RELIABILITY-CONDITIONED MULTI-MODAL RETRIEVAL FOR TIME-SERIES ANALYSIS WITH IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series analysis increasingly combines numerical trajectories with textual context, yet the two modalities provide complementary and unevenly reliable evidence. Existing language-model pipelines improve temporal representation or retrieve demonstrations through lexical, semantic, or temporal similarity, but they usually apply one retrieval rule to every query or require downstream model feedback to learn selection. We introduce Reliability-Conditioned Query-Adaptive Similarity for Hybrid retrieval (RC-QASH), a multi-modal In-Context Learning (ICL) approach that treats text/time-series fusion as a conservative query-level reliability decision. Using only nested out-of-fold (OOF) records from the demonstration bank, its gate summarizes view-specific class distributions, confidence, disagreement, and ranking overlap, then either retains the Hybrid anchor or shifts the modality balance. On Finance and Weather, the full pipeline achieves the best cross-domain point estimates in the DeepSeek baseline comparison, with equal-domain mean Accuracy of 57.75% and PRIMARY of 49.19%, where PRIMARY averages balanced accuracy and macro-F1. Relative to Time-Series Top-3, the strongest reported comparator on these aggregate metrics, these scores correspond to improvements of 6.20% and 6.52%, respectively. Shared banks, prompt rendering, and fixed decoders isolate retrieval changes. Matched DeepSeek and Llama comparisons evaluate the controller on changed prompts. The gate concentrates its changes on queries where static Hybrid retrieval is weaker, and these comparisons show positive mean gains within the selected subsets. These results show that retrieval geometry can support selective multi-modal adaptation without downstream-output supervision.

1 INTRODUCTION

Time-series analysis supports decisions in settings where a numerical trend is inseparable from its surrounding context. Financial movement depends on the event and market regime described in text, while weather trends combine recent measurements with location-specific conditions. MTBench formalizes this setting through paired text and time series in Finance and Weather prediction tasks (Chen et al., 2025). Similar trajectories can arise under different events, and similar descriptions can accompany different local dynamics. Effective prediction must therefore identify both relevant historical cases and the modality that makes each case informative.

Temporal encoders capture local shapes, periodic structure, and longer histories, while language-model approaches serialize numerical observations or align temporal representations with language. In-Context Learning (ICL) offers a complementary route: a Large Language Model (LLM) predicts from labeled examples in its prompt without task-specific parameter updates (Brown et al., 2020). Demonstration selection then becomes part of the prediction procedure. Useful cases must match the aspects that determine a query's future trend, not simply resemble its wording.

These advances leave a reliability bottleneck in paired temporal prediction. Text-only retrieval can overlook dynamics, whereas time-series retrieval can miss the event semantics that distinguish otherwise similar trajectories. A global Hybrid weight combines both views but assumes that their relative value is constant across queries. Recent alternatives either retain a global temporal/semantic

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.

mixture or learn query-dependent coefficients from downstream correctness (Liu et al., 2026; Xie, 2026). Neither directly answers whether retrieval geometry alone can identify when a static mixture should change, before calling the downstream model.

We address this question with Reliability-Conditioned Query-Adaptive Similarity for Hybrid retrieval (RC-QASH). Its central insight is to inspect the evidence behind a ranking rather than similarity alone. Concentrated neighbor labels indicate within-view confidence; disagreement between label distributions reveals cross-view conflict; ranking overlap shows whether a weight change can alter the prompt. A controller learns from nested Out-of-Fold (OOF) demonstration-bank records to choose a temporal-oriented, anchor, or text-oriented action. The anchor remains the default unless an alternative passes a bank-selected confidence threshold. This couples a query-specific reliability estimate to a selective change in demonstration retrieval.

We evaluate the design on Finance and Weather using lexical, dense semantic, temporal, and static-fusion references under a shared prompt and parser. DeepSeek supports the main baseline comparison, while DeepSeek and Llama support matched controller analyses. We separate complete-pipeline performance, static-anchor weakness on selected queries, and recovery on that same subset, distinguishing retrieval's overall benefit from the controller's additional effect. Section 5 presents the results.

Our contributions are:

- We formulate query-dependent modality reliability through within-view confidence, cross-view conflict, and ranking agreement.
- We develop a selective text/time-series router using demonstration-bank supervision and an anchor fallback when directional evidence is insufficient.
- We distinguish pipeline gains from controller effects through baseline comparison, selected-query weakness, prompt recovery, and backbone transfer.

2 Related Work

2.1 In-Context Demonstration Selection

ICL depends on demonstration content, label mapping, order, and coverage (Min et al., 2022; Lu et al., 2022; Liu et al., 2022). Accounts based on latent-concept inference and implicit algorithm execution (Xie et al., 2022; Akyürek et al., 2023) explain its capabilities but do not specify which examples to retrieve. EPR and UDR compare random, lexical, dense, and learned selection (Rubin et al., 2022; Li et al., 2023). Our corresponding controls use random examples, probabilistic BM25 retrieval (Robertson & Zaragoza, 2009), and dense text matching under one renderer.

Learned retrievers exploit downstream feedback: EPR learns example compatibility, UDR transfers list-wise selection across tasks, and MDR adapts selection to the inference model (Rubin et al., 2022; Li et al., 2023; Wang et al., 2024a). MoD learns retrieval experts (Wang et al., 2024c), and CASE reduces calls for subset search (Purohit et al., 2025). These methods optimize example selection; RC-QASH instead holds the bank and base scorers fixed and chooses a reliable mixture for each temporal query. Generic similarity lacks a modality-reliability estimate, while feedback-trained retrievers require downstream evaluations to obtain their supervision.

2.2 Multi-Modal Retrieval and Fusion

Multi-modal few-shot models show that interleaved examples from different modalities can condition a language model without task-specific fine-tuning (Alayrac et al., 2022; Zhao et al., 2024; Sun et al., 2024b). Their behavior depends separately on retrieval, ordering, and the modality content of each demonstration (Qin et al., 2024; Xu et al., 2025), which is why we treat the ordered demonstration prompt rather than its unordered membership as the intervention.

Adaptive multi-modal gating is a broad idea with established antecedents (Arevalo et al., 2017). MAF applies it to few-shot retrieval by predicting query-conditioned face, scene, and text coefficients from actual downstream correctness over 66 weight triples (Xie, 2026). DRUM uses Large Vision–Language Model (LVLM) feedback and list-wise ranking to train a multi-modal demonstration

encoder (Yi-Ge et al., 2025), while GenICL directly learns language-model preferences because metric-learning surrogates can miss downstream utility (Zhang et al., 2025b). Their downstream supervision requires repeated model evaluation rather than pre-inference reliability alone, limiting use when model calls are unavailable during controller training.

2.3 Language Models for Time-Series Analysis

Temporal encoders exploit patches, periodicity, variate-wise attention, and multi-scale mixing (Nie et al., 2023; Wu et al., 2023; Liu et al., 2024; Wang et al., 2024b). Timer-XL extends context length (Liu et al., 2025); TiRex uses recurrent state tracking (Auer et al., 2025); and Chronos-2 shares information across related series and covariates through group attention (Ansari et al., 2025). These advances broaden numerical forecasting without specifying which labeled multi-modal demonstrations to retrieve.

Language-model adaptations serialize numbers (Gruver et al., 2023; Xue & Salim, 2023), reprogram temporal patches (Jin et al., 2024), or align temporal and textual representations (Sun et al., 2024a). General temporal pretraining learns reusable sequence representations (Zhou et al., 2023; Goswami et al., 2024; Ansari et al., 2024), while MTBench evaluates reasoning from paired text and time series (Chen et al., 2025).

