TSLM-ICL: TOWARD EFFECTIVE IN-CONTEXT LEARNING FOR TIME SERIES LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series language models (TSLMs) combine temporal signals with natural-language information, enabling tasks such as perception, reasoning, and decision making. In-context learning (ICL) offers a training-free way to adapt frozen TSLMs through demonstrations, yet existing time-series ICL methods are largely task-specific and do not address the heterogeneous structure of multimodal time-series data. We introduce **TSLM-ICL**, a training-free, model-agnostic retrieval framework for multimodal demonstration selection. TSLM-ICL represents samples using token-level temporal and textual embeddings, compares variable-sized representations via optimal transport, and selects relevant demonstrations under a fixed context budget. Across five public multimodal time-series benchmarks and six frozen TSLMs, TSLM-ICL improves accuracy by up to 16% over zero-shot prompting and consistently outperforms random, text-only, TimeRAG, and TS-RAG retrieval, achieving the best average accuracy in 14 of 15 dataset–budget configurations.

1 INTRODUCTION

Time series language models (TSLMs) have emerged as a specialized class of multimodal large language models that combine temporal signals with natural-language information (Guan et al., 2026; Yu et al., 2026; Kong et al., 2025; Xie et al., 2025b). Recent work demonstrates their capabilities across both in-distribution (Kong et al., 2025; Guan et al., 2026) and out-of-distribution (Yu et al., 2026; Guan et al., 2026) tasks. However, adapting TSLMs to new tasks, especially beyond their training distribution, often requires costly training or fine-tuning. Training-free adaptation offers an alternative by leveraging pretrained knowledge without parameter updates. A prominent paradigm is *in-context learning* (ICL) (Brown et al., 2020), which conditions models on task instructions and demonstrations at inference time. While ICL has been widely studied in text-based LLMs and explored for task-specific time series applications (Ning et al., 2026; Yang et al., 2025b), mostly for forecasting, its broader use for TSLMs remains an open area of research.

Existing work on time-series in-context learning has primarily focused on task-specific applications, particularly forecasting. Recent methods retrieve relevant examples to improve forecasting (Ning et al., 2026; Yang et al., 2025b), while models such as Chronos-2, TiRex-2, TimesFM-3 incorporate related series or covariates through contextual conditioning (Ansari et al., 2025; Podest et al., 2026; Das et al., 2024). However, using retrieved demonstrations to adapt TSLMs to general multimodal time-series–language tasks remains largely unexplored. Existing retrieval methods developed for text (Liu et al., 2022) or fixed-length univariate time series (Baldassini et al., 2024) do not directly extend to this setting, where samples may contain variable-length, multivariate, and unaligned series alongside text. These heterogeneous inputs require methods that jointly represent and compare temporal and textual information to identify informative demonstrations.

Motivated by work on multimodal ICL (Qin et al., 2024), we ask: *How can we design an effective off-the-shelf retrieval framework for in-context learning with time-series language models?* We frame this question around three decisions: how to *represent* multimodal samples, how to *compare* them, and how to *select* complementary demonstrations. These motivate **TSLM-ICL**, a training-free framework for multimodal time-series ICL. First, we represent each sample using embeddings from a pretrained multivariate time-series foundation model and a pretrained language model. Second, we compare the resulting embedding sets using optimal transport (Villani et al., 2009; Peyré &

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.

Cuturi, 2019), which accommodates varying numbers of embeddings while preserving fine-grained information that may be lost by pooling. Finally, we select the k most relevant demonstrations within the available context budget.

Across five public multimodal time-series benchmarks covering diverse tasks, including concept perception, reasoning, question answering, and decision making (Chen et al., 2026; Yu et al., 2026; Cai et al., 2024; Guan et al., 2026; Kong et al., 2025), and six frozen TSLMs, TSLM-ICL consistently outperforms competing retrieval strategies in 14 of 15 dataset–budget settings, achieving improvements of up to 16% points over zero-shot prompting. We compare TSLM-ICL against zero-shot prompting, random selection, text-only retrieval, and two prominent time-series retrieval approaches adapted to our setup (Ning et al., 2026; Yang et al., 2025b): TimeRAG, which compares raw time series using dynamic time warping, and TS-RAG, which ranks demonstrations based on time-series encoder summary-token representation. Through further analysis, we examine the respective contributions of representation, transport-based comparison, and demonstration selection to the overall performance. Our contributions are as follows:

- We introduce **TSLM-ICL**, a training-free retrieval framework for multimodal time-series ICL that combines time-series and textual representations, encoded embedding optimal transport, and top-k demonstration selection.
- We systematically evaluate the representation, comparison, and selection choices underlying TSLM-ICL, showing that the proposed ICL method improves frozen TSLMs on a range of multimodal time-series tasks.
- We establish a broad evaluation protocol spanning five public benchmarks, multiple subtasks, and six frozen TSLMs, in which every method retrieves from the same leave-one-out sub-dataset pools under matched demonstration and context budgets, enabling controlled comparisons of multimodal time-series retrieval strategies.

2 Related Work

Time Series Language Models (TSLMs). Early TSLMs primarily extended forecasting models with LLM components, using textual information as an additional source of context for time-series prediction. Representative approaches, including TEMPO, Time-LLM, and UniTime (Jin et al., 2024; Cao et al., 2024; Liu et al., 2024b), demonstrated the potential of combining language models with temporal data, but remained largely focused on forecasting. More recent work has moved toward general-purpose time-series–language understanding and reasoning. ChatTS (Xie et al., 2025a) introduced a dedicated framework for encoding time series and jointly training them with text, evaluating the resulting model on a broader range of time-series analysis tasks. Time-MQA (Kong et al., 2025) similarly adapted a pretrained language model using LoRA (Hu et al., 2021) on mixed time-series and textual data to support multimodal time-series question answering and analysis. More recently, TimeOmni (Guan et al., 2026) combined supervised fine-tuning with human-guided data and reinforcement learning to improve the generalizability of multimodal time-series reasoning. In parallel, vision-language models (VLMs) have demonstrated strong performance on multimodal time-series tasks (Yu et al., 2026; Gwiazda et al., 2026), further expanding the range of architectures capable of processing temporal and textual information jointly. Despite progress toward general-purpose TSLMs (including VLM-based approaches), existing work focuses primarily on model training/architecture, while training-free adaptations such as ICL remain largely unexplored.

To evaluate these emerging capabilities, several benchmarks and evaluation setups have been introduced alongside ChatTS, Time-MQA, and TimeOmni (Kong et al., 2025; Xie et al., 2025a; Guan et al., 2026). Stand-alone benchmarks such as TimeSeriesExam and TSRBench (Cai et al., 2024; Yu et al., 2026) evaluate models on their understanding of fundamental time-series concepts, including noise, anomalies, and pattern recognition. CiK and Time-MMD (Williams et al., 2025; Liu et al., 2024a) provide multimodal time-series datasets primarily for forecasting evaluation, while MTBench (Chen et al., 2026) focuses on multimodal reasoning in financial and weather domains using news, indicators, and contextual descriptions. Despite this growing body of benchmarks covering diverse tasks, domains, and modalities, they do not systematically evaluate demonstration-based ICL and retrieval across diverse time-series–language tasks.

In-context learning (ICL) in both unimodal and multimodal settings (Luo et al., 2024; Liu et al., 2022; Baldassini et al., 2024; Doveh et al., 2025) spans several research directions, including demonstration representation, comparison, and selection (Luo et al., 2024). For demonstration representation, prior work has explored different ways of encoding multimodal data for effective retrieval. For example, RICES (Alayrac et al., 2022; Baldassini et al., 2024) employs CLIP (Radford et al., 2021) to encode visual information for retrieval. (Qin et al., 2024) investigate different comparison metrics, including cosine similarity and Euclidean distance. For demonstration selection, most existing approaches rely on top-k retrieval based on example similarity (Liu et al., 2022; Baldassini et al., 2024), while others explicitly promote diversity among the retrieved examples (Kapuriya et al., 2025; Ye et al., 2023; Peng et al., 2024). In the time series domain, however, few works have explored retrieval for ICL. TimeRAG (Yang et al., 2025b) uses Dynamic Time Warping (DTW) to compare time series, while TS-RAG (Ning et al., 2026) leverages embeddings from Chronos (Ansari et al., 2024), a pretrained time-series forecasting model. Nevertheless, both approaches are primarily designed for retrieval based forecasting and rely solely on time-series information. Moreover, they are inherently univariate and do not naturally support variable-length inputs without additional modifications, while also excluding textual information.

