUROC of retrieval similarity and of other observable quantities (candidate dispersion, deviation of the candidate mean from the base forecast, context volatility, and the base MSE of the preceding sample) for identifying $\mathcal{G}^+$. Finally, we measure the improvement obtained by replacing the base forecast with the most similar candidate.

Results. As shown in Table 8, the most similar candidate is the best one no more often than a random choice, and the retrieval ranking is nearly uncorrelated with candidate error. None of the observable quantities separates $\mathcal{G}^+$ from $\mathcal{G}^-$ (AUROC close to 0.5), and using the most similar candidate directly degrades both metrics severely.

Table 8: Relation between retrieval similarity and candidate usefulness. Values in parentheses are the expected rates under random choice.

	TS-RAG		Cross-RAG	
	MSE	MAE	MSE	MAE
Most similar candidate is the best	10.5% (10.0%)		6.4% (6.7%)	
Spearman correlation (rank vs. MSE)	0.04		0.04	
AUROC of observables for $\mathcal{G}^+$	0.45–0.53		0.46–0.53	
Δ of the most similar candidate	-157%	-64.2%	-392%	-78.6%

Conclusion. Retrieval similarity indicates neither which candidate is useful nor which samples admit an improvement, which limits how existing methods exploit the candidate set.

A.5 Summary

Retrieval contributes positively to forecasting, and the contribution stems from the retrieved content (Section A.2). However, existing methods realize only a small fraction of the improvement offered by the retrieved candidates, which is concentrated in a small share of samples (Section A.3). Because retrieval similarity cannot reveal which candidates are useful (Section A.4), a refiner should learn from the improvement each candidate offers over the base forecast, which motivates ROAR. Two limitations remain. Cross-RAG's retrieved candidates are on average no better than random draws, so its positive contribution rests mainly on the shuffling control. Moreover, no observable feature identifies samples with an opportunity at inference time; whether a learned gate can adapt to the input is examined in the main experiments.

B Proofs and Additional Analysis

B.1 From Candidate Opportunity to Forecast Improvement

Fix a Stage-I query, its reference base forecast, and the candidate weights and gate produced by the refiner. The weights satisfy $p_{ik} \geq 0$ and $\sum_k p_{ik} = 1$, and the gate satisfies $u_i \in [0, 1]$. Write $\mathbf{r}_i = \sum_k p_{ik}\mathbf{Z}_{ik}$ and $\widehat{\mathbf{y}}_i = (1 - u_i)\widehat{\mathbf{y}}_i^{0,R} + u_i\mathbf{r}_i$. Throughout this subsection, δ_i is the squared point-forecast error summed over Ω_i and divided by n_i.

Expanding the squared errors around $\mathbf{r}_i$ and using $\sum_k p_{ik}(\mathbf{Z}_{ik} - \mathbf{r}_i) = 0$ gives

$$\begin{aligned}\delta_i(\mathbf{r}_i) &= \sum_{k=1}^{J} p_{ik}\delta_i(\mathbf{Z}_{ik}) - \frac{1}{n_i}\sum_{k=1}^{J} p_{ik}\|\mathbf{Z}_{ik} - \mathbf{r}_i\|_{\Omega_i}^2 \\ &= \delta_i^{r,\min} + R_i^{\text{agg}} - \frac{1}{n_i}\sum_{k=1}^{J} p_{ik}\|\mathbf{Z}_{ik} - \mathbf{r}_i\|_{\Omega_i}^2.\end{aligned} \tag{17}$$

Applying the same squared-error identity to the mixture of the base forecast and proposal yields

$$\delta_i(\widehat{\mathbf{y}}_i) = (1-u_i)\delta_i^0 + u_i\delta_i(\mathbf{r}_i) - \frac{u_i(1-u_i)}{n_i}\|\mathbf{r}_i - \widehat{\mathbf{y}}_i^{0,R}\|_{\Omega_i}^2. \tag{18}$$

Combining the two identities gives, for every query,

$$\begin{aligned}\delta_i^0 - \delta_i(\widehat{\mathbf{y}}_i) = u_i\big[\delta_i^0 - \delta_i^{r,\min} - R_i^{\mathrm{agg}}\big] \\ + \frac{u_i}{n_i}\sum_{k=1}^{J} p_{ik}\|\mathbf{Z}_{ik} - \mathbf{r}_i\|_{\Omega_i}^2 \\ + \frac{u_i(1-u_i)}{n_i}\|\mathbf{r}_i - \widehat{\mathbf{y}}_i^{0,R}\|_{\Omega_i}^2.\end{aligned} \tag{19}$$

When $g_i > 0$, its definition implies $\delta_i^0 - \delta_i^{r,\min} = (\delta_i^0 + \epsilon s_i^2)g_i$. Substitution proves Eq. 13. Dropping the two nonnegative squared-norm terms gives

$$\frac{\delta_i^0 - \delta_i(\widehat{\mathbf{y}}_i)}{\delta_i^0 + \epsilon s_i^2} \geq u_i\left(g_i - \frac{R_i^{\mathrm{agg}}}{\delta_i^0 + \epsilon s_i^2}\right). \tag{20}$$

The denominator is positive because $s_i > 0$ and $\epsilon > 0$. The stated sufficient condition makes the right-hand side strictly positive. The condition is not necessary, since the mixing terms can provide additional gains.