RationaleTS is the closest task-level neighbor because it retrieves label-conditioned rationale priors with a temporal/semantic score (Liu et al., 2026). Its single global mixture, however, cannot respond to query-level agreement or conflict between the two retrieval views. More broadly, representation and forecasting advances do not determine which labeled multi-modal examples should condition a new query, leaving demonstration reliability as a separate unresolved problem.

3 Problem Formulation for Multi-Modal Time-Series Analysis

We denote a query by q, its textual context by x_q, and its observed time series by z_q. The unknown trend label y_q belongs to the fixed class set $\mathcal{C}$. The demonstration bank $\mathcal{D}$ contains n labeled examples; index i identifies an example with text x_i, time series z_i, and label y_i. Thus, the query and bank are

$$q = (x_q, z_q), \qquad y_q \in \mathcal{C}, \qquad \mathcal{D} = \{(x_i, z_i, y_i)\}_{i=1}^{n}. \tag{1}$$

A retriever returns an ordered list of K_{demo} demonstrations, which is rendered with the same instruction and query into an ICL prompt for a downstream language model. The primary evaluation uses three trend classes in both domains. Finance's five native output classes are collapsed into negative, neutral, and positive; the five-class scores remain a secondary analysis.

For query q and bank example i, let $s_{\text{text}}(q, i)$ denote text-embedding cosine similarity and $s_{\text{ts}}(q, i)$ denote similarity between normalized temporal representations. The text coefficient α controls their balance, with $1 - \alpha$ weighting the temporal view. Hybrid retrieval ranks examples by the combined score $s_\alpha(q, i)$:

$$s_\alpha(q, i) = \alpha s_{\text{text}}(q, i) + (1 - \alpha) s_{\text{ts}}(q, i). \tag{2}$$

The static text weight is α_{anchor}. Two alternatives, α_{ts} and α_{text}, shift retrieval toward temporal and textual similarity, respectively. All three remain text coefficients, ordered as

$$\alpha_{\text{ts}} < \alpha_{\text{anchor}} < \alpha_{\text{text}}. \tag{3}$$

The prompt, demonstration count, candidate actions, and their numerical configuration are fixed before external evaluation and reported in Appendix A.

We report accuracy, balanced accuracy (mean class recall), and macro-F1 (mean class-specific F1), using explicit class lists. The latter two values form the class-balanced summary score PRIMARY:

$$\text{PRIMARY} = \tfrac{1}{2}(\text{BalancedAccuracy} + \text{MacroF1}). \tag{4}$$

We use this class-balanced score to assess recognition across trend categories under class imbalance, with accuracy as a complementary metric. All metrics are percentages; Δ denotes a difference in percentage points, with its direction specified in each comparison. Unrecognized outputs count as incorrect and do not form an extra class in macro metrics.

4 RELIABILITY-CONDITIONED MULTI-MODAL RETRIEVAL

4.1 OVERVIEW

Figure 1 organizes RC-QASH into two paths. The upper path fits the controller entirely within the labeled demonstration bank. The lower path applies the resulting controller to an external query: retrieve three views, summarize their reliability, select a conservative action, and construct an ordered prompt. The controller changes only candidate ranking; the instruction, demonstration count, language model, and parser remain fixed.

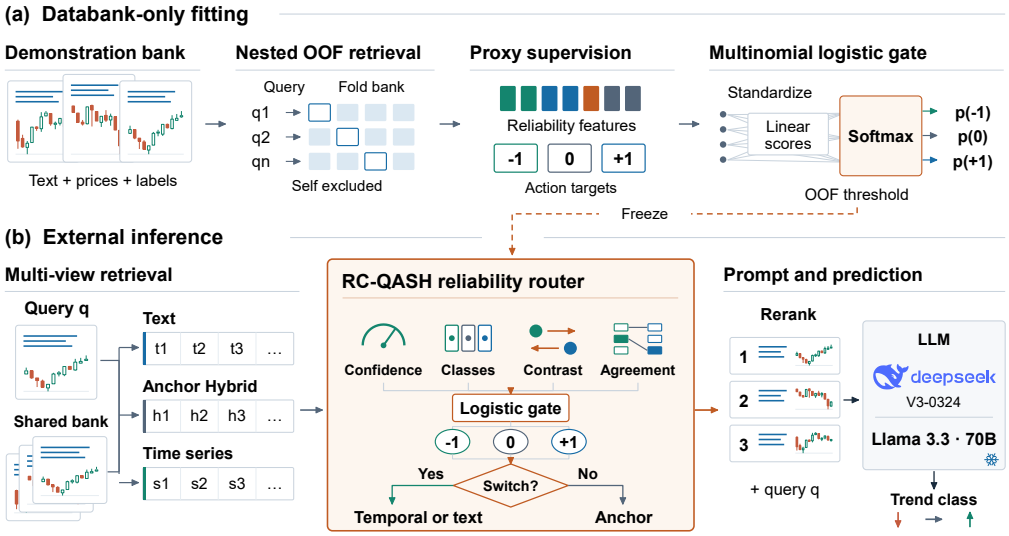

Figure 1: Overview of RC-QASH. (a) Databank-only OOF records train a multinomial logistic gate and select its threshold. (b) Retrieval confidence, class distributions, view contrast, and rank agreement guide conservative reranking before prompting DeepSeek or Llama. Candlesticks summarize observed Finance input prices; Weather uses the same pipeline.

4.2 DATABANK-ONLY CONTROLLER FITTING

To limit controller complexity with small databanks, we fix three candidate mixtures before fitting. For query q, the action a_q identifies a temporal-oriented shift (-1), the anchor (0), or a text-oriented shift $(+1)$. These mixture actions differ from the pure-modality feature views. The mapping $\alpha(a_q)$ gives the text weight associated with each action:

$$\alpha(a_q) = \begin{cases} \alpha_{\text{ts}}, & a_q = -1 \quad \text{(temporal-oriented)}, \\ \alpha_{\text{anchor}}, & a_q = 0 \quad \text{(anchor)}, \\ \alpha_{\text{text}}, & a_q = +1 \quad \text{(text-oriented)}. \end{cases} \tag{5}$$

The training target t_q is the action that repairs the anchor's neighbor-label prediction, or zero when no repair is selected. Each candidate action a retrieves under its mixture score $s_{\alpha(a)}$. Finance decodes its leading labels by three-class majority vote, with the rank-one label breaking a tie; Weather uses the maximum class mass in Equation 7. If the anchor is correct, $t_q = 0$. Otherwise, a uniquely correct alternative supplies the target. If both alternatives are correct, the one with greater prior-corrected, rank-discounted gold-class support supplies the target; a support tie returns zero. If neither repairs the anchor, $t_q = 0$. This is proxy-repair supervision, not downstream LLM utility.

We generate training records with nested cross-validation on the demonstration bank. In every fold, held-out queries retrieve only from the fold's training partition. Finance uses stratified group splits to keep duplicate groups together; Weather uses stratified splits. A standardization layer and a class-balanced multinomial logistic regression form the gate, implemented with scikit-learn (Pedregosa et al., 2011). The decision threshold is selected by inner out-of-fold PRIMARY and ties favor the

larger threshold. We denote the selected domain-specific threshold by τ_d, where d denotes Finance or Weather.