3 Background

3.1 Problem Formulation

Multimodal examples. We consider a pre-trained *time series language model* (TSLM) θ that consumes inputs mixing a time series modality with text. Each example input is a pair

$$x = (\tau, w), \qquad \tau = (\tau_1, \ldots, \tau_C), \quad \text{with } \tau_c \in \mathbb{R}^{L_c}, \qquad w \in \mathcal{V}^*. \tag{1}$$

where τ is a time series with C variates and variate c has its *own* length L_c, where variates need not be aligned or equal-length, and w is a token sequence over vocabulary $\mathcal{V}$. Depending on the task family, y is the prediction text; a reasoning output, a question answer, or a class label.

In-context learning with a TSLM. Given θ, an optional instruction I, an *ordered* context set of k demonstrations $S = \big((x_1, y_1), \ldots, (x_k, y_k)\big)$, and a query $x^* = (\tau^*, w^*)$, the model estimates

$$p_\theta\big(y^* \mid x^*, I, S\big) \;=\; \prod_{n=1}^{|y^*|} p_\theta\big(y_n^* \mid y_{<n}^*,\, x^*,\, I,\, S\big) \tag{2}$$

during autoregressive inference. Crucially, θ is frozen: adaptation to the target task comes entirely from the demonstrations in S, with no model learning — or supervised fine-tuning (SFT) — of the TSLM itself.

3.2 What Factors Affect Multimodal Time Series In-Context Learning?

Following Qin et al. (2024), we view multimodal time-series ICL retrieval as three linked decisions: **sample representation**, **sample comparison**, and **sample selection**. Given a query x^* from a subdataset of N samples and its leave-one-out demonstration pool $P = \{(x_i, y_i)\}_{i=1}^{N-1}$, a representation function E maps the query and each demonstration input into a common space, yielding structured representations $E(x^*)$ and $E(x_i)$ that may be each a single or set of representation vectors, or even raw sequences. A comparison function then quantifies demonstration relevance, either as a similarity $\mathrm{sim}\big(E(x^*), E(x_i)\big)$ or as a distance $\mathrm{dist}\big(E(x^*), E(x_i)\big)$, which enters the ranking as $\mathrm{sim} = -\mathrm{dist}$, and a selection strategy constructs the ordered demonstration set S by choosing k demonstrations according to these scores. In this work, these choices define the building blocks of our methodology, particularly in light of the heterogeneity inherent to multimodal time-series inputs.

4 Method

TSLM-ICL retrieves demonstrations through three stages. We first represent each multimodal sample as a set of time-series and text embeddings, compare query and demonstration sets using optimal transport, and select the k most relevant demonstrations.

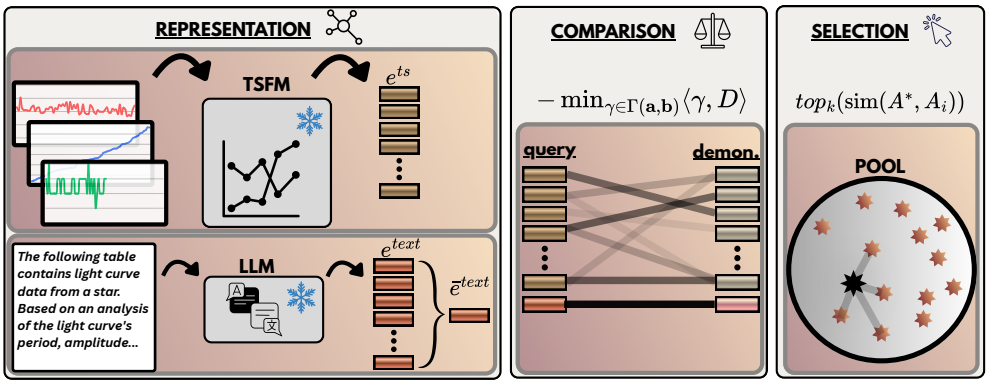


Figure 1: **TSLM-ICL method components.** *Representation:* a time-series foundation (TSFM) and LLM encode the multivariate time series and text, respectively into token embeddings. The time series embeddings are kept whole whereas the text embeddings are pooled into a single token. *Comparison:* two samples are scored by optimal transport between their token sets. *Selection:* the top-k pool samples are selected based on the OT similarity score.

4.1 Representation: Multivariate Time Series Foundation Models & LLM

To address the heterogeneous nature of the time series which spans multiple variates with potentially different lengths, we employ multivariate time series foundation models that can capture both intra- and inter-variate dependencies. To this end we employ Chronos-2 (Ansari et al., 2025), Chronos-2 is a pretrained time-series foundation model that performs zero-shot univariate, multivariate, and covariate-informed forecasting by using group attention to share information across related time series, while supporting long-context inputs of up to 8,192 time steps. After encoding τ, we keep the per-variate patch embeddings from the last encoder layer, together with the per-variate `[REG]` (summary) embeddings and the output (future) patch embeddings. The text w is encoded by the pre-trained sentence encoder MiniLM-L6 (Wang et al., 2020) into a single average pooled embedding, thus forming a multi modal set of embeddings $\{e_1^{ts}, \ldots, e_R^{ts}, \bar{e}^{text}\}$, where $e_r^{ts} \in \mathbb{R}^{d_{ts}}$ are the time-series token embeddings and $\bar{e}^{text} \in \mathbb{R}^{d_{text}}$ is the text embedding, the bar denoting mean pooling over the sentence encoder's token embeddings. However, the embeddings within a single set can be highly heterogeneous: patches from different variates, temporal regimes, and modalities may vary significantly in the embedding space. To this end, a suitable comparison metric between such sets is essential, which we address in the following sections.

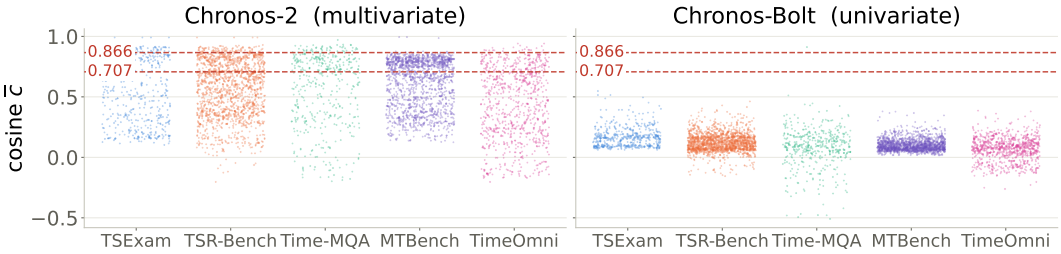


Figure 2: **Within-sample token embedding cohesion.** Each dot is one sample: the mean cosine $\bar{c}$ between the distinct tokens of its embedding set. Dashed lines mark $\cos 30° = 0.866$ and $\cos 45° = 0.707$.

4.2 Comparison: Optimal Transport

A common approach to compare two sets of token embeddings is to average each set and take the cosine similarity or Euclidean distance of the two resulting vectors. This discards part of each set's *energy*, namely the spread of its tokens around their mean. Formally, given a token embedding set $A = \{e_r^{ts}\}_{r=1}^R$ with pooled representation $\bar{e}^{ts} = \frac{1}{R}\sum_{r=1}^R e_r^{ts}$, the energy admits the standard bias–variance decomposition around the pooled representation:

$$\mathcal{E}(A) \;=\; \frac{1}{R}\sum_{r=1}^{R}\|e_r^{ts}\|_2^2 \;=\; \underbrace{\|\bar{e}^{ts}\|_2^2}_{\text{bias}} \;+\; \underbrace{\sigma^2}_{\text{variance}}, \qquad \sigma^2 \;=\; \frac{1}{R}\sum_{r=1}^{R}\|e_r^{ts}-\bar{e}^{ts}\|_2^2. \tag{3}$$

Pooling followed by cosine similarity or Euclidean distance only ever compares the bias term $\bar{e}^{ts}$, discarding the variance term σ^2 responsible for the remaining energy. More precisely, we can quantify this term exactly with the following standard identity. Assume the token embeddings $\{e_r^{ts}\}_{r=1}^{R}$ are ℓ_2-normalized, so that $\mathcal{E}(A) = 1$ and σ^2 is exactly the fraction of the set's energy left unexplained by its mean; let $\bar{c} = \frac{1}{R(R-1)}\sum_{r\neq s}\langle e_r^{ts}, e_s^{ts}\rangle$ denote the mean pairwise cosine similarity between distinct token embeddings.