B.2 Training Allocation Across Queries

Fix a nonempty source mini-batch $\mathcal{B}$. Recall that $s_i > 0$, $\beta \geq 0$, and $a_i = h_i g_i \geq 0$. Let $S_{\mathcal{B}} = \sum_{i\in\mathcal{B}} s_i^2$. Since $w_i = s_i^2(1+\beta a_i)$, we have $\sum_{i\in\mathcal{B}} w_i = S_{\mathcal{B}}(1+\beta m)$. The normalized forecasting-loss weights therefore satisfy

$$\widetilde{q}_i = \frac{w_i}{\sum_{k\in\mathcal{B}} w_k} = \frac{q_i(1+\beta a_i)}{1+\beta m}. \tag{21}$$

When $m > 0$, $q_i^{\mathrm{opp}} = q_i a_i / m$ defines a probability distribution. Substituting $\alpha = \beta m/(1+\beta m)$ gives $\widetilde{q}_i = (1-\alpha)q_i + \alpha q_i^{\mathrm{opp}}$. Multiplying by ℓ_i and summing over i proves Eq. 14. If $m = 0$, positivity of q_i and nonnegativity of a_i imply $a_i = 0$ for every query, so $\widetilde{q}_i = q_i$. For all $m \geq 0$, $\widetilde{q}_i > 0$. When $m > 0$, $q_i^{\mathrm{opp}} > 0$ exactly for queries with $h_i g_i > 0$.

The redistribution does not decrease the weighted mean opportunity:

$$\begin{aligned}\sum_{i\in\mathcal{B}} \widetilde{q}_i a_i - m &= \frac{\beta\left(\sum_{i\in\mathcal{B}} q_i a_i^2 - m^2\right)}{1+\beta m} \\ &= \frac{\beta\ \mathrm{Var}_q(a)}{1+\beta m} \geq 0.\end{aligned} \tag{22}$$

The increase is strict when $\beta > 0$ and the opportunity values are not constant across the batch. These identities hold for any per-query forecasting loss ℓ_i. They apply to the forecasting term in Eq. 8, while the auxiliary gate loss retains its uniform averaging.

B.3 Joint Calibration in the Linearized Prediction Space

Coordinates and Correction Spaces. Fix a nonempty source calibration set $\mathcal{S}$ and write $\boldsymbol{\vartheta} = (\eta, \psi_u)$, $\boldsymbol{\vartheta}^R = (\eta^R, \psi_u^R)$, and $\lambda = \lambda_{\mathrm{ref}} \geq 0$. Define the scaled prediction map and target vector by

$$\mathbf{f}(\boldsymbol{\vartheta}) = \left[\frac{\widehat{y}_{i,t}(\boldsymbol{\vartheta})}{\sqrt{|\mathcal{S}|n_i}}\right]_{i\in\mathcal{S},\, t\in\Omega_i}, \qquad \mathbf{y}_{\mathcal{S}} = \left[\frac{y_{i,t}}{\sqrt{|\mathcal{S}|n_i}}\right]_{i\in\mathcal{S},\, t\in\Omega_i}. \tag{23}$$

Empty valid-entry sets contribute no coordinates and zero loss. With $\mathbf{e} = \mathbf{y}_{\mathcal{S}} - \mathbf{f}(\boldsymbol{\vartheta}^R)$, the calibration objective on $\mathcal{S}$ is exactly

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\vartheta}) = \|\mathbf{y}_{\mathcal{S}} - \mathbf{f}(\boldsymbol{\vartheta})\|^2 + \lambda\|\mathbf{f}(\boldsymbol{\vartheta}) - \mathbf{f}(\boldsymbol{\vartheta}^R)\|^2. \tag{24}$$

Assume that $\mathbf{f}$ is differentiable at the reference. Let $\mathbf{J}_\eta$ and $\mathbf{J}_u$ denote its Jacobian blocks with respect to η and ψ_u at $\boldsymbol{\vartheta}^R$. Because the proposal and reference features remain fixed, the per-query blocks before masking and scaling are

$$\begin{aligned}\mathbf{J}_{\eta,i} &= (1-u_i^R)\left.\frac{\partial \widehat{\mathbf{y}}_i^0(\eta)}{\partial \eta}\right|_{\eta^R},\\ \mathbf{J}_{u,i} &= (\mathbf{r}_i^R - \widehat{\mathbf{y}}_i^{0,R})\left(\left.\nabla_{\psi_u} u_i\right|_{\psi_u^R}\right)^\top.\end{aligned} \tag{25}$$

Restricting these blocks to Ω_i, dividing by $\sqrt{|\mathcal{S}|n_i}$, and stacking over queries yields $\mathbf{J}_\eta$ and $\mathbf{J}_u$. Thus forecasting-module updates change the base prediction with sensitivity scaled by $1-u_i^R$, while gate updates act along the fixed proposal direction for each query.