4.3 Multi-View Retrieval

The view index v denotes text-only, time-series-only, or anchor-Hybrid retrieval over the same bank. We retain K_{pool} ranked examples per view and use the leading K_{stat} neighbors for label statistics. Here $i_{v,r}$ identifies the example at rank r in view v, with rank one highest and $y_{i_{v,r}}$ its label. For class c, let π_c denote its fraction in the current bank. The neighbor weight $\omega_{v,r}(q)$ corrects for class frequency and discounts lower ranks:

$$\omega_{v,r}(q) = \frac{1}{\pi_{y_{i_{v,r}}} \log_2(r+1)}. \tag{6}$$

Let $p_v(c \mid q)$ denote the normalized support for class c in view v. The indicator $\mathbb{I}[y_{i_{v,r}} = c]$ equals one when the neighbor has label c and zero otherwise. Normalizing the class-specific weights yields

$$p_v(c \mid q) = \frac{\sum_{r=1}^{K_{\text{stat}}} \mathbb{I}[y_{i_{v,r}} = c]\, \omega_{v,r}(q)}{\sum_{r=1}^{K_{\text{stat}}} \omega_{v,r}(q)}, \qquad c \in \mathcal{C}. \tag{7}$$

Prior correction reduces the influence of bank class imbalance; rank discount gives more weight to closer demonstrations.

4.4 Retrieval-Reliability Representation

The reliability vector $\phi(q)$ contains d_ϕ real-valued features in four groups. View-confidence features describe top similarity, the top-K_{stat} versus tail gap, top-one margin, label diversity, purity, and entropy. Distribution features retain each view's class probabilities. We denote purity, the largest class probability, by $\rho_v(q)$, and normalized entropy, which measures label dispersion, by $\bar{H}_v(q)$. Writing $|\mathcal{C}|$ for the number of classes and using $0 \log 0 = 0$, we compute

$$\rho_v(q) = \max_{c \in \mathcal{C}} p_v(c \mid q), \qquad \bar{H}_v(q) = -\frac{1}{\log |\mathcal{C}|} \sum_{c \in \mathcal{C}} p_v(c \mid q) \log p_v(c \mid q). \tag{8}$$

Contrast features compare text and time-series purity, entropy, and score gap. For distributional disagreement, p_{text} and p_{ts} denote the two class-probability vectors for the same query, and m denotes their element-wise mean. With KL denoting Kullback–Leibler Divergence, their Jensen–Shannon Divergence (JSD) (Lin, 1991) is

$$\text{JSD}(p_{\text{text}}, p_{\text{ts}}) = \tfrac{1}{2}\,\text{KL}(p_{\text{text}} \| m) + \tfrac{1}{2}\,\text{KL}(p_{\text{ts}} \| m), \quad m = \tfrac{1}{2}(p_{\text{text}} + p_{\text{ts}}). \tag{9}$$

Agreement features measure overlap at K_{stat} and K_{pool}, each unimodal view's overlap with the anchor, and whether the maximum-mass classes of the two unimodal views agree. Feature distributions use Equation 7 in both domains, independently of the target decoder. No external gold label or downstream output enters these features.

4.5 Conservative Reliability Routing

The multinomial logistic gate applies one softmax to linear scores of standardized $\phi(q)$. Its output $p_a(q)$ is the probability of action a, rather than class support $p_v(c \mid q)$; $p_0(q)$ is the anchor probability. We denote the higher-probability directional action by $a^*(q)$:

$$a^*(q) = \operatorname*{arg\,max}_{a \in \{-1,+1\}} p_a(q), \tag{10}$$

A directional tie favors -1. The final accepted action, denoted by $\widehat{a}_q$, uses this alternative only when its probability exceeds $p_0(q)$ and meets the domain threshold τ_d:

$$\widehat{a}_q = \begin{cases} a^*(q), & p_{a^*(q)}(q) > p_0(q) \text{ and } p_{a^*(q)}(q) \geq \tau_d, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

This shrinkage rule makes the anchor the explicit fallback when reliability evidence is weak.

The selected action changes only the retrieval score in Equation 2. Prompt format, demonstration count, downstream model, decoding temperature, and output parser remain identical.

4.6 Prompt Construction and Conditional Evaluation

The selected weight reranks the bank and produces an ordered K_{demo}-example list. An action need not change this list, so we compare the resulting ordered demonstration identifiers with the anchor prompt. Only changed lists require a new LLM call; unchanged queries reuse the anchor response. This avoids repeated calls for identical prompts and conditions the estimate on the realized anchor responses. All queries still require feature extraction and gate inference. Appendix M (page 20) gives the instruction templates, demonstration format, and output-label constraints used by the evaluators.

5 Experimental Evaluation

5.1 Experimental Setup

Data and lockbox protocol. Finance contains a 375-query demonstration bank and a disjoint 375-query external test set. Weather contains 979 bank queries and 980 external test queries. The repaired task prompt is frozen before retrieval comparisons. Weather prompt repair alone raises the controlled Vanilla accuracy from 43.78% to 47.14% and PRIMARY from 35.24% to 38.87%, so all methods use the same repaired version. Gate fitting, target construction, and threshold search use only the demonstration bank. No external label is used as a feature or for post-hoc refitting.

The same controller configuration is used in every formal external run. Its development chronology is documented in Appendix K.

Models and repetitions. The primary decoder is DeepSeek-V3-0324, the March 2025 release of the DeepSeek-V3 Mixture-of-Experts model (671B total parameters; 37B activated per token) (DeepSeek-AI, 2024; 2025). Here, B denotes billion parameters. Llama-3.3-70B-Instruct (70B parameters, December 2024 release) supplies a single-run cross-backbone replication (Meta, 2024). Appendices A and B give decoding settings and per-repetition results. Each response vector replaces only changed prompts; Appendix B.1 distinguishes the headline and matched response groups.

Retrieval implementation. Text retrieval uses cached 1024-dimensional Qwen3-Embedding-0.6B representations (Zhang et al., 2025a). Temporal retrieval uses Weather's 168-hour window and a return-path representation for Finance. Each fold fits temporal scaling only on its retrieval bank, then normalizes bank and query vectors. The three mixtures reuse the same text and temporal similarity matrices. Appendix A.1 specifies the representations and gate-fitting configuration.

Baselines. Vanilla uses the repaired instruction and query without demonstrations, as in standard language-model prompting comparisons (Brown et al., 2020). Random Top-3, BM25 Text Top-3, and Dense Text Top-3 correspond to the random, sparse lexical, and dense semantic selector families compared in EPR and UDR (Rubin et al., 2022; Li et al., 2023); BM25 follows the probabilistic relevance framework (Robertson & Zaragoza, 2009). These are selector-family controls under MTBench, not the trained EPR/UDR models.

Time-Series Top-3 is a modality-isolation control that retrieves only by the frozen temporal similarity. The static Hybrid anchor applies Equation 2 with α_{anchor}, representing the global fusion assumption also examined by retrieval-based temporal reasoning (Liu et al., 2026). RC-QASH adds only the reliability gate and conservative action rule. All methods share the repaired prompt, candidate bank, ordered demonstration renderer, model, and parser. The numerical retrieval and gate configuration is collected in Appendix A; response-vector counts appear in Appendix B, and a bank-selected static mixture is reported in Appendix D.

Evaluation reporting. DeepSeek contributes three RC-QASH response vectors per domain, and Llama contributes one. Parser failures count as incorrect. The main text reports descriptive performance; Appendix H gives query-resampling intervals and paired tests conditional on these vectors. Repetition details remain in Appendices B and G.