Remark 1 (Dispersion of normalized token embedding sets)**.** $\sigma^2 = \frac{R-1}{R}\,(1-\bar{c})$ *(derivation in Appendix C.1).*

Hence the dispersion is fixed by how aligned a set's tokens are: σ^2 vanishes only when all tokens coincide ($\bar{c} = 1$). Empirically, 95% of the Chronos-2 token embedding sets we consider have $\bar{c} < \cos 30^\circ = \frac{\sqrt{3}}{2}$ (Figure 2), and for each of them

$$\sigma^2 \;>\; 0.1340 \cdot \frac{R-1}{R} \;\xrightarrow{R\to\infty}\; 0.1340. \tag{4}$$

With $\cos 45^\circ$, the floor rises to $0.2929 \cdot \frac{R-1}{R}$, which still holds for 64% of the token embedding sets.

Thus, unless a set's tokens are already near-collinear ($\bar{c}$ close to 1), a fixed fraction of its energy is dispersion that the pooled representation $\bar{e}^{ts}$ cannot capture. Recovering this energy requires comparing tokens directly rather than through their mean, which motivates a token-by-token, or vector-by-vector, comparison between the query and demonstration token embedding sets — precisely the setting *optimal transport* is designed for.

Optimal transport (OT) measures the minimum cost of transforming one distribution into another by optimally matching their mass. In our setting, each distribution is represented by a set of embeddings, the query $A^* = \{e_r^*\}_{r=1}^{R}$ and the demonstration $B = \{e_s\}_{s=1}^{R'}$. Here we assign each embedding an equal share of the total mass, resulting in uniform distributions (masses) being compared. These uniform masses are referred to as the *marginals* in the OT formulation. Therefore, the marginals are set to $\mathbf{a} = \frac{1}{R}\mathbf{1}$ and $\mathbf{b} = \frac{1}{R'}\mathbf{1}$, where $\mathbf{1}$ is a vector of ones. A *coupling* (or transport plan) $\gamma \in \mathbb{R}^{R\times R'}$ specifies how the mass of each query embedding is distributed across the demonstration embeddings, while preserving the marginals. The cost of matching query embedding e_r^* to demonstration embedding e_s is given by the *ground cost* $D_{rs} = \|e_r^* - e_s\|_2$. OT then finds the coupling that minimizes the total matching cost:

$$\Gamma(\mathbf{a},\mathbf{b}) = \{\gamma \in \mathbb{R}_{\geq 0}^{R\times R'} : \gamma\mathbf{1} = \mathbf{a},\, \gamma^\top\mathbf{1} = \mathbf{b}\}, \quad \mathrm{sim}_{\mathrm{OT}}(A^*, B) = -\min_{\gamma\in\Gamma}\langle\gamma, D\rangle. \tag{5}$$

The optimal plan γ provides a soft token-level embedding correspondence: a query embedding may match whichever demonstration embeddings are nearest in the encoded space. This makes OT suitable for Eq. equation 1: it accommodates unaligned, unequal-length variates, is defined when $R \neq R'$, and does not require a fixed-dimensional sample representation. Moreover, transport does more than accommodate variable-size inputs: it compares the full embedding distributions rather than only their means. Two standard properties of the resulting 1-Wasserstein distance make this precise (Villani et al., 2009; Peyré & Cuturi, 2019): it is a metric that both lower-bounds the mean-pooled Euclidean distance and distinguishes embedding sets with unequal dispersion:

Remark 2 (OT retains token-level embedding geometry, including information about dispersion (derivation in Appendix C.2))**.**

$$\min_{\gamma\in\Gamma(\mathbf{a},\mathbf{b})}\langle\gamma, D\rangle \;\geq\; \|\bar{e}^* - \bar{e}\|_2. \tag{6}$$

Moreover,

$$\bar{e}^* = \bar{e} \quad \textit{and} \quad \sigma_{A^*}^2 \neq \sigma_B^2 \qquad \Longrightarrow \qquad 0 < \min_{\gamma\in\Gamma(\mathbf{a},\mathbf{b})}\langle\gamma, D\rangle. \tag{7}$$

Eq. equation 6 follows from Kantorovich–Rubinstein duality (Villani et al., 2009). For Eq. equation 7, identical empirical distributions have identical means and variances, hence identical dispersion; therefore, $\sigma^2_{A^*} \neq \sigma^2_B$ implies that the two distributions differ, and W_1, being a metric, is strictly positive. Thus, even when pooling makes two samples indistinguishable, OT retains their token-level embedding geometry.

The constraints in $\Gamma(\mathbf{a}, \mathbf{b})$ conserve mass exactly: every query token is matched and every demonstration token receives its prescribed mass, hence the total mass is conserved for all token embedding sets, including those that may have been overlooked by pooling. In practice, we obtain the coupling with entropically regularized Sinkhorn iterations, an efficient matrix-scaling algorithm for approximately solving the transport problem; see Appendix B.3 for implementation details.

4.3 SELECTION: TOP-k

Given the similarity measure defined in Eq. equation 5, we rank demonstrations by $\mathrm{sim}(A^*, A_i)$, where A_i and A^* denote the token embedding sets of demonstration x_i and the query, respectively, and select the top-k demonstrations in descending order of relevance. We compare this independent ranking with random sampling and *maximal marginal relevance* (MMR) (Carbonell & Goldstein, 1998; Kapuriya et al., 2025; Ye et al., 2023), which additionally promotes diversity among selected demonstrations. As shown in Section 6, MMR performs comparably to top-k, while incurring $O(kN)$ additional computational overhead from iteratively evaluating candidates against the selected set.

4.4 TSLM-ICL

TSLM-ICL combines the three stages. *Representation:* each sample $x = (\tau, w)$ is mapped to a single token embedding set that carries both modalities,

$$A = \{(e^{ts}_r, \mathbf{0}_{d_{text}})\}_{r=1}^{R} \cup \{(\mathbf{0}_{d_{ts}}, \bar{e}^{text})\}, \quad \mathbf{a} = \left(\tfrac{1-\alpha}{R}, \ldots, \tfrac{1-\alpha}{R}, \alpha\right) \in \Delta^{R+1}, \tag{8}$$

where $\{e^{ts}_r\}_{r=1}^{R} \subset \mathbb{R}^{d_{ts}}$ are the time-series encoder tokens (patches) of all C variates encoded jointly, $\bar{e}^{text} \in \mathbb{R}^{d_{text}}$ is the pooled sentence embedding of w, and $\mathbf{0}_d$ is the zero vector in $\mathbb{R}^d$, so the time-series and text token embeddings occupy disjoint subspaces of $\mathbb{R}^{d_{ts}+d_{text}}$. All tokens are ℓ_2-normalized, $\alpha \in (0, 1)$ is the mass assigned to the text, and Δ^{R+1} is the probability simplex over the $R + 1$ points. Throughout, A^* denotes the token embedding set of the query x^* and A that of a demonstration x. *Comparison:* two samples are scored by the entropically regularized transport between their token embedding sets,

$$\mathrm{sim}\big(A^*, A\big) \;=\; -\big\langle \gamma_\varepsilon, D \big\rangle, \qquad \gamma_\varepsilon \;=\; \underset{\gamma \in \Gamma(\mathbf{a},\mathbf{b})}{\arg\min} \langle \gamma, D \rangle - \varepsilon H(\gamma), \tag{9}$$

with D the pairwise Euclidean cost of Eq. equation 5 and H the entropy of the plan, solved with Sinkhorn iterations. The mass α sets the relative weight of the two modalities: $\alpha \to 0$ recovers time-series-only transport, whereas $\alpha \to 1$ recovers text-only retrieval. *Selection:* the k demonstrations are chosen from the pool P by top-k relevance under this similarity (Eq. equation 9), and are placed in the prompt of Eq. equation 2 in increasing order of relevance, so the most similar demonstration sits immediately before the query. Proposition 1 (Appendix C.3) shows that the disjoint subspaces of Eq. equation 8 keep the optimal plan within each modality, so the mean bound of Remark 2 carries over to this non-uniform, multimodal setting.