Writing $\mathbf{J} = [\mathbf{J}_\eta\ \mathbf{J}_u]$, the linearized output correction for a parameter increment $\mathbf{d}$ is $\boldsymbol{\Delta} = \mathbf{J}\mathbf{d}$. The available correction spaces are

$$\mathcal{V}_\eta = \mathrm{range}(\mathbf{J}_\eta), \qquad \mathcal{V}_u = \mathrm{range}(\mathbf{J}_u), \qquad \mathcal{V}_{\mathrm{joint}} = \mathrm{range}(\mathbf{J}) = \mathcal{V}_\eta + \mathcal{V}_u. \tag{26}$$

Optimal Correction and Forecasting Gain. For any of these spaces $\mathcal{V}$, define

$$\mathcal{Q}_\mathcal{V}(\boldsymbol{\Delta}) = \|\mathbf{e}-\boldsymbol{\Delta}\|^2 + \lambda\|\boldsymbol{\Delta}\|^2, \qquad \boldsymbol{\Delta}\in\mathcal{V}. \tag{27}$$

Since $\mathbf{e}-\mathbf{P}_\mathcal{V}\mathbf{e}$ is orthogonal to $\mathcal{V}$, completing the square gives

$$\mathcal{Q}_\mathcal{V}(\boldsymbol{\Delta}) = \|\mathbf{e}\|^2 - \frac{\|\mathbf{P}_\mathcal{V}\mathbf{e}\|^2}{1+\lambda} + (1+\lambda)\left\|\boldsymbol{\Delta} - \frac{\mathbf{P}_\mathcal{V}\mathbf{e}}{1+\lambda}\right\|^2. \tag{28}$$

The unique optimal output correction is therefore $\boldsymbol{\Delta}_\mathcal{V}^\star = \mathbf{P}_\mathcal{V}\mathbf{e}/(1+\lambda)$. Its forecasting MSE reduction is

$$\begin{aligned}G(\mathcal{V}) &= 2\langle \mathbf{e}, \boldsymbol{\Delta}_\mathcal{V}^\star\rangle - \|\boldsymbol{\Delta}_\mathcal{V}^\star\|^2\\ &= c_\lambda\|\mathbf{P}_\mathcal{V}\mathbf{e}\|^2, \qquad c_\lambda = \frac{1+2\lambda}{(1+\lambda)^2}.\end{aligned} \tag{29}$$

This proves Eq. 15.

For $\mathcal{U}\subseteq\mathcal{V}_{\mathrm{joint}}$, the operator $\mathbf{P}_{\mathcal{V}_{\mathrm{joint}}} - \mathbf{P}_\mathcal{U}$ is the orthogonal projector onto $\mathcal{V}_{\mathrm{joint}}\cap\mathcal{U}^\perp$. Orthogonality gives

$$\|\mathbf{P}_{\mathcal{V}_{\mathrm{joint}}}\mathbf{e}\|^2 = \|\mathbf{P}_\mathcal{U}\mathbf{e}\|^2 + \|(\mathbf{P}_{\mathcal{V}_{\mathrm{joint}}} - \mathbf{P}_\mathcal{U})\mathbf{e}\|^2. \tag{30}$$

Multiplying by $c_\lambda > 0$ proves Eq. 16 for either $\mathcal{U}\in\{\mathcal{V}_\eta, \mathcal{V}_u\}$. The gain is strict exactly when the residual projects nontrivially onto the added orthogonal directions.

B.4 Optimization Error and Nonlinear Updates

To quantify departures from the linearized optima, assume that $\mathbf{f}$ is continuously differentiable in a convex neighborhood of $\boldsymbol{\vartheta}^R$ and its Jacobian is L_J-Lipschitz there in operator norm. For a correction space $\mathcal{V}$, consider a parameter increment $\mathbf{d}_\mathcal{V}$ whose segment from the reference lies in this neighborhood. For a single-component update, set the other parameter block to zero. The actual output correction satisfies

$$\begin{aligned}\boldsymbol{\Delta}_{\mathrm{act}}(\mathbf{d}_\mathcal{V}) &= \mathbf{f}(\boldsymbol{\vartheta}^R + \mathbf{d}_\mathcal{V}) - \mathbf{f}(\boldsymbol{\vartheta}^R)\\ &= \mathbf{J}\mathbf{d}_\mathcal{V} + \mathbf{T}(\mathbf{d}_\mathcal{V}),\\ \|\mathbf{T}(\mathbf{d}_\mathcal{V})\| &\le \frac{L_J}{2}\|\mathbf{d}_\mathcal{V}\|^2.\end{aligned} \tag{31}$$

The remainder bound follows by integrating the Jacobian difference along the segment from the reference.