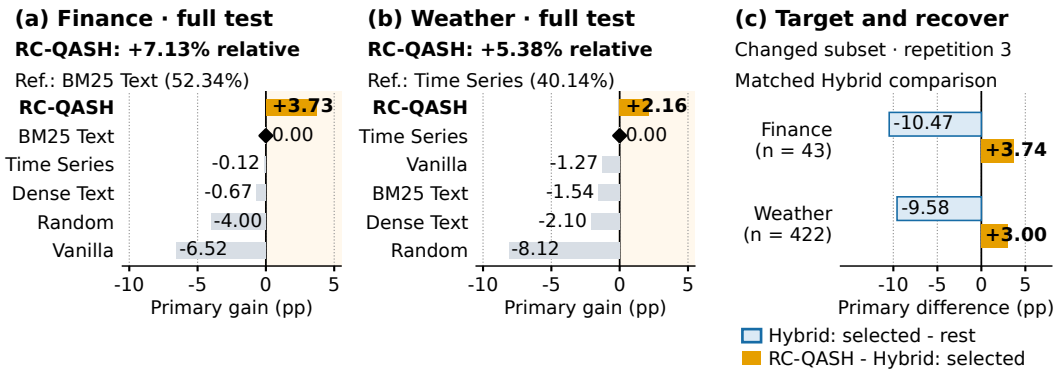

Figure 2: PRIMARY gains with DeepSeek-V3-0324. Panels (a–b) use a shared difference scale: zero is the strongest plotted baseline (BM25 Text in Finance; Time Series in Weather). All retrieval rows use Top-3 examples; absolute scores are in Table 1. Headings report relative percentage gains. Panel (c) separates Hybrid weakness (selected minus remaining queries) from recovery on the same selected queries (RC-QASH minus Hybrid, repetition 3). Its horizontal axis measures percentage-point differences.

5.2 MAIN COMPARISON ACROSS RETRIEVAL BASELINES

Table 1 compares the selector families introduced in Section 5.1 under one DeepSeek protocol. Vanilla provides the no-demonstration reference. Random Top-3 improves Finance accuracy to 54.67% but reduces Weather accuracy from 47.14% to 44.80% and PRIMARY from 38.87% to 32.02%. Demonstrations are therefore not beneficial merely because they are present; irrelevant examples can amplify the domain's class imbalance.

Table 1: DeepSeek-V3-0324 retrieval comparison (671B total / 37B active parameters). Scores are percentages; bold marks the best point estimate. Relative Gain compares RC-QASH with the strongest other row in each column.

	Finance		Weather		Equal-Domain Mean	
Method	Accuracy	PRIMARY	Accuracy	PRIMARY	Accuracy	PRIMARY
Vanilla	46.13	45.82	47.14	38.87	46.64	42.35
Random Top-3	54.67	48.34	44.80	32.02	49.73	40.18
BM25 Text Top-3	56.53	52.34	50.92	38.60	53.73	45.47
Dense Text Top-3	55.20	51.67	51.53	38.04	53.37	44.86
Time-Series Top-3	56.00	52.22	52.76	40.14	54.38	46.18
RC-QASH	**60.09**	**56.07**	**55.41**	**42.30**	**57.75**	**49.19**
Relative Gain	+6.30%	+7.13%	+5.02%	+5.38%	+6.20%	+6.52%

The text retrievers are more reliable but remain domain dependent. BM25, the sparse lexical control used in ICL retrieval studies, is the strongest non-adaptive Finance selector in Table 1, at 56.53% accuracy and 52.34% PRIMARY. Its Weather PRIMARY score is 38.60%, essentially returning to the Vanilla level. Dense Text Top-3 raises Weather accuracy to 51.53%, yet lowers PRIMARY to 38.04%. Its recall moves toward the decreasing class by 13.50 points while stable and increasing recall fall by 8.39 and 7.66 points. Lexical or semantic similarity can therefore retrieve useful textual analogues without preserving balanced temporal decisions.

Time-Series Top-3 addresses the complementary weakness. It reaches 40.14% PRIMARY on Weather and 52.22% on Finance, outperforming Dense Text Top-3 on the balanced metric in both domains. It still discards textual evidence and does not match the adaptive joint system. RC-QASH reaches 60.09/56.07 on Finance and 55.41/42.30 on Weather (Accuracy/PRIMARY), producing the best point estimate in every domain column of Table 1. Its Equal-Domain Mean is 57.75% Accuracy and 49.19% PRIMARY, compared with 53.73/45.47 for BM25 and 54.38/46.18 for Time-Series Top-3. Relative gains from these displayed scores are 6.20% in equal-domain Accuracy and 6.52% in PRIMARY

(Appendix A.2). BM25 is the comparator for Finance; Time-Series Top-3 is the comparator for Weather and the equal-domain mean.

Appendix H reports the corresponding query-level intervals and paired transition tests.

5.3 Controller Targeting of Static-Anchor Weaknesses

Figure 2(c) summarizes the changed-prompt mechanism in PRIMARY. Table 2 asks where the controller intervenes and what happens on those queries. Membership is determined by an ordered prompt change, not by whether a downstream response is correct. Panel (a) shows lower static-Hybrid performance on the selected subset in both domains. Its PRIMARY score is 10.47 points below the remaining queries in Finance and 9.58 points below them in Weather. The corresponding accuracy gaps are 9.68 and 7.28 points. The gate therefore concentrates its interventions on queries that expose weaknesses of the static prompt.

Panel (b) evaluates the same changed-prompt subset in the third DeepSeek response vector. Relative to Hybrid, accuracy improves by 4.65 points in Finance and 4.74 in Weather; PRIMARY improves by 3.74 and 3.00 points. The two panels connect lower anchor performance with recovery on the same queries. Appendix C summarizes the cross-vector and Llama comparisons.

Table 2: The controller targets weaknesses of the static Hybrid prompt and can improve the selected queries. (a) Hybrid performance on queries with and without an ordered demonstration prompt change. (b) Descriptive Hybrid–RC-QASH comparison on the changed-prompt subset in DeepSeek-V3-0324 repetition 3. Values are percentages and deltas are percentage points; the cross-vector summary appears in Appendix C.

(a) Hybrid Weakness Enrichment

			Accuracy			PRIMARY		
Domain	Selected	Coverage	Selected	Others	Gap	Selected	Others	Gap
Finance	43/375	11.5%	51.16	60.84	−9.68	46.38	56.85	−10.47
Weather	422/980	43.1%	49.53	56.81	−7.28	35.32	44.90	−9.58

(b) Realized Intervention on the Selected Subset

		Accuracy			PRIMARY		
Domain	Changed	Hybrid	RC-QASH	Δ	Hybrid	RC-QASH	Δ
Finance	43	51.16	**55.81**	+4.65	46.38	**50.11**	+3.74
Weather	422	49.53	**54.27**	+4.74	35.32	**38.32**	+3.00

5.4 Sparse and Domain-Dependent Adaptation

Finance changes both the weight and ordered prompt on 43 of 375 queries. Weather selects 463 non-anchor actions, but 41 leave the ordered list unchanged; 422 prompts require recomputation. The gate retains the anchor weight for 88.5% of Finance and 52.8% of Weather queries, adapting a domain-dependent subset rather than applying one new global mixture.

Table 3: Sparse prompt interventions on the external test sets. N is the test-set size; coverage is the fraction whose ordered Top-3 prompt differs from the static anchor.