5 EXPERIMENTS

Datasets. We evaluate on five public time-series multimodal benchmarks — TimeSeriesExam, TSRBench, MTBench, Time-MQA, and TimeOmni-1 (Cai et al., 2024; Yu et al., 2026; Chen et al., 2026; Kong et al., 2025; Guan et al., 2026). Each dataset has multiple subtasks. Every subtask of every benchmark is treated as its own *sub-dataset*, which is the unit of both retrieval and reporting: the demonstration pool for a query is **leave-one-out**, with respect to all other samples of the same sub-dataset. Appendix D reports additional details on the datasets such as datasets statistics, implementation specifics, and sub-dataset breakdowns.

Table 1: Methods and TSLMs.

Method	Multi-variate	Dist.	Diversi-fication	Text enc.	TS enc.	Train free
Random	×	—	× (uniform)	×	×	✓
TimeRAG	×	DTW	× (top-k)	×	×	✓
TS-RAG	(✓)	cosine	× (top-k)	×	✓	✓
Text	×	cosine	× (top-k)	✓	×	✓
Ours	✓	OT	× (top-k)	✓	✓	✓

Model	Params	TS-trained	Series input
Qwen3-4B	4B	×	Numeric text
Qwen3-VL-8B	8B	×	Rendered plot
ChatTS-8B	8B	✓	TS encoder
TimeMQA-7B	7B	✓	Numeric text
TimeOmni-1-7B	7B	✓	Numeric text
Gemma-3-12B	12B	×	Rendered plot

Models. We evaluate six TSLMs (Table 1, right) spanning different ways of providing time-series inputs. Three general-purpose LLMs have no time-series training: Qwen3-4B (Yang et al., 2025a), which reads series as numeric text, and Qwen3-VL-8B (Bai et al., 2025) and Gemma-3-12B (Team et al., 2025), which process rendered plots. The remaining three are adapted to the modality: ChatTS-8B (Xie et al., 2025a), which inserts patch embeddings into the text stream; Time-MQA (Kong et al., 2025), continually pre-trained on TSQA; and TimeOmni-1-7B (Guan et al., 2026), trained with SFT and RL for time-series reasoning. All models remain frozen and use identical decoding and context settings. Each has a 32K context window; when a prompt exceeds it, we retain the longest ranked prefix that fits, so the effective number of demonstrations may fall below k. All retrievers use the same budget; further details are provided in Appendices E.1 and E.

Retrieval methods. We compare TSLM-ICL with *Random* selection, *TimeRAG* (Yang et al., 2025b) (DTW on the first variate), *TS-RAG* (Ning et al., 2026) (cosine similarity of pooled time-series embeddings), and *Text* (cosine similarity of question embeddings), plus zero-shot ($k = 0$). TS-RAG pools embeddings across variates and uses its original Chronos-Bolt encoder; Appendix G.1 also evaluates it with Chronos-2. Every method retrieves from the same leave-one-out pool at the same k and context budget, isolating the retrieval objective; Table 1 summarizes the resulting capabilities.

5.1 Method Results

Table 2: **Accuracy by retrieval method and dataset, per demonstration budget.** Mean over six TSLMs and shared subtask cells. The *Average* weights every subtask equally; best values within each k block are in **bold**.

	Method	TSExam	TSR-Bench	Time-MQA	MTBench	TimeOmni	Average
	Zero-shot	0.476	0.356	0.431	0.421	0.244	0.376
$k=1$	Random	0.499	0.373	0.460	0.477	0.339	0.417
	Text	0.505	0.383	0.534	0.496	0.351	0.434
	TimeRAG	0.558	0.425	0.586	0.504	0.351	0.463
	TS-RAG	0.551	0.433	0.577	0.505	0.347	0.464
	TSLM-ICL	**0.608**	**0.466**	**0.588**	**0.511**	**0.355**	**0.488**
$k=4$	Random	0.521	0.388	0.525	0.491	0.407	0.445
	Text	0.522	0.416	0.582	0.516	0.424	0.469
	TimeRAG	0.593	0.453	0.661	0.529	0.432	0.503
	TS-RAG	0.593	0.474	0.665	0.528	0.444	0.513
	TSLM-ICL	**0.649**	**0.501**	**0.683**	**0.531**	**0.445**	**0.533**
$k=8$	Random	0.526	0.398	0.551	0.492	0.410	0.452
	Text	0.527	0.421	0.609	0.517	0.426	0.475
	TimeRAG	0.601	0.466	0.700	0.529	0.441	0.514
	TS-RAG	0.596	0.481	0.708	**0.531**	0.449	0.521
	TSLM-ICL	**0.649**	**0.505**	**0.728**	0.529	**0.453**	**0.539**

Main Results. Table 2 demonstrates the consistent effectiveness of TSLM-ICL across all five benchmarks and demonstration budgets. TSLM-ICL achieves the highest accuracy in 14 out of 15 dataset-by-budget comparisons, with the only exception being MTBench at $k = 8$, where it trails TS-RAG by only 0.002 accuracy points. In terms of average accuracy, TSLM-ICL improves over TimeRAG by 2.5, 3.0, and 2.5 percentage points for $k = 1$, $k = 4$, and $k = 8$, respectively, while the corresponding improvements over TS-RAG are 2.4, 2.0, and 1.8 percentage points. The gains are even more pronounced compared to retrieval based on random or text-only representations: TSLM-ICL improves over random selection by 7.1, 8.8, and 8.7 percentage points, and over text-based retrieval by 5.4, 6.4, and 6.4 percentage points for $k = 1$, $k = 4$, and $k = 8$, respectively. These

results emphasize the importance of incorporating temporal information into the retrieval process, as time-series representations consistently provide more informative demonstrations for downstream decision-making than text-only or randomly selected examples. Appendix G.1 shows that TSLM-ICL's advantage over TS-RAG persists when both use the same encoder.

Model Performance. Figure 3 aggregates average subtask accuracy by model and demonstration budget k. TSLM-ICL outperforms the competing retrieval methods for every model. Accuracy improves substantially from $k = 1$ to $k = 4$, but gains from $k = 4$ to $k = 8$ flatten for ChatTS and Time-MQA and reverse for TimeOmni. This pattern may reflect limitations in processing longer multimodal contexts, where additional demonstrations yield diminishing returns (Appendix E.1). Vision-based models are less affected, as plots avoid the token overhead of numeric series. Specifically, Qwen3-VL alone exceeds 0.55 average accuracy, reaching 0.57 and 0.59 at $k = 4$ and $k = 8$, respectively, highlighting the value of combining temporal retrieval with visual representations at larger budgets.

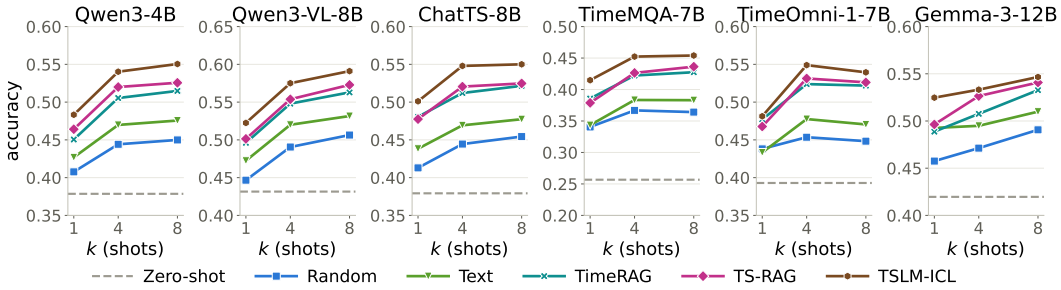


Figure 3: **Accuracy by demonstration budget and model.** Curves average shared benchmark-subtask cells and dashed lines show matching zero-shot accuracy.

6 Method Analysis

Component Analysis. We conduct leave-one-component-out ablations across six models, 39 accuracy-scored subtasks, and $k \in \{1, 4, 8\}$, comparing the full method with variants that remove text, remove time-series tokens, or replace token-level OT with mean-pooled cosine or Euclidean comparison. Table 3 reports each removal separately for subtasks with fewer than two versus at least two average variates, distinguishing predominantly univariate from multivariate inputs. Removing time-series tokens causes the largest degradation at every budget, establishing temporal geometry as the primary retrieval signal. Removing text produces a smaller average penalty: temporal similarity retrieves broadly relevant demonstrations, while text refines selection according to the query intent and accompanying context. The effect of the comparison function depends on the number of variates. On univariate series, replacing OT with pooling has relatively small effects (up to 0.003), however, in (Appendix G.3) we show that OT consistently outperforms pooling when the selected demonstrations differ. In the multivariate case, pooling degrades performance by up to 0.016 at $k = 4$, due to collapsing distinct channel-level structure. Thus, token-level temporal comparison is particularly important for heterogeneous multivariate inputs, yet effective for univariate as well.