Suppose the increment has linearized objective gap

$$\mathcal{Q}_\mathcal{V}(\mathbf{J}\mathbf{d}_\mathcal{V}) - \mathcal{Q}_\mathcal{V}(\boldsymbol{\Delta}_\mathcal{V}^\star) \le \xi_\mathcal{V}, \qquad \xi_\mathcal{V}\ge 0. \tag{32}$$

Equation 28 makes the left-hand side equal to $(1+\lambda)\|\mathbf{J}\mathbf{d}_\mathcal{V} - \boldsymbol{\Delta}_\mathcal{V}^\star\|^2$. Combining this identity with the remainder bound yields

$$\begin{aligned}\|\boldsymbol{\Delta}_{\mathrm{act}}(\mathbf{d}_\mathcal{V}) - \boldsymbol{\Delta}_\mathcal{V}^\star\| &\le \rho_\mathcal{V},\\ \rho_\mathcal{V} &= \sqrt{\frac{\xi_\mathcal{V}}{1+\lambda}} + \frac{L_J}{2}\|\mathbf{d}_\mathcal{V}\|^2.\end{aligned} \tag{33}$$

Define the actual forecasting gain as

$$G_{\mathrm{act}}(\mathbf{d}_{\mathcal{V}}) = \|\mathbf{e}\|^2 - \|\mathbf{e} - \boldsymbol{\Delta}_{\mathrm{act}}(\mathbf{d}_{\mathcal{V}})\|^2. \tag{34}$$

Let $\mathbf{b}_{\mathcal{V}} = \boldsymbol{\Delta}_{\mathrm{act}}(\mathbf{d}_{\mathcal{V}}) - \boldsymbol{\Delta}^{\star}_{\mathcal{V}}$. Expanding the squared error gives

$$G_{\mathrm{act}}(\mathbf{d}_{\mathcal{V}}) - G(\mathcal{V}) = 2\langle \mathbf{e} - \boldsymbol{\Delta}^{\star}_{\mathcal{V}}, \mathbf{b}_{\mathcal{V}} \rangle - \|\mathbf{b}_{\mathcal{V}}\|^2. \tag{35}$$

Cauchy–Schwarz and Eq. 33 imply

$$\begin{aligned} |G_{\mathrm{act}}(\mathbf{d}_{\mathcal{V}}) - G(\mathcal{V})| &\le E_{\mathcal{V}}, \\ E_{\mathcal{V}} &= 2\|\mathbf{e} - \boldsymbol{\Delta}^{\star}_{\mathcal{V}}\|\rho_{\mathcal{V}} + \rho^2_{\mathcal{V}}. \end{aligned} \tag{36}$$

Consequently, for joint and single-component increments satisfying these conditions at the same reference and penalty strength, and either $\mathcal{U} \in \{\mathcal{V}_{\eta}, \mathcal{V}_u\}$,

$$\begin{aligned} G_{\mathrm{act}}(\mathbf{d}_{\mathcal{V}_{\mathrm{joint}}}) - G_{\mathrm{act}}(\mathbf{d}_{\mathcal{U}}) \ge c_{\lambda}\|(\mathbf{P}_{\mathcal{V}_{\mathrm{joint}}} - \mathbf{P}_{\mathcal{U}})\mathbf{e}\|^2 \\ - E_{\mathcal{V}_{\mathrm{joint}}} - E_{\mathcal{U}}. \end{aligned} \tag{37}$$

Thus the joint-calibration advantage persists when the additional projected-residual gain exceeds the combined optimization and linearization error bounds.

B.5 Correction from Observable Features

Fix the reference base forecaster and learned candidate aggregator. Let $\mathbf{e}_i^{0,R} = \mathbf{y}_i - \widehat{\mathbf{y}}_i^{0,R}$ and $\mathbf{d}_i^R = \mathbf{r}_i^R - \widehat{\mathbf{y}}_i^{0,R}$ denote the base residual and proposal direction. Under a fixed query distribution, assume their squared norms over valid entries have finite expectations after division by n_i. For the observable gate features $\boldsymbol{\phi}_i^R$, define

$$\begin{aligned} A(\boldsymbol{\phi}) &= \mathbb{E}\left[\frac{\|\mathbf{d}_i^R\|^2_{\Omega_i}}{n_i} \,\middle|\, \boldsymbol{\phi}_i^R = \boldsymbol{\phi} \right], \\ B(\boldsymbol{\phi}) &= \mathbb{E}\left[\frac{\langle \mathbf{e}_i^{0,R}, \mathbf{d}_i^R \rangle_{\Omega_i}}{n_i} \,\middle|\, \boldsymbol{\phi}_i^R = \boldsymbol{\phi} \right]. \end{aligned} \tag{38}$$

For the prediction $\widehat{\mathbf{y}}_i^{0,R} + u\mathbf{d}_i^R$, the conditional MSE relative to the base forecast is

$$\mathbb{E}\Big[\delta_i(\widehat{\mathbf{y}}_i^{0,R} + u\mathbf{d}_i^R) - \delta_i^0 \,\Big|\, \boldsymbol{\phi}_i^R = \boldsymbol{\phi} \Big] = u^2 A(\boldsymbol{\phi}) - 2uB(\boldsymbol{\phi}). \tag{39}$$