Domain	N	Non-Anchor	Changed	Coverage
Finance	375	43	43	11.5%
Weather	980	463	422	43.1%

5.5 Heterogeneous Downstream Intervention Utility

Retrieval-label repair is a proxy for downstream benefit. A historical development analysis evaluated 340 changed queries (58 Finance and 282 Weather) in three matched repetitions, yielding 1,020 anchor–proposal pairs from 2,040 model calls. Its retained aggregate reports 29.3% beneficial and 19.0% harmful changes for Finance, and 29.8% beneficial and 22.0% harmful changes for Weather. Accepting every proposal would add 0.62 and 1.06 domain-wide accuracy points, respectively, despite heterogeneous per-query outcomes.

Historical out-of-fold utility predictors performed weakly: pooled beneficial/harmful Areas Under the Receiver Operating Characteristic Curve (AUCs) remained near chance, with the best reported domain-specific beneficial AUC around 0.61. These aggregate observations motivate distinguishing retrieval reliability from downstream prompt utility; a learned utility gate is not part of RC-QASH.

Table 4: Matched proposal utility reported by the retained historical development aggregate. Beneficial and harmful rates are measured among changed queries; domain-wide gain is the accept-all accuracy change in percentage points.

Domain	Changed	Beneficial	Harmful	Domain Gain
Finance	58	29.3%	19.0%	+0.62
Weather	282	29.8%	22.0%	+1.06

5.6 Proxy-to-Downstream Transfer

Table 5 compares proxy and downstream PRIMARY gains using the matched response groups in Appendix B.1. Every cell retains a positive direction, but the downstream effect is smaller. The downstream/proxy ratios are 17% and 37% for DeepSeek on Finance and Weather, and 46% and 80% for Llama. These are descriptive comparisons of different prediction functions: better neighbor-label support does not guarantee that the LLM will use the demonstrations in the same way.

Table 5: Attenuation from retrieval-proxy improvement to downstream PRIMARY improvement. Values are percentage-point deltas over the static anchor; the ratio is downstream divided by proxy. Backbones are DeepSeek-V3-0324 and Llama-3.3-70B-Instruct.

Domain	Backbone	Proxy Δ	LLM Δ	Ratio
Finance	DeepSeek	+1.49	+0.26	0.17
Weather	DeepSeek	+2.19	+0.82	0.37
Finance	Llama	+1.49	+0.68	0.46
Weather	Llama	+2.19	+1.75	0.80

5.7 Ranking-Order Effects

An adaptive weight preserves Top-3 membership but changes order for 5 Finance and 54 Weather queries. These are real prompt interventions because ICL is order sensitive (Lu et al., 2022; Qin et al., 2024). Finance order-only changes have positive net accuracy in the DeepSeek runs, whereas Weather varies across repetitions. Reconstruction therefore uses ordered identifiers rather than unordered overlap.

6 Conclusion

RC-QASH achieves the best point estimates in the DeepSeek baseline comparison and focuses adaptation on queries with weaker static-Hybrid performance. Changed-prompt comparisons show positive mean gains across the evaluated domains and backbones. The controller learns entirely from demonstration-bank reliability, while proxy-to-downstream attenuation leaves room for further improvement. Predicting prompt utility across domains without exhaustive downstream calls is the next target for multi-modal temporal prediction.

Reproducibility Statement

The method configuration, nested calibration, metric policy, response reconstruction, and paired statistics are documented in the text and appendices. Canonical external-test tables are rebuilt from frozen raw outputs. The artifact includes the Conda environment, zero-API rebuild, chronology, call ledger, and source registry. The development utility table is checked against a surviving aggregate record; its 2,040 responses must be restored to reparse outputs and recompute AUCs. Token and billing fields were not preserved, so monetary cost is not reported.

AI Use Statement

Generative AI tools assisted with method discussion and refinement, experimental design and methodology feedback, repository inspection, literature discovery, statistical-analysis code, artifact documentation, result interpretation, scientific figures, manuscript structure, LaTeX drafting, and consistency checks. External-test claims were checked against frozen artifacts, historical development utility against its retained aggregate, and literature metadata against primary sources. The authors reviewed the AI-assisted code and text and remain responsible for the final content, claims, and artifacts.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2204.14198.

Abdul Fatir Ansari, Lorenzo Stella, 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 Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024. URL https://arxiv.org/abs/2403.07815.

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

John Arevalo, Thamar Solorio, Manuel Montes-y Gomez, and Fabio A. Gonzalez. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017. URL https://arxiv.org/abs/1702.01992.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23719.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler,

Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.14165.

Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. MTBench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*, 2025. URL https://arxiv.org/abs/2503.16858.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI. DeepSeek-V3-0324 model card. Model release, March 2025, 2025. URL https://huggingface.co/deepseek-ai/DeepSeek-V3-0324.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, 1994. doi: 10.1201/9780429246593.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235. PMLR, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

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. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3eb7ca52e8207697361b2c0fb3926511-Abstract-Conference.html.

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. URL https://openreview.net/forum?id=Unb5CVPtae.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. In *Proceedings of ACL*, 2023. doi: 10.18653/v1/2023.acl-long.256. URL https://aclanthology.org/2023.acl-long.256/.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115. URL https://ieeexplore.ieee.org/document/61115/.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of the Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10/.

Qingxiang Liu, Zhiqing Cui, Xiaoliang Luo, Yuqian Wu, Zhuoyang Jiang, Huaiyu Wan, Sheng Sun, Lvchun Wang, Wei Yu, and Yuxuan Liang. Rationale-grounded in-context learning for time series reasoning with multimodal large language models. *arXiv preprint arXiv:2601.02968*, 2026. URL https://arxiv.org/abs/2601.02968.

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*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=KMCJXjlDDr.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of ACL*, 2022. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Meta. Llama-3.3-70B-Instruct model card. Model release, December 6, 2024, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of EMNLP*, 2022. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

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. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011. URL https://jmlr.org/papers/v12/pedregosa11a.html.

Kiran Purohit, Venktesh V, Sourangshu Bhattacharya, and Avishek Anand. Sample efficient demonstration selection for in-context learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49959–49982. PMLR, 2025. URL https://proceedings.mlr.press/v267/purohit25a.html.

Libo Qin, Qiguang Chen, Hao Fei, Zhi Chen, Min Li, and Wanxiang Che. What factors affect multi-modal in-context learning? an in-depth exploration. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/deeb4d6bdb5860fd7faf321dd5486d25-Abstract-Conference.html.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of NAACL-HLT*, 2022. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191/.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. TEST: Text prototype aligned embedding to activate LLM's ability for time series. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=Tuh4nZVb0g.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. doi: 10.1109/CVPR52733.2024.01365.

Huazheng Wang, Jinming Wu, Haifeng Sun, Zixuan Xia, Daixuan Cheng, Jingyu Wang, Qi Qi, and Jianxin Liao. MDR: Model-specific demonstration retrieval at inference time for in-context learning. In *Proceedings of NAACL-HLT*, 2024a. doi: 10.18653/v1/2024.naacl-long.235. URL https://aclanthology.org/2024.naacl-long.235/.

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*, 2024b. URL https://openreview.net/forum?id=7oLshfEIC2.

Song Wang, Zihan Chen, Chengshuai Shi, Cong Shen, and Jundong Li. Mixture of demonstrations for in-context learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024c. doi: 10.52202/079017-2796. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a0da098e0031f58269efdcba40eedf47-Abstract-Conference.html.

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. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Hangling Xie. MAF: Multimodal adaptive few-shot prompting for sentiment analysis with MLLMs. *arXiv preprint arXiv:2606.15694*, 2026. URL https://arxiv.org/abs/2606.15694.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Nan Xu, Fei Wang, Sheng Zhang, Hoifung Poon, and Muhao Chen. From introspection to best practices: Principled analysis of demonstrations in multimodal in-context learning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3299–3324. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.170. URL https://aclanthology.org/2025.naacl-long.170/.