Table 3: **What each removal costs, by variate count.** Change in mean accuracy from the full method, per budget k. Subtasks split by mean variates per record: < 2 (27 subtasks) and ≥ 2 (12). Negative implies degradation from the full method.

	Uni. (< 2)			Multi. (≥ 2)		
Removed	$k = 1$	$k = 4$	$k = 8$	$k = 1$	$k = 4$	$k = 8$
w/o text	$-.017$	$-.014$	$-.011$	$-.007$	$-.006$	$+.006$
text only	$-.044$	$-.057$	$-.056$	$-.075$	$-.080$	$-.082$
w/o OT (cos)	$-.002$	$-.002$	$-.001$	$-.011$	$-.016$	$-.014$
w/o OT (euc)	$-.000$	$-.003$	$-.002$	$-.008$	$-.016$	$-.013$

Top-k versus Diversity. Figure 4 varies only the selection strategy, holding the representation and transport comparison fixed. To discourage near-duplicate demonstrations, MMR builds S greedily from $S_0 = \emptyset$, appending $x_{t+1} = \arg\max_{x_i \in P \setminus S_t} \left[\mathrm{sim}(A^*, A_i) - \lambda \max_{x_j \in S_t} \mathrm{sim}(A_i, A_j) \right]$,

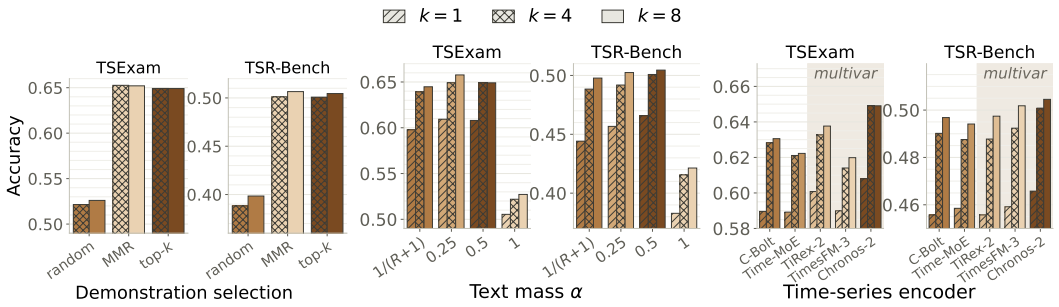


Figure 4: **Demonstration selection (left), text mass (middle) and time-series encoder (right).** Each panel moves one component of TSLM-ICL and holds the rest fixed. *Left:* random, MMR ($\lambda = 0.5$) and top-k (ours) selection, shown for $k \in \{4, 8\}$. *Middle:* Text mass α of the transport marginal; $\frac{1}{R+1}$ weighs the text point like one series token, and $\alpha = 1$ puts all mass on it, i.e. text-only retrieval. *Right:* encoder comparison of Chronos-Bolt, Time-MoE, TiRex-2, TimesFM-3, and Chronos-2 (ours).

where the first term measures query relevance and the second penalizes redundancy. We use $\lambda = 0.5$; $\lambda = 0$ recovers top-k. Random selection, which ignores relevance, trails top-k by 9.3–12.8 accuracy points across TSExam and TSR-Bench. In contrast, MMR changes mean accuracy by at most 0.3 points and shows no consistent improvement across the evaluated budgets. Moreover, MMR also requires $O(kN)$ additional candidate comparisons per query, TSLM-ICL uses plain top-k.

Ablation: Text Mass and Series Encoder. Figures 4 and 4 examine the effects of text mass and the series encoder, respectively. For text mass, text-only retrieval ($\alpha = 1$) performs worst, trailing $\alpha = 0.5$ by 11.7 points on TSExam and 8.4 on TSR-Bench, while $\alpha = 1/(R{+}1)$ underweights text and loses 0.8 and 1.4 points, respectively. Intermediate values are similar, with $\alpha = 0.25$ and $\alpha = 0.5$ differing by at most 0.7 points; TSLM-ICL fixes $\alpha = 0.5$, assigning equal mass to both modalities. For the series encoder, averaged over $k \in \{1, 4, 8\}$, multivariate encoders outperform univariate ones by 0.9 percentage points on TSExam and 0.4 on TSR-Bench. Chronos-2 performs best at every budget, exceeding the strongest alternative by 1.2 and 0.6 percentage points, respectively, suggesting that joint channel encoding is beneficial when it preserves temporal geometry relevant to transport.

7 Discussion

Our results show that effective training-free adaptation of TSLMs depends on how multimodal time-series representations are compared. Across six frozen models and five benchmarks, TSLM-ICL consistently improves over existing retrieval strategies, with temporal representations providing the primary retrieval signal and text providing complementary semantic information. The benefit of token-level optimal transport is particularly evident for multivariate inputs, where pooling can lose channel-level structure. Moreover, the comparison with TS-RAG using the same encoder shows that the gains of TSLM-ICL arise not only from the encoder, but also from comparing its token embeddings with optimal transport.

The results further suggest that representation and comparison matter more than the specific selection strategy once relevant demonstrations are identified. MMR does not consistently improve over top-k, while larger budgets provide diminishing returns for several models. Overall, these findings highlight structured relevance estimation as the central challenge in TSLM-ICL: retrieving demonstrations that match both the temporal structure and semantic context of the query.

AI USE STATEMENT

In this work, we used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have not used generative AI tools for other tasks with required disclosure, and Generate synthetic data sets are not applicable to this work. Additionally, we used generative AI tools for create or modify scientific figures or images, create or edit software code, creation of artifacts, draft parts of a research paper, transcribe recordings of research material, summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, format references, suggest a structure for a research paper. We have reviewed all AI-assisted work. Cross-checked several times, used other forms of AI for verification, reviewed by 3 people. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736. Curran Associates, Inc., 2022. doi: 10.52202/068431-1723. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

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. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Folco Bertini Baldassini, Mustafa Shukor, Matthieu Cord, Laure Soulier, and Benjamin Piwowarski. What makes multimodal in-context learning work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. Timeseriesexam: A time series understanding exam. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024. URL https://openreview.net/forum?id=oXRX7ABgLz.

Defu Cao, Furong Jia, Sercan Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 18546–18578, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5132940b1bced8a7b28e9695d49d435a-Paper-Conference.pdf.

Jaime G Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, volume 98, pp. 290941–291025, 1998.

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, 2026. URL https://arxiv.org/abs/2503.16858.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

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

Sivan Doveh, Shaked Perek, M. Jehanzeb Mirza, Wei Lin, Amit Alfassy, Assaf Arbelle, Shimon Ullman, and Leonid Karlinsky. Towards multimodal in-context learning for vision and language models. In Alessio Del Bue, Cristian Canton, Jordi Pont-Tuset, and Tatiana Tommasi (eds.), *Computer Vision – ECCV 2024 Workshops*, pp. 250–267, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-93806-1.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 152139–152170, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f7227d037ee7202106f511f78e9ecb80-Paper-Conference.pdf.

Malgorzata Gwiazda, Yifu Cai, Mononito Goswami, Arjun Choudhry, and Artur Dubrawski. Timeseriesexamagent: Creating time series reasoning benchmarks at scale. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DewXWSvQPH.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James 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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and

Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23857–23880, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/680b2a8135b9c71278a09cafb605869e-Paper-Conference.pdf.

Janak Kapuriya, Manit Kaushik, Debasis Ganguly, and Sumit Bhatia. Exploring the role of diversity in example selection for in-context learning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2962–2966, 2025.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-MQA: Time series multi-task question answering with context enhancement. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1437. URL https://aclanthology.org/2025.acl-long.1437/.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-MMD: Multi-domain multimodal dataset for time series analysis. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=fuD0h4R1IL.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114, 2022.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 4095–4106, New York, NY, USA, 2024b. Association for Computing Machinery.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*, 2024.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James 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. *Advances in Neural Information Processing Systems*, 38:163170–163199, 2026.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9090–9101, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.492. URL https://aclanthology.org/2024.acl-long.492/.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and trends® in machine learning*, 11(5-6):355–607, 2019.