Minimizing this quadratic over $u \in [0, 1]$ yields

$$u^{\star}(\boldsymbol{\phi}) = \mathrm{clip}_{[0,1]}\left(\frac{B(\boldsymbol{\phi})}{A(\boldsymbol{\phi})} \right) \tag{40}$$

when $A(\boldsymbol{\phi}) > 0$. If $A(\boldsymbol{\phi}) = 0$, then $B(\boldsymbol{\phi}) = 0$, and $u^{\star} = 0$ is one minimizer. For $A(\boldsymbol{\phi}) > 0$, a gate value that strictly improves conditional MSE exists if and only if $B(\boldsymbol{\phi}) > 0$. This characterizes the MSE-optimal correction among gates that are arbitrary measurable functions of the available features. It provides a benchmark for the learned gate.

C Additional Experiment Settings and Results

C.1 Experimental Settings

Evaluation Datasets Statistics of evaluation datasets are presented in Table 9.

Table 9: Statistics of evaluation datasets. "# Channels" denotes the number of channels.

Dataset	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Exchange
# Channels	7	7	7	7	21	321	8
Timestamps	17420	17420	69680	69680	52696	26304	7588
Frequency	1 hour	1 hour	15 mins	15 mins	10 mins	1 hour	1 day
Split ratio	6:2:2	6:2:2	6:2:2	6:2:2	7:1:2	7:1:2	7:1:2

Table 10: **Hyperparameter settings of ROAR** (Chronos-Bolt backbone).

	Base training	Stage I	Stage II
Trainable parameters	η	ψ_c, ψ_u	η, ψ_u
Loss	Pinball	Eq. 8	Eq. 11
Optimizer	AdamW	AdamW	AdamW
Learning rate	3×10^{-4}	3×10^{-4}	η: 3×10^{-5}; ψ_u: 3×10^{-4}
LR schedule	–	Constant	Constant
Weight decay	0.01	0.01	0.01
Batch size	256	256	256
Training steps	10,000	2,500	2,000
Dropout rate	0.2	0.2	0.2
Gradient clipping	1.0	1.0	1.0
Shared settings			
Input length L / horizon H		512 / 64	
Retrieved examples K		10	
ROAR-specific settings			
Opportunity weight β		0.5	
Difficulty threshold t_h		Median normalized base error (0.1546)	
Gate supervision weight λ_{gate}		1	
Reference penalty λ_{ref}		15	
Scorer / gate architecture		MLP ($3 \to 16 \to 1$)	
Initial gate value u_i		≈ 0.1 (bias -2.197)	
IQR quantiles in ϕ_i^R		0.2 / 0.8	

Comparing Baselines We set two sets of baseline methods for comparison, including 7 TSFMs, Moirai2.0 (Liu et al., 2025a), Chronos (Ansari et al., 2024), Chronos-Bolt (Ansari et al., 2024), MOMENT (Goswami et al., 2024), TTM (Ekambaram et al., 2024), Moirai (Woo et al., 2024), TimesFM (Das et al., 2024) and Time-MoE (Shi et al., 2025); and 5 RAG-based time series forecasting (TSF) methods, RAF (Tire et al., 2026), TS-RAG (Ning et al., 2025), Cross-RAG (Lee et al., 2026), PFRP Du et al. (2026) and RATL He et al. (2026). For RAG-based TSF methods, we use the same Chronos-Bolt backbone. PFRP and RATL were not designed for the zero-shot setting, so we adapt them to it. Both adapted methods use the same backbone and the same training and inference protocols as our method and the other baselines. We evaluate all methods with MSE and MAE.

Implementation Details. We adopt Chronos-Bolt (Ansari et al., 2024) as the frozen backbone. Following Ning et al. (2025), the input length and forecasting horizon are set as 512 and 64, and the pre-training corpus, retrieval database, and remaining retrieval settings are the same as TS-RAG. The number of retrieved examples K is set as 10, yielding $J = 20$ candidates. The forecasting module is first trained for 10,000 steps, followed by 2,500 steps for Stage I and 2,000 steps for Stage II. For Chronos-Bolt, ℓ_i in Stage I is the pinball loss over its quantile outputs, each shifted by the same correction. We use AdamW with a constant learning rate, which is set as 3×10^{-4} for the scorer

and gate, and 3×10^{-5} for the forecasting module in Stage II. The weight decay is set as 0.01, the dropout rate as 0.2, the batch size as 256, and the gradient clipping threshold as 1.0. We set $\beta = 0.5$, $\lambda_{\text{gate}} = 1$, and $\lambda_{\text{ref}} = 15$. The threshold t_h is set as the median normalized base error over 12,800 source-training queries before training. The sensitivity to K and λ_{ref} is reported in Appendix C.2. All experiments are conducted on 2 GeForce RTX 4090 GPUs on an Ubuntu platform with 128GB of memory. More hyperparameter settings are listed in Table 10 in Appendix C.