Hao Xue and Flora D. Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023. doi: 10.1109/TKDE.2023.3342137.

Ellen Yi-Ge, Jiechao Gao, Wei Han, and Wei Zhu. DRUM: Learning demonstration retriever for large MUlti-modal models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 1051–1063. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-srw.83. URL https://aclanthology.org/2025.acl-srw.83/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025a. URL https://arxiv.org/abs/2506.05176.

Zheng Zhang, Shaocheng Lan, Lei Song, Jiang Bian, Yexin Li, and Kan Ren. Learning to select in-context demonstration preferred by large language model. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11345–11360. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.findings-acl.592. URL https://aclanthology.org/2025.findings-acl.592/.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. MMICL: Empowering vision-language model with multi-modal in-context learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5KojubHBr8.

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. doi: 10.52202/075280-1877.

A HYPERPARAMETER CONFIGURATION

Table 6 lists the numerical configuration shared by the external evaluations. The retrieval and routing symbols are defined in Sections 3–4; T_{decode} denotes the LLM sampling temperature.

Table 6: Functional names and fixed evaluation values for retrieval, reliability representation, routing, and decoding.

Symbol	Function	Value
α_{ts}	Temporal-Oriented Text Weight	0.10
α_{anchor}	Static-Anchor Text Weight	0.25
α_{text}	Text-Oriented Text Weight	0.40
K_{demo}	Prompt Demonstration Count	3
K_{stat}	Distribution Neighborhood Size	3
K_{pool}	Retained Ranking Depth	10
d_{ϕ}	Reliability-Feature Dimension	36
τ_{Finance}	Finance Routing Threshold	0.80
τ_{Weather}	Weather Routing Threshold	0.40
T_{decode}	LLM Decoding Temperature	0.70

A.1 RETRIEVAL REPRESENTATIONS AND GATE FITTING

The retained embedding preparation records identify the hosted model `qwen3-embedding-0.6b` and 1024 output dimensions. Text similarities are dot products of row-normalized embeddings. Weather represents each series by its 168 observed hourly values. Finance filters non-finite and non-positive prices, computes the cumulative log-return path relative to its first retained price, and linearly interpolates that path to 64 uniformly spaced points on a normalized time interval. It appends six statistics: total log return, sample standard deviation of successive log returns, the fitted path slope, signed maximum drawdown, final-step log return, and path range. The resulting Finance representation has 70 dimensions. Signed maximum drawdown is the minimum relative price change from the running maximum, so it is non-positive. Temporal coordinates are standardized using the current retrieval bank; bank and query vectors are then row-normalized before their dot products are computed. Finance clips the resulting cosine scores to the interval from -1 to 1.

The gate uses `StandardScaler` followed by class-balanced logistic regression. Its inverse regularization strength is set by `C=1.0`, and `max_iter=3000` limits optimization. The released environment pins scikit-learn 1.5.0; the unspecified classifier options use that version's defaults: L2 regularization, the L-BFGS solver, an intercept, and tolerance 10^{-4}. Finance and Weather use five outer retrieval folds and three and four inner retrieval folds, respectively. Their base seeds are 20260804 and 20260808. Thresholds range from 0.35 to 0.80 for Finance and from 0.40 to 0.80 for Weather, both in steps of 0.05. Appendix F distinguishes these retrieval folds from the cross-fitting used to select a threshold.

A.2 RELATIVE GAIN CALCULATION

For each main-table column, let S_{RC} denote the RC-QASH score and S_{base} the largest score among the other reported selectors. The relative gain G_{rel}, expressed as a percentage, is

$$G_{\mathrm{rel}} = 100\,\frac{S_{\mathrm{RC}} - S_{\mathrm{base}}}{S_{\mathrm{base}}}. \tag{12}$$

We calculate this presentation summary from the displayed two-decimal scores. The reference is BM25 Text Top-3 for both Finance metrics and Time-Series Top-3 for Weather and the equal-domain mean. A relative gain is distinct from the percentage-point difference between scores; neither changes the response-vector aggregation in Appendix B. Figure 2(c) retains the deltas reported in Table 2 rather than subtracting its rounded component scores.

B Baseline Repetition Details

Table 7 records the response-vector count behind each main-table cell. Finance BM25 and RC-QASH use three complete DeepSeek vectors. The other baseline cells use one complete vector. Table 8 reports the three Finance BM25 outcomes used in the main-table mean.

Table 7: Complete DeepSeek response vectors contributing to each main-table cell.

Method	Finance	Weather
Vanilla	1	1
Random Top-3	1	1
BM25 Text Top-3	3	1
Dense Text Top-3	1	1
Time-Series Top-3	1	1
RC-QASH	3	3

Table 8: Finance BM25 repetition results. The main table reports the arithmetic mean across the three complete response vectors.

Response Vector	Accuracy	PRIMARY
1	54.13	50.19
2	58.40	54.55
3	57.07	52.28
Mean	56.53	52.34
Standard Deviation	2.18	2.18

B.1 Response Groups Used by the Reported Comparisons

Finance uses the same three RC-QASH repetitions for headline and matched comparisons, with current-prompt anchor controls. Weather's two groups differ in repetition 1 and share repetitions 2–3. The headline group supplies Table 1, Figure 2(a–b), and Table 13; the matched group supplies Tables 5, 9, and 12, and Appendix G. Table 2(b) and Figure 2(c) use the shared repetition 3; Table 10 uses matched repetition 1. Controller settings and changed-query membership are identical across groups.

Each reconstructed vector reuses anchor responses on unchanged prompts. Metrics are computed per vector before averaging, so repetitions represent changed-prompt reruns conditional on those anchor responses.

C Changed-Prompt Effects Across Backbones

Table 9 summarizes the Table 2(b) changed-prompt comparison across backbones, using the matched groups in Appendix B.1.

Table 9: Hybrid–RC-QASH comparison on queries whose ordered Top-3 prompt changes. Values are percentages and deltas are percentage points. DeepSeek is the mean of three response vectors; Llama is one response vector.

			Accuracy			PRIMARY		
Domain	Backbone	Changed	Hybrid	RC-QASH	Δ	Hybrid	RC-QASH	Δ
Finance	DeepSeek	43	51.16	52.71	+1.55	46.38	47.96	+1.58
Weather	DeepSeek	422	49.53	52.13	+2.61	35.32	37.04	+1.72
Finance	Llama	43	41.86	55.81	+13.95	30.86	40.39	+9.53
Weather	Llama	422	33.41	39.81	+6.40	31.31	35.11	+3.80

D BANK-SELECTED STATIC RETRIEVAL DIAGNOSTIC

This diagnostic chooses one mixture per domain from demonstration-bank out-of-fold predictions: α_{ts} for Finance and α_{text} for Weather. It differs from the α_{anchor} Hybrid reference in Table 2. Table 10 uses one complete DeepSeek vector and the original matched RC-QASH repetition 1, not the headline mean. Its deltas are Static minus RC-QASH, the reverse of the controller-gain tables. The Weather static point estimate is higher, whereas the Finance direction favors RC-QASH; both intervals cross zero. This domain-dependent diagnostic is separate from the baseline ladder in Table 1.

Table 10: Bank-selected static hybrid under the repaired prompt. Deltas compare one complete response vector with original matched RC-QASH repetition 1 (Static minus RC-QASH); intervals use 20,000 gold-class-stratified query-bootstrap draws.