Patrick Podest, Marco Pichler, Elias Bürger, Levente Zólyomi, Bernhard Voggenberger, Wilhelm Berghammer, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex-2: Generalizing tirex to multivariate data and streaming. *arXiv preprint arXiv:2607.01204*, 2026.

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. *Advances in Neural Information Processing Systems*, 37:123207–123236, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine*

Learning, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados,

and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ih2WuBT1Fn.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *Proc. VLDB Endow.*, 18(8):2385–2398, April 2025a. ISSN 2150-8097. doi: 10.14778/3742728.3742735. URL https://doi.org/10.14778/3742728.3742735.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. In *Proceedings of the VLDB Endowment, 2025*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. TimeRAG: Boosting LLM time series forecasting via retrieval-augmented generation. In *ICASSP 2025 — IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025b.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. Complementary explanations for effective in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4469–4484, 2023.

Fangxu Yu, Xingang Guo, Lingzhi Yuan, Haoqiang Kang, Hongyu Zhao, Lianhui Qin, Furong Huang, Bin Hu, and Tianyi Zhou. TSRBench: A comprehensive multi-task multi-modal time series reasoning benchmark for generalist models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=S9G3yu6vHl.

APPENDIX CONTENTS

A Notation **16**

B Implementation Details **18**

C Proofs **19**

D Datasets **23**

E Models **26**

F Retrieval Methods **28**

G Additional Experiments **31**

H Extended Results **34**

A NOTATION

Table 4 lists the notation used in the main text and the appendix, grouped by the stage of the method in which it first appears.

Table 4: **Notation.**

Symbol	Description
Problem formulation and in-context learning (Section 3.1)	
θ	Frozen time-series language model (TSLM)
$x = (\tau, w)$	Multimodal sample input: time series τ and text w
$\tau = (\tau_1, \dots, \tau_C)$	Multivariate time series with C variates
$\tau_c \in \mathbb{R}^{L_c}$	Variate c, of its own length L_c; variates need not be aligned
$w \in \mathcal{V}^*$	Text (question and context), a token sequence over vocabulary $\mathcal{V}$
y	Target output: answer, label or reasoning text
$x^* = (\tau^*, w^*),\ y^*$	Query input and its target
I	Optional task instruction
$P = \{(x_i, y_i)\}_{i=1}^{N-1}$	Leave-one-out demonstration pool: the other samples of the query's sub-dataset
N	Number of samples in a sub-dataset
S	Ordered set of k demonstrations placed in the prompt
k	Demonstration budget (number of demonstrations requested)
$E(\cdot)$	Representation function mapping a sample to its representation
$\mathrm{sim},\ \mathrm{dist}$	Similarity used to rank demonstrations; a distance enters as $\mathrm{sim} = -\mathrm{dist}$
Representation (Sections 4.1 and 4.4)	
$e_r^{ts} \in \mathbb{R}^{d_{ts}}$	r-th token embedding (Chronos-2 patch, `[REG]` or future token); $e_r^{ts,*}$ for the query
$e_r,\ e_s$	Tokens of a generic embedding set (Section 4.2, Remark 2); e_r^* for the query
$R,\ R'$	Number of time-series tokens of the query and of a demonstration
$\bar{e}^{text} \in \mathbb{R}^{d_{text}}$	Sentence embedding of w (MiniLM); the bar denotes mean pooling
$d_{ts},\ d_{text}$	Embedding dimensions of the time-series (768) and text (384) encoders
$\mathbf{0}_d$	Zero vector in $\mathbb{R}^d$, used to place the two modalities in disjoint subspaces
$A^*,\ A$	Token embedding sets of the query and of a demonstration (Eq. equation 8)
$\bar{e}^{ts},\ \bar{e}^{ts,*}$	Mean-pooled time-series token embedding of a demonstration and of the query; $\bar{e}$, $\bar{e}^*$ for a generic set
$\tilde{e}_s,\ \tilde{e}_r^*$	Zero-padded tokens in $\mathbb{R}^{d_{ts}+d_{text}}$ (Appendix C.3)
$\mathcal{E}(A)$	Energy of a set: mean squared norm of its tokens (Eq. equation 3)
σ^2	Dispersion of a set around its mean
$\bar{c}$	Mean pairwise cosine similarity between the distinct tokens of a set
Comparison: optimal transport (Section 4.2, Appendix B.3)	
$\mathbf{a} \in \Delta^{R+1},\ \mathbf{b}$	Marginals (token masses) of the query and demonstration sets
Δ^{R+1}	Probability simplex over $R+1$ points
α	Text mass: marginal mass assigned to the text token ($\alpha = 0.5$)
D	Ground-cost matrix, $D_{rs} = \|e_r^* - e_s\|_2$
$\gamma,\ \Gamma(\mathbf{a}, \mathbf{b})$	Transport plan (coupling) and the set of plans with marginals $\mathbf{a}, \mathbf{b}$
W_1	Optimal (exact) transport cost, the 1-Wasserstein distance
$\mu_{A^*},\ \mu_B$	Uniform empirical distributions of two token embedding sets (Appendix C.2)
ρ_{text}	Distance between the text tokens of the query and a demonstration (Appendix C.3)
ε	Entropic (Sinkhorn) regularization ($\varepsilon = 0.03$)
$\gamma_\varepsilon,\ H(\gamma)$	Entropic optimal plan and the entropy of a plan
$K,\ \mathbf{u},\ \mathbf{v}$	Gibbs kernel $e^{-D/\varepsilon}$ and the Sinkhorn scaling vectors
Q	Number of pairs solved together in one Sinkhorn batch (Algorithm 1)
Selection (Sections 4.3 and 6)	
λ	MMR redundancy weight ($\lambda = 0$ recovers top-k)
S_t	Demonstrations selected after t greedy MMR steps
Baselines (Appendix F)	
$f_{\text{text}}(w)$	Pooled MiniLM sentence embedding (Text baseline)
$\pi,\ M$	DTW warping path $\big((p_1, q_1), \dots, (p_M, q_M)\big)$ and its length (TimeRAG)
$z(\cdot)$	Per-variate z-scoring

Continued on next page

Table 4, continued

Symbol	Description
$h^{\mathrm{REG}}(\cdot),\ \bar{h}(x)$	Chronos-Bolt summary token of one variate, and its mean over variates (TS-RAG)
Evaluation (Appendices E.1 and H)	
β	Prompt token budget (28,608 tokens)
m	Effective number of demonstrations that fit within β ($m \leq k$)
$J,\ n_j$	Number of common cells of a table entry, and the number of items in cell j
$\Delta,\ \mathrm{SE}(\Delta)$	Paired accuracy difference and its standard error
$S_{\mathrm{OT}},\ S_{\mathrm{pool}}$	Demonstrations retrieved with OT and with mean pooling (Appendix G.3)
$\Delta\mathrm{acc}$	Per-query accuracy of TSLM-ICL minus that of a pooled variant (Appendix G.3)

B Implementation Details

B.1 Shared Hyperparameters

Table 5 lists the settings shared by every run. Inference and ICL settings are the same for all retrieval methods, so methods differ only in which demonstrations they select. The retrieval block gives TSLM-ICL's configuration.

Table 5: **Shared hyperparameters.** Inference and ICL settings apply to every method and model; retrieval settings are TSLM-ICL's.

	Setting	Value
Inference	Engine / precision	vLLM, bfloat16 (ChatTS: float16)
	Hardware	1× NVIDIA RTX 6000 (48 GB) per job
	Context window	32,768 tokens
	Decoding	greedy (temperature 0, top-p 1), seed 0
	Max. new tokens	4,096
	Prompt budget β	$32{,}768 - 4{,}096 - 64 = 28{,}608$ tokens
	Reasoning mode	Qwen3 thinking off; TimeOmni: its `<think>/<answer>` prompt
ICL	Demonstrations k	$\{1, 4, 8\}$ (zero-shot: $k = 0$)
	Pool	leave-one-out within the sub-dataset
	Ordering	increasing relevance (most similar next to the query)
	Overflow	drop least relevant demonstrations (App. E.1)
Retrieval (TSLM-ICL)	Series encoder	Chronos-2 (`amazon/chronos-2`), float32, $d_{ts} = 768$
	Series tokens	all patch tokens + `[REG]` + future token, per variate
	Variate grouping	same-length variates encoded jointly (group attention)
	Text encoder	`all-MiniLM-L6-v2`, pooled, $d_{text} = 384$
	Normalization	per-variate z-score of the input; ℓ_2 per token
	Text mass α	0.5
	Ground cost	Euclidean
	Sinkhorn ε	0.03
	Stopping	$\|\gamma^\top \mathbf{1} - \mathbf{b}\|_2 < 10^{-6}$ (checked every 10 iter.), $\leq$ 1,000 iter.