C.2 Sensitivity Analysis

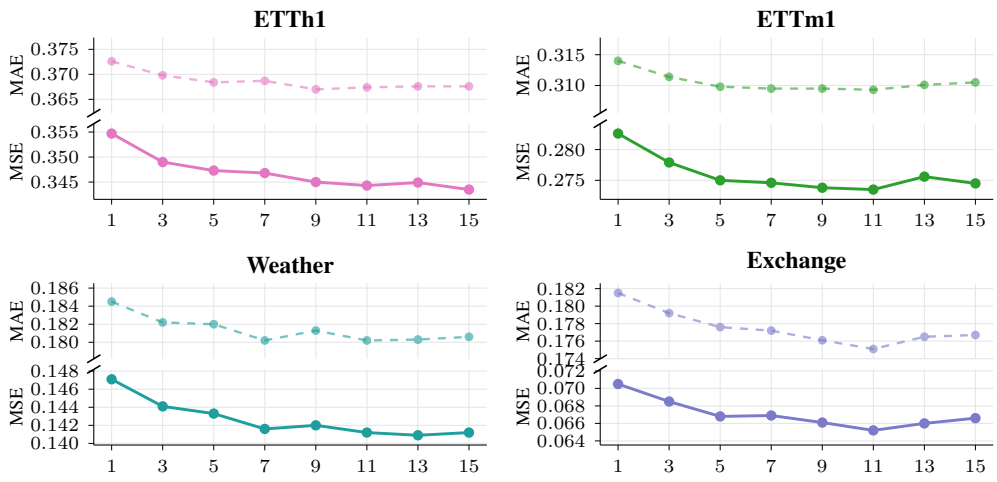
In this section, we examine the parameter sensitivity on three key hyper-parameters: the number of retrieved sequences K, the reference penalty λ_{ref}, and the gate weight λ_{gate}. We follow the zero-shot protocol of the main experiments with the Chronos-Bolt backbone, and evaluate on ETTh1, ETTm1, Weather and Exchange. Specifically, we vary $K \in \{1, 3, \ldots, 15\}$, $\lambda_{\text{ref}} \in \{1, 5, 10, 15, 20\}$ and $\lambda_{\text{gate}} \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$. All other settings, including the retrieval database and training configuration, are kept identical to those in the main experiments. The results are reported in Figs 51, 52 and 53, where the upper and lower panels of each subplot show MAE and MSE, respectively.

Number of retrieved sequences K. As shown in Fig. 51, the forecasting error generally decreases as K increases from 1, since aggregating more neighbors provides a more reliable reference for the backbone. Compared with K=1, the best MSE is reduced by 3.2%, 3.2%, 4.2% and 7.5% on ETTh1, ETTm1, Weather and Exchange, respectively. Most of this gain is obtained by K=5 on ETTh1 and ETTm1 and by K=7 on Weather, whereas Exchange keeps improving until K=11. Beyond the best value, the error fluctuates slightly and rises by up to 2.1% on Exchange, possibly because additional neighbors of lower relevance introduce noise.

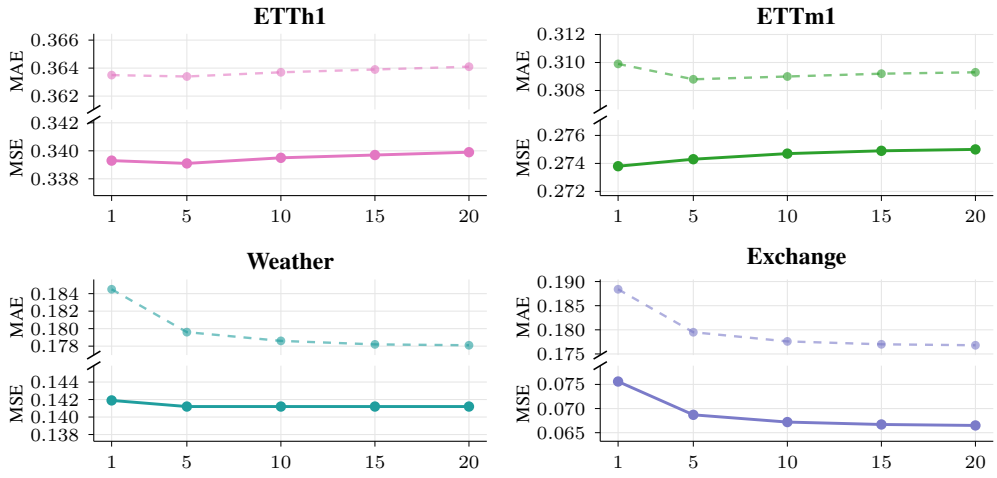
Reference penalty λ_{ref}. As shown in Fig. 52, ROAR is insensitive to λ_{ref} on ETTh1 and ETTm1, where the MSE varies by less than 0.5% across the whole range. On Weather and Exchange, a too small penalty (λ_{ref}=1) leads to noticeably higher errors, e.g., the MAE on Weather rises from 0.180 to 0.185 and the MSE on Exchange from 0.069 to 0.076, indicating that without sufficient regularization the fused prediction may deviate excessively from the reference. For $\lambda_{\text{ref}} \geq 5$, the performance is stable on ETTh1, ETTm1 and Weather, while Exchange benefits slightly from a larger penalty.