Domain	Static Weight	Accuracy	PRIMARY	ΔPRIMARY [95% CI]
Finance	α_{ts}	56.00	52.09	$-3.55\ [-8.35, 1.19]$
Weather	α_{text}	55.61	42.67	$+2.06\ [-1.51, 5.69]$

E COMPLETE FEATURE INVENTORY

The d_ϕ features share the same semantics in both domains; only class names differ. The three view-confidence blocks each contain top-1 similarity, top3-versus-tail gap, top-1 margin, purity, normalized entropy, and top-3 class diversity. Nine distribution features retain the three class probabilities for text, time series, and Hybrid. Four contrasts measure text-minus-time-series purity, entropy, and gap, plus text/time-series Jensen–Shannon divergence. Five agreement features measure text/time-series top-3 and top-10 overlap, text/Hybrid and time-series/Hybrid top-3 overlap, and predicted-class agreement. The four feature groups therefore contain 18, 9, 4, and 5 coordinates.

F NESTED CALIBRATION ALGORITHM

1. Split the demonstration bank into five outer folds (grouped for Finance). For each outer fold, build query features and proxy targets using only its training partition as the retrieval bank.
2. Within the outer training partition, generate inner out-of-fold retrieval records. Cross-fit the gate on these records using action-stratified folds, obtaining held-out action probabilities. Select τ_d by the resulting proxy PRIMARY, breaking exact ties toward the larger threshold.
3. Fit the standardized class-balanced logistic gate to inner records and predict actions for the untouched outer validation fold.
4. Concatenate outer-fold predictions to measure internal performance. Generate full-bank out-of-fold records, cross-fit their gate probabilities, select the final threshold, and fit the gate used on the external set.
5. On external queries, compute features without labels, select an action, and issue a fresh LLM call only when the ordered Top-3 differs from the anchor.

The gate-probability cross-fitting uses at most three folds for Finance and four for Weather, capped by the smallest action-class count. These folds are distinct from the inner retrieval splits. Let s_d denote the domain's base seed and f the outer-fold index, starting at one. Inner retrieval uses $s_d + 100f$ in both domains; probability cross-fitting uses $s_d + 1000 + f$ in Finance and $s_d + 1000f$ in Weather. Final Finance retrieval records reuse its five outer splits, with probability cross-fitting seeded by $s_d + 9999$. Weather instead generates five full-bank retrieval folds with seed $s_d + 9000$ and cross-fits their probabilities with $s_d + 10000$.

G PER-REPETITION MATCHED TRANSITIONS

Table 11: DeepSeek correctness transitions from the static anchor to RC-QASH in the primary three-class views. $W \to R$ and $R \to W$ denote wrong-to-right and right-to-wrong transitions.

Domain	Rep.	Prediction Changes	$W \to R$	$R \to W$	ΔAccuracy	ΔPRIMARY
Finance	1	11	4	4	+0.00	+0.05
Finance	2	11	5	5	+0.00	+0.19
Finance	3	11	6	4	+0.53	+0.53
Weather	1	149	56	54	+0.20	-0.28
Weather	2	152	59	48	+1.12	+1.35
Weather	3	150	71	51	+2.04	+1.39

H QUERY-LEVEL UNCERTAINTY AND PAIRED EVIDENCE

We use 20,000 gold-class-stratified query-bootstrap samples for the Vanilla–RC-QASH and static-anchor–RC-QASH comparisons. Each draw is shared across repetitions, and replicate deltas are averaged within the draw (Efron & Tibshirani, 1994). Exact two-sided McNemar tests summarize wrong-to-right and right-to-wrong transitions (McNemar, 1947); p denotes the resulting test p-value. These calculations condition on the realized frozen response vectors. They measure query-resampling uncertainty rather than API-decoding or provider uncertainty, and the main text therefore organizes the empirical argument around point estimates and matched mechanisms.

H.1 CONTROLLER COMPARISON WITH THE STATIC ANCHOR

Table 12 reports complete-test matched deltas after unchanged anchor responses are spliced with freshly evaluated changed prompts. DeepSeek averages three matched response vectors; Llama has one. These intervals describe full-test metric changes conditional on the realized responses, rather than uncertainty on the changed-prompt subset alone. Their widths depend on paired prediction changes and the metric.

Table 12: Matched RC-QASH gains over the static Hybrid anchor. Intervals are gold-class-stratified query-bootstrap intervals; invalid outputs in either response vector count as incorrect.

Domain	Backbone	ΔAccuracy	95% CI	ΔPRIMARY	95% CI
Finance	DeepSeek	+0.18	$[-1.07, 1.42]$	+0.26	$[-0.59, 1.14]$
Weather	DeepSeek	+1.12	$[-0.71, 2.89]$	+0.82	$[-1.18, 2.80]$
Finance	Llama	+1.60	$[-0.53, 4.00]$	+0.68	$[-1.20, 2.56]$
Weather	Llama	+2.76	$[0.41, 5.00]$	+1.75	$[-0.73, 4.22]$

H.2 FULL RETRIEVAL PIPELINE COMPARED WITH VANILLA

Dense Text Top-3 produces 34 net correct transitions on Finance and 43 on Weather, although its Weather class-balanced score does not improve. For RC-QASH, Finance repetitions produce 51, 52, and 54 net correct transitions relative to Vanilla; Weather produces 84, 75, and 84. Table 13 provides the paired intervals and tests for these frozen response vectors.

Table 13: Paired retrieval gains over repaired Vanilla prompting with DeepSeek. Intervals use 20,000 gold-class-stratified query-bootstrap draws. The RC-QASH McNemar entries give the range across three repetitions; Dense Text Top-3 has one complete run.

Candidate	Domain	ΔAccuracy	95% CI	ΔPRIMARY	95% CI	McNemar p
Dense Text Top-3	Finance	+9.07	$[3.20, 14.93]$	+5.85	$[-0.53, 12.26]$	0.0031
Dense Text Top-3	Weather	+4.39	$[0.71, 7.96]$	-0.83	$[-4.99, 3.35]$	0.0220
RC-QASH	Finance	+13.96	$[8.27, 19.64]$	+10.25	$[3.89, 16.55]$	2.9×10^{-6}–9.8×10^{-6}
RC-QASH	Weather	+8.27	$[4.69, 11.84]$	+3.43	$[-0.72, 7.56]$	1.2×10^{-5}–1.3×10^{-4}

I PROMPT REPAIR CONTROLS

Table 14: Effect of repairing the task prompt before retrieval. Finance is a 750-item historical prompt-ablation diagnostic; Weather is the 980-item frozen external baseline. The two rows have different scopes and are not pooled.

Domain	Scope	N	Original Acc.	Repaired Acc.	ΔAcc.	Orig. PRIMARY	Repaired PRIMARY
Finance	Prompt Audit	750	34.27	43.73	+9.47	39.64	44.80
Weather	External Baseline	980	43.78	47.14	+3.37	35.24	38.87

The Finance full repair improves the diagnostic three-class PRIMARY score by 5.16 points and removes eight invalid combined-prompt outputs. Its sample pool contains the full 750-item prompt audit, so it is evidence about prompt design rather than a replacement for the canonical 375-query external Vanilla row. Weather repair improves PRIMARY by 3.63 points. Both repairs precede every retrieval comparison and are not counted as RC-QASH component effects.

J FINANCE FIVE-CLASS RESULTS AND PARSER AUDIT

All DeepSeek Finance and Weather records used in the canonical tables parse to valid labels. Llama Finance has 11 invalid static-anchor outputs; after sparse replacement, 9 remain in the complete RC-QASH vector. They are counted as incorrect and excluded from no class in the fixed-label macro averages. The Llama Weather repairs cover nine invalid anchor responses and one invalid RC-QASH response; the final Weather vectors have no invalid outputs. Full confusion matrices and fixed-label per-class precision, recall, and F1 are included in the anonymous artifact.