B.2 Retrieval Pipeline

Retrieval is fully decoupled from generation. For each sub-dataset of N samples, a precompute job encodes every sample once and fills the symmetric $N \times N$ similarity matrix, which is cached on disk and keyed by every parameter that changes it. At evaluation time the leave-one-out pool of query x_i is row i with the diagonal removed, so top-k selection is a sort of that row. Two consequences follow. First, all six TSLMs read the same demonstrations for a given method, sub-dataset and k, so differences between models are not caused by retrieval. Second, the encoders never share a GPU with the vLLM engine.

Token embedding set construction. Each variate is z-scored and all variates of a sample that share a length are passed to Chronos-2 as one multivariate item, so group attention mixes them. Ragged variates, which occur in 34 samples corpus-wide, are encoded in separate groups. Samples are encoded one at a time, so no padding patches enter a token embedding set. A variate of length L contributes $\lceil L/16 \rceil$ patch tokens plus its `[REG]` and future tokens. The sample's question and context text are embedded into one pooled MiniLM vector. After ℓ_2 normalization the two blocks are zero-padded into $\mathbb{R}^{d_{ts}+d_{text}}$ as in Eq. equation 8. Unit-norm points in disjoint subspaces lie exactly $\sqrt{2}$ apart, so every entry of D is at most 2.

B.3 Entropic Optimal Transport and Batched Sinkhorn

Entropic OT. For marginals $\mathbf{a} \in \Delta^R$, $\mathbf{b} \in \Delta^{R'}$ and cost D, the entropic problem of Eq. equation 9, $\min_{\gamma \in \Gamma(\mathbf{a},\mathbf{b})} \langle \gamma, D \rangle - \varepsilon H(\gamma)$ with $H(\gamma) = -\sum_{rs} \gamma_{rs}(\log \gamma_{rs} - 1)$, has a unique solution of the form $\gamma_\varepsilon = \mathrm{diag}(\mathbf{u})\, K\, \mathrm{diag}(\mathbf{v})$ with Gibbs kernel $K = e^{-D/\varepsilon}$ (Cuturi, 2013). The scalings are found by alternately matching the column and row marginals, $\mathbf{v} \leftarrow \mathbf{b} \oslash K^\top \mathbf{u}$ and $\mathbf{u} \leftarrow \mathbf{a} \oslash K\mathbf{v}$.

We score a pair by the transport cost of the resulting plan, $\langle \gamma_\varepsilon, D \rangle$, without the entropy term, and set $\mathrm{sim} = -\langle \gamma_\varepsilon, D \rangle$.

Batching. A sub-dataset requires $N(N-1)/2$ solves, up to ∼250k on the largest TSR-Bench sub-datasets. For token embedding sets of a few hundred tokens, one Sinkhorn iteration costs less than the GPU kernel launches around it, so we solve many pairs per launch (Algorithm 1). Pairs in a batch are zero-padded to the batch's longest source and target token embedding sets. Padded tokens receive zero mass and a zeroed kernel entry, so they neither emit nor receive mass, and each pair converges to the same fixed point as when solved alone. The iteration, initialization and stopping rule match POT's `sinkhorn_knopp` (Flamary et al., 2021), so the batched and per-pair solvers agree up to floating-point reduction order. The batch size is set from a 512 MiB budget for the cost and kernel tensors, $Q = \min\big(4096, \lfloor 2^{29}/(8\,R_{\max}^2)\rfloor\big)$ with $R_{\max}$ the largest token embedding set in the sub-dataset, and halves on an out-of-memory error.

Algorithm 1: Batched Sinkhorn (one batch of Q pairs of token embedding sets)

Require: token embedding sets $\{(A_t^*, \mathbf{a}_t, A_t, \mathbf{b}_t)\}_{t=1}^{Q}$; regularization ε; tolerance $\epsilon_{\text{tol}} = 10^{-6}$; max. iterations $T = 1000$

$\hat{R}^* \leftarrow \max_t |A_t^*|, \quad \hat{R} \leftarrow \max_t |A_t|$

Pad each A_t^*, A_t with zero vectors and $\mathbf{a}_t, \mathbf{b}_t$ with zeros to lengths $\hat{R}^*, \hat{R}$

$\Omega_{t,rs} \leftarrow \mathbb{1}[a_{t,r} > 0]\,\mathbb{1}[b_{t,s} > 0]$ ▷ padding mask, $Q \times \hat{R}^* \times \hat{R}$

$D_{t,rs} \leftarrow \|A_{t,r}^* - A_{t,s}\|_2$ ▷ batched `cdist`

$K \leftarrow \exp(-D/\varepsilon) \odot \Omega$

$\mathbf{u} \leftarrow \mathbf{a}, \quad \mathbf{v} \leftarrow \mathbf{b}$

for $\iota = 1, \ldots, T$ **do**

  $\mathbf{v} \leftarrow \mathbf{b} \oslash \max(K^\top \mathbf{u}, \eta)$ ▷ batched matmul; η = smallest float32

  $\mathbf{u} \leftarrow \mathbf{a} \oslash \max(K\mathbf{v}, \eta)$

  if $\iota \bmod 10 = 0$ **and** $\max_t \|\mathbf{v}_t \odot K_t^\top \mathbf{u}_t - \mathbf{b}_t\|_2 < \epsilon_{\text{tol}}$ **then**

    break

return $\mathrm{OT}_t = \mathbf{u}_t^\top (K_t \odot D_t)\,\mathbf{v}_t$ for $t = 1, \ldots, Q$ ▷ $\langle \gamma_t, D_t \rangle$ without forming γ_t

Numerical range. The solver runs in the plain kernel domain in float32. Since $D_{rs} \le 2$, the smallest kernel entry at $\varepsilon = 0.03$ is $e^{-2/0.03} \approx 10^{-29}$, above float32's smallest normal value ($\approx 1.2 \times 10^{-38}$). This is why no log-domain solver is needed at the values we evaluate ($\varepsilon \in \{0.03, 0.05, 0.1\}$), and also why $\varepsilon = 0.01$ (e^{-200}) cannot be run with this solver.

B.4 Answer Extraction and Scoring

Model outputs are parsed deterministically. Any `<think>` block is stripped, the content of a final `<answer>` tag is preferred when present, and an ordered list of task-specific rules (option letter, closed label vocabulary, or numeric list) is applied; the first rule that fires determines the answer. There is no substring-containment fallback. An output that no rule parses is scored as incorrect. Multiple-choice and label tasks are scored by exact match against the gold answer, and TimeOmni uses its official scoring function. Accuracy is averaged over the samples of a sub-dataset. Error bars are ± 1 SE of the reported mean, propagated from the binomial grading error of each (model, sub-dataset, k) cell.

C Proofs

C.1 Derivation of Remark 1

This is a standard identity; we include the derivation for completeness.

Proof. Since all embeddings are ℓ_2-normalized, $\|e_r\|_2^2 = 1$ for every r, and therefore $\mathcal{E}(A) = 1$. Expanding the squared norm of the pooled representation gives

$$\|\bar{e}\|_2^2 = \left\| \frac{1}{R} \sum_{r=1}^{R} e_r \right\|_2^2 \tag{10}$$

$$= \frac{1}{R^2} \sum_{r=1}^{R} \sum_{s=1}^{R} \langle e_r, e_s \rangle \tag{11}$$

$$= \frac{1}{R^2} \left(\sum_{r=1}^{R} \|e_r\|_2^2 + \sum_{r \neq s} \langle e_r, e_s \rangle \right). \tag{12}$$

The first sum equals R by normalization, while the definition of the mean pairwise cosine similarity gives

$$\sum_{r \neq s} \langle e_r, e_s \rangle = R(R-1)\bar{c}.$$

Consequently,

$$\|\bar{e}\|_2^2 = \frac{R + R(R-1)\bar{c}}{R^2} = \frac{1 + (R-1)\bar{c}}{R}. \tag{13}$$

Using the bias–variance decomposition from equation 3 and $\mathcal{E}(A) = 1$, the dispersion term satisfies

$$\sigma^2 = 1 - \|\bar{e}\|_2^2 \tag{14}$$

$$= 1 - \frac{1 + (R-1)\bar{c}}{R} \tag{15}$$

$$= \frac{R-1}{R}(1 - \bar{c}). \tag{16}$$

□

C.2 Derivation of Remark 2

Both statements are standard (Villani et al., 2009; Peyré & Cuturi, 2019); we include the short argument for completeness.