Gate weight λ_{gate}. As shown in Fig. 53, increasing λ_{gate} yields a mild but monotonic reduction in MSE on all datasets, with the largest relative gain of about 1.2% on ETTm1, while the MAE remains almost unchanged. This suggests that the gating term mainly suppresses large errors, and that ROAR is robust to its exact weight.

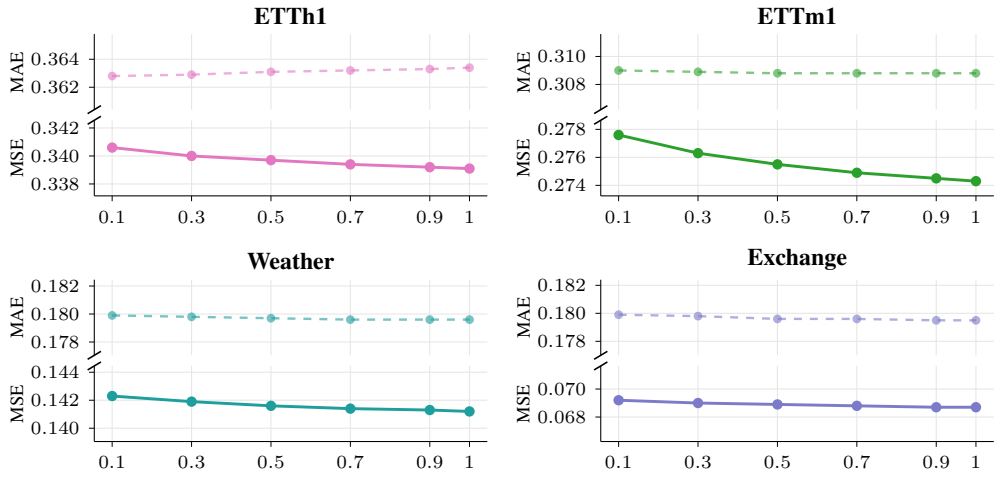
Overall, ROAR maintains stable performance across a broad range of values for all three hyper-parameters. The only setting to avoid is an overly small reference penalty, and moderate choices of K and λ_{ref} are sufficient to obtain near-optimal results, which indicates that ROAR does not rely on delicate hyper-parameter tuning.



(1) Number of retrieved sequences K



(2) Reference penalty λ_{ref}



(3) Gate weight λ_{gate}

Figure 5: Hyper-parameter sensitivity of ROAR. In each panel, the upper and lower parts show MAE and MSE, respectively.

C.3 EFFICIENCY ANALYSIS.

To assess the computational overhead introduced by ROAR, we compare it with the RAG-based baselines in terms of per-iteration latency, FLOPs, total parameters and trainable parameters. All methods share the same Chronos-Bolt backbone and retrieval setting, and the latency is averaged over ETTh1, ETTm1 and Weather under identical hardware and batch size. We further report the two intermediate variants of ROAR to isolate the cost of each stage. The results are summarized in Table 11.

Several observations can be made. First, the base variant of ROAR (w/o Stage I & II) is already lightweight: it has the fewest total parameters among all methods, and its trainable parameters (2.27M) are less than half of those of TS-RAG (4.78M) and about a quarter of those of Cross-RAG (8.71M). Second, Stage I introduces almost no overhead. It keeps the FLOPs and total parameters unchanged, increases the latency by only 0.31 ms, and involves merely 162 trainable parameters, since the backbone and fusion module are frozen and only two small MLPs are optimized. Third, the full ROAR roughly doubles the latency, FLOPs and total parameters compared with the other methods, because Stage II requires an additional forward pass through the reference branch. Nevertheless, the number of trainable parameters remains 2.27M, and the absolute latency stays below 16 ms per iteration.

Overall, the performance gains of ROAR come with a moderate and predictable inference cost, while its trainable parameters remain substantially fewer than those of existing RAG-based methods, making it practical for deployment in the zero-shot setting.

Table 11: Efficiency comparisons across RAG-based TSF methods. Iteration time is averaged over ETTh1, ETTm1 and Weather.

Method	Avg Iter Time (ms)↓	FLOPs (G)↓	Total Params (M)↓	Trainable Params (M)↓	Trainable Ratio (%)↓
TS-RAG	8.14	6.96	210.07	4.7754	2.273
Cross-RAG	7.42	6.96	214.01	8.7122	4.071
ROAR w/o Stage I & II	7.69	6.92	207.56	2.2654	1.091
ROAR w/o Stage II	8.00	6.92	207.56	1.6×10^{-4}	7.8×10^{-5}
ROAR (full)	15.68	13.84	415.12	2.2655	0.546

C.4 VISUALIZATIONS

We present more detailed forecasting visualizations between our ROAR and the ablative version "ROAR w/o Stages I & II" and baseline methods TS-RAG and Cross-RAG in Figs.6 and 7.