Table 15: Finance five-class audit with DeepSeek. Aggregate performance (left) and class recall (right) use the fixed five-class label list. RC-QASH is the mean of three canonical reconstructions; support is fixed across methods.

Method	Accuracy	BAcc.	Macro-F1	PRIMARY
Vanilla	41.07	31.87	25.89	28.88
RC-QASH	**53.60**	**34.85**	**29.77**	**32.31**

Class	Support	Vanilla	RC-QASH
$< -4\%$	8	12.50	12.50
-4% to -2%	18	50.00	44.44
-2% to $+2\%$	255	44.31	60.00
$+2\%$ to $+4\%$	72	38.89	**52.78**
$> +4\%$	22	13.64	4.55

K FINANCE VERSION CHRONOLOGY

The Finance v0, v0.1, and v0.2 internal calibration summaries were generated on 2026-08-09 at 00:29, 00:32, and 00:36 (UTC+8), each with `external_test_used=false` and zero API calls. Their out-of-fold PRIMARY changes were -0.27, -0.55, and -0.07 points. A ten-seed v0.2 sensitivity analysis finished at 00:40. The zero-API external proxy analysis followed at 01:16 and reported gains of +1.49 for v0 and +3.17 for v0.2. The formal v0 LLM repetitions then completed at 09:55, 10:08, and 10:22.

The authors selected v0 before examining external outcomes, but this decision was not written down contemporaneously. The supplement therefore includes a retrospective author attestation dated 2026-08-10 and separately identifies the filesystem-backed chronology. This record supports the stated execution order without presenting the attestation as preregistration. No result in the paper uses v0.1 or v0.2 as the reported method.

L MODEL CALLS AND ENVIRONMENT

Table 16: Reconstructable request attempts for the matched static-anchor/RC-QASH evidence. Repairs are successful parser-invalid reruns; request retries are transient API failures followed by another attempt.

Backbone	Anchor	Fresh RC-QASH	Total	Repairs	Retries
DeepSeek	1,356	1,467	2,823	1	0
Llama	1,364	468	1,832	11	1

The selected Vanilla vectors contain 375 Finance and 980 Weather responses, but their source runs lack complete attempt ledgers. Formal evaluators use temperature 0.7, a 180-second timeout, and at most eight attempts with capped backoff. Responses omit token, latency, and billing fields, so cost cannot be reconstructed.

The released Conda specification records Python 3.10.14 and all package versions. Inference hardware is unavailable from the hosted API and was not logged locally, so the artifact provides an executable environment snapshot without making a hardware attribution.

M EVALUATION PROMPT TEMPLATES

The following templates reproduce the instruction text used by the Finance and Weather RC-QASH evaluators, with line breaks adjusted for typesetting. Braced fields identify runtime substitutions, not additional model instructions. For each query, the renderer joins three labeled demonstration blocks in retrieval order, followed by the unlabeled query. Hybrid and RC-QASH use the same template and differ only in the ordered demonstrations.

M.1 FINANCE INSTRUCTION AND SERIALIZATION

The Finance evaluator uses prompt version `fixed_7d_1d_qwen_score_icl_v1`. The field `demonstration_blocks` is the blank-line-separated concatenation of three demonstration blocks. `format_sample_input(query)` expands to the input template below. The retrieval-description sentence mentioning Qwen cosine similarity is literal shared prompt text; the evaluated selector determines which demonstrations fill the template.

```
You are a financial prediction expert with knowledge of time-series analysis and financial
news interpretation.
Your task is to predict the stock trend using historical price data and accompanying news.
Forecast the stock-price trend within the NEXT 1-DAY OUTPUT WINDOW.
The evaluation target is the percentage change between the first and last prices of that
future 1-day output sequence:
r = 100 * (P_future_end - P_future_start) / P_future_start
where:
- P_future_start is the first stock price in the future 1-day output window;
- P_future_end is the last stock price in the future 1-day output window.
The observed 7-day historical price change is forecasting evidence.
It is NOT itself the target label.
Assign exactly one five-way label:
1. r < -4%:
   <-4%
2. -4% <= r < -2%:
   -2% ~ -4%
3. -2% <= r <= +2%:
   -2% ~ +2%
4. +2% < r <= +4%:
   +2% ~ +4%
5. r > +4%:
   >+4%
The string "-2% ~ -4%" is the repository's required label for the mathematical interval
from -4% to -2%. Use that exact spelling.
Below are labeled historical demonstrations selected from a separate databank using Qwen
text-embedding cosine similarity.
Use these demonstrations to understand the task and output format.
Do not copy a demonstration label merely because its news appears similar. Predict the
current query independently from its own price history and news.
{demonstration_blocks}
Current query
Input:
{format_sample_input(query)}
Briefly analyze the current query's time-series data and news to forecast the next 1-day
output window.
Your FINAL LINE must contain exactly one of:
^^^<-4%^^^
^^^-2% ~ -4%^^^
^^^-2% ~ +2%^^^
^^^+2% ~ +4%^^^
^^^>+4%^^^
The final marked label must appear exactly once.
Do not place any text after the final label line.
```

Input and demonstration blocks. The serializer pairs timestamps and observed prices in a Python list of tuples; `time_series_data` is that list, and `text` contains the news. Each demonstration carries its own bank label; the query has no supplied label.

```
Time Series Stock Price Data:
The data contains stock prices recorded at 1-hour intervals over the past 7 days,
from {timestamps[0]} to {timestamps[-1]}.
Historical data:
{time_series_data}
News Data:
The following news contains information relevant to the stock's company or sector.
{text}
```

```
Demonstration {number}
Input:
{format_sample_input(sample)}
Output:
^^^{label}^^^
```

M.2 Weather Instruction and Serialization

The Weather field `demonstration_text` joins three blocks in order. The helper `format_ts` serializes the 168 observed hourly values as a JSON list after rounding each value to three decimal places. The query and demonstrations use the same serializer; `text` contains the weather event report.

```
You are a weather forecasting AI.
Predict the temperature trend for the first complete
24-hour day immediately following the provided
7-day historical window.
The historical input contains exactly 168 hourly
temperature observations.
TARGET DEFINITION
Let past_mean be the mean of the final 24 hourly
temperatures in the historical input.
Let future_mean be the mean temperature of the
immediately following 24-hour day.
Define:
delta_temperature = future_mean - past_mean
Use exactly these rules:
- increasing:
  delta_temperature >= 0.5
- decreasing:
  delta_temperature <= -0.5
- stable:
  -0.5 < delta_temperature < 0.5
The trend within the historical seven-day window
is evidence for forecasting, not the target itself.
Below are three retrieved examples from the
databank. Their labels refer to their own future
24-hour periods.
==================================================
RETRIEVED DEMONSTRATIONS
==================================================
{demonstration_text}
==================================================
QUERY
==================================================
Historical temperature observations:
{format_ts(query_source['input_window'])}
Weather event report:
{query_source['text']}
Use both modalities and the retrieved examples
to forecast the next-day trend.
Briefly reason about the evidence.
Your FINAL LINE must be exactly one of:
^^^decreasing^^^
^^^stable^^^
^^^increasing^^^
Do not put anything after the final label.
```

Demonstration block.

```
Demonstration {number}
Historical temperature observations:
{format_ts(source['input_window'])}
Weather event report:
{source['text']}
Observed future trend:
^^^{label}^^^
```