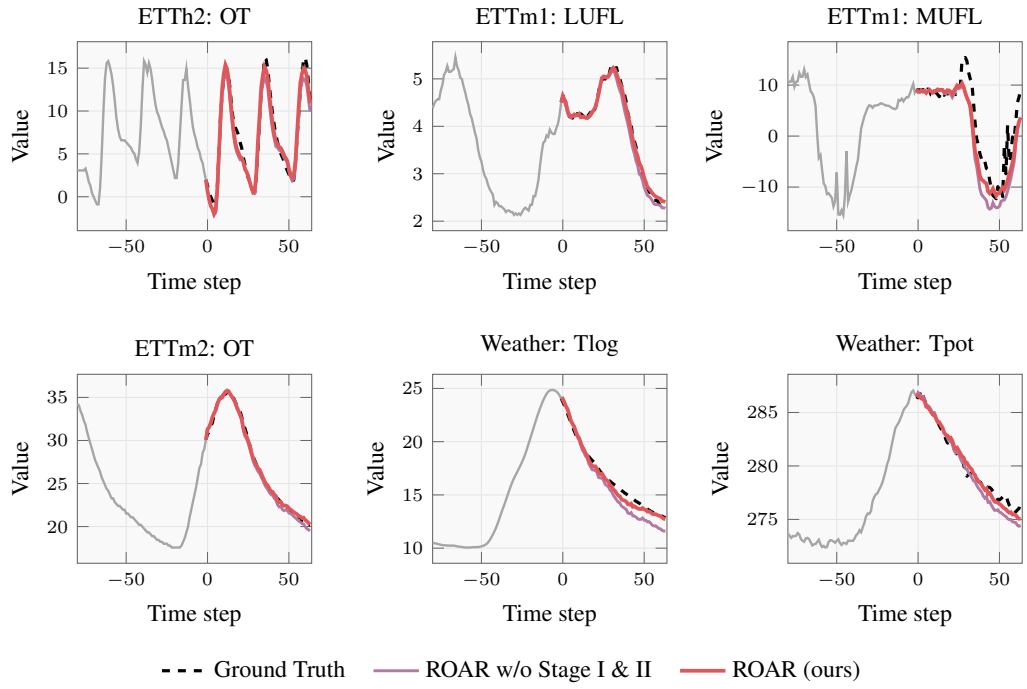


Figure 6: Additional forecasting visualizations comparing ROAR with the ablative variant ROAR w/o Stage I & II on ETTh2, ETTm1, ETTm2 and Weather.

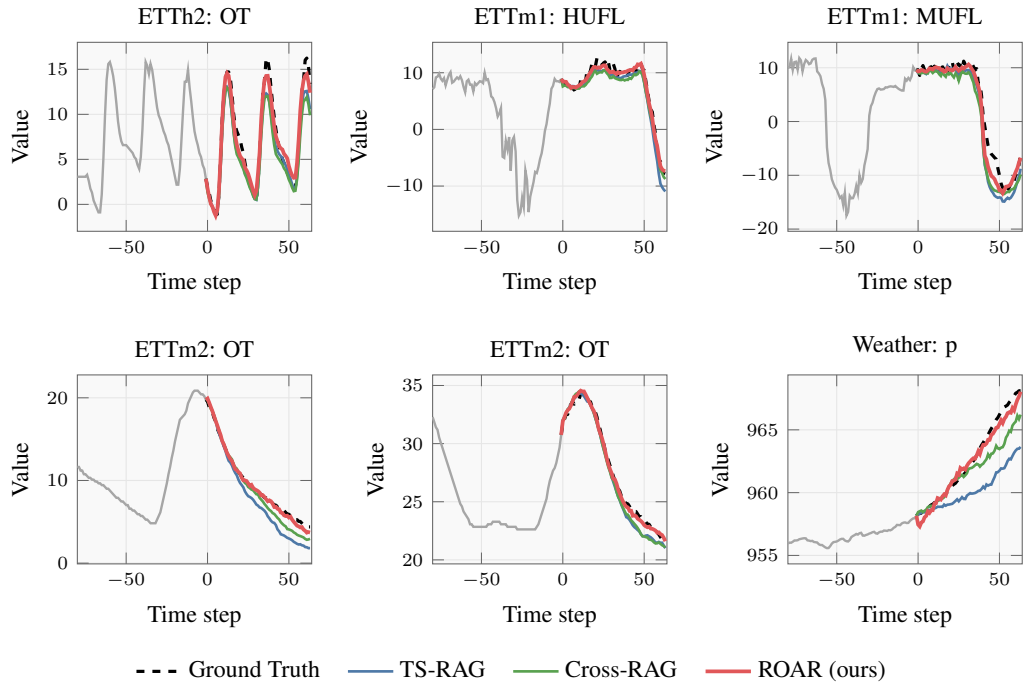


Figure 7: Additional forecasting visualizations comparing ROAR with TS-RAG and Cross-RAG on ETTh2, ETTm1, ETTm2 and Weather.