Proof. Since $\gamma \in \Gamma(\mathbf{a}, \mathbf{b})$ has marginals $a_r = 1/R$ and $b_s = 1/R'$, we have

$$\sum_{r=1}^{R} \sum_{s=1}^{R'} \gamma_{rs}(e_r^* - e_s) = \sum_{r=1}^{R} e_r^* \underbrace{\sum_{s=1}^{R'} \gamma_{rs}}_{= a_r = 1/R} - \sum_{s=1}^{R'} e_s \underbrace{\sum_{r=1}^{R} \gamma_{rs}}_{= b_s = 1/R'} \tag{17}$$

$$= \frac{1}{R} \sum_{r=1}^{R} e_r^* - \frac{1}{R'} \sum_{s=1}^{R'} e_s = \bar{e}^* - \bar{e}. \tag{18}$$

Therefore, by the triangle inequality and the non-negativity of γ_{rs},

$$\|\bar{e}^* - \bar{e}\|_2 = \left\| \sum_{r=1}^{R} \sum_{s=1}^{R'} \gamma_{rs}(e_r^* - e_s) \right\|_2 \tag{19}$$

$$\leq \sum_{r=1}^{R} \sum_{s=1}^{R'} \gamma_{rs} \|e_r^* - e_s\|_2 = \langle \gamma, D \rangle. \tag{20}$$

Since this inequality holds for every feasible $\gamma \in \Gamma(\mathbf{a}, \mathbf{b})$, it also holds for the minimum-cost feasible coupling, yielding

$$\|\bar{e}^* - \bar{e}\|_2 \leq \min_{\gamma \in \Gamma(\mathbf{a}, \mathbf{b})} \langle \gamma, D \rangle. \tag{21}$$

Equivalently, this follows from Kantorovich–Rubinstein duality (Villani et al., 2009). For equation 7, let $\mu_{A^*} = \sum_{r=1}^{R} a_r \, \delta_{e_r^*}$ and $\mu_B = \sum_{s=1}^{R'} b_s \, \delta_{e_s}$ denote the uniform empirical distributions of the two sets. Suppose the optimal transport cost is zero and let γ be an optimal plan. Since $\gamma_{rs} \geq 0$ and $D_{rs} = \|e_r^* - e_s\|_2 \geq 0$, a zero cost implies

$$\sum_{r,s} \gamma_{rs} D_{rs} = 0 \quad \Longrightarrow \quad \gamma_{rs} D_{rs} = 0 \qquad \text{for every } (r, s), \tag{22}$$

so $\gamma_{rs} > 0$ only if $e_r^* = e_s$. Hence, for every point x,

$$\mu_{A^*}(\{x\}) = \sum_{r:\, e_r^* = x} a_r = \sum_{r:\, e_r^* = x} \sum_{s=1}^{R'} \gamma_{rs} = \sum_{s:\, e_s = x} \sum_{r=1}^{R} \gamma_{rs} = \sum_{s:\, e_s = x} b_s = \mu_B(\{x\}), \tag{23}$$

where the middle equality holds because every nonzero γ_{rs} in either sum has $e_r^* = e_s = x$. Thus $\mu_{A^*} = \mu_B$. Contrapositively, $\mu_{A^*} \neq \mu_B$ implies

$$\min_{\gamma \in \Gamma(\mathbf{a}, \mathbf{b})} \langle \gamma, D \rangle > 0, \tag{24}$$

regardless of the means. In particular, this holds when $\bar{e}^* = \bar{e}$ but $\sigma^2_{A^*} \neq \sigma^2_B$, since distributions with different second central moments are distinct. This proves equation 7. □

C.3 Dual embedding space: Proof of Proposition 1

TSLM-ICL places time-series and text tokens in disjoint subspaces of $\mathbb{R}^{d_{ts}+d_{text}}$ and assigns them non-uniform mass (Eq. equation 8), so the uniform marginals assumed in Remark 2 no longer apply. Proposition 1 extends Eq. equation 6 to this setting.

Setup and notation. Write the query token embedding set as $A^* = \{\tilde{e}_r^*\}_{r=1}^{R+1}$, with zero-padded time-series tokens $\tilde{e}_r^* = (e_r^{ts,*}, \mathbf{0}_{d_{text}})$ for $r \leq R$ and text token $\tilde{e}_{R+1}^* = (\mathbf{0}_{d_{ts}}, \bar{e}^{text,*})$, and marginals $\mathbf{a} = \left(\frac{1-\alpha}{R}, \ldots, \frac{1-\alpha}{R}, \alpha\right)$. Define the demonstration set $A = \{\tilde{e}_s\}_{s=1}^{R'+1}$ analogously from $\{e_s^{ts}\}_{s=1}^{R'}$ and $\bar{e}^{text}$, with marginals $\mathbf{b} = \left(\frac{1-\alpha}{R'}, \ldots, \frac{1-\alpha}{R'}, \alpha\right)$. Let $\bar{e}^{ts,*} = \frac{1}{R}\sum_{r=1}^{R} e_r^{ts,*}$ and $\bar{e}^{ts} = \frac{1}{R'}\sum_{s=1}^{R'} e_s^{ts}$ be the pooled time-series token embeddings, $\rho_{text} = \|\bar{e}^{text,*} - \bar{e}^{text}\|_2$ the text distance, $D_{rs} = \|\tilde{e}_r^* - \tilde{e}_s\|_2$ the ground cost, and $W_1 = \min_{\gamma \in \Gamma(\mathbf{a}, \mathbf{b})} \langle \gamma, D \rangle$. We call the entries $(r, R'+1)$ with $r \leq R$ and $(R+1, s)$ with $s \leq R'$ *cross-modal*.

Proposition 1 (Dual embedding space). *Let A^* and A be the token embedding sets of Eq. equation 8 with marginals $\mathbf{a}$ and $\mathbf{b}$, and let $W_1 = \min_{\gamma \in \Gamma(\mathbf{a}, \mathbf{b})} \langle \gamma, D \rangle$ be their exact transport cost. If*

$$\|e_r^{ts,*} - e_s^{ts}\|_2 + \|\bar{e}^{text,*} - \bar{e}^{text}\|_2 \;<\; 2\sqrt{2} \qquad \textit{for all } r \leq R,\ s \leq R', \tag{25}$$

then every optimal plan places no mass across modalities, and

$$W_1 \;\geq\; \alpha \, \|\bar{e}^{text,*} - \bar{e}^{text}\|_2 \;+\; (1-\alpha) \, \|\bar{e}^{ts,*} - \bar{e}^{ts}\|_2. \tag{26}$$

The cost matrix. Since all tokens are ℓ_2-normalized and the two modalities occupy orthogonal subspaces, the ground cost takes three forms:

$$D_{rs} = \|e_r^{ts,*} - e_s^{ts}\|_2 \;\; (r \leq R,\ s \leq R'), \qquad D_{R+1,R'+1} = \rho_{text}, \qquad D_{r,R'+1} = D_{R+1,s} = \sqrt{2}. \tag{27}$$

The last identity follows from $\|(x, \mathbf{0}) - (\mathbf{0}, y)\|_2^2 = \|x\|_2^2 + \|y\|_2^2 = 2$. Hence Condition equation 25 reads $D_{rs} + D_{R+1,R'+1} < D_{r,R'+1} + D_{R+1,s}$ for all $r \leq R$ and $s \leq R'$: matching within each modality is strictly cheaper than routing the same mass through two cross-modal entries.

Proof. The proof proceeds in four steps.

Step 1 (mass balance). Let $\gamma \in \Gamma(\mathbf{a}, \mathbf{b})$ be any feasible plan. Row $R+1$ and column $R'+1$ both carry mass α, so

$$\sum_{r=1}^{R} \gamma_{r,R'+1} \;=\; \alpha - \gamma_{R+1,R'+1} \;=\; \sum_{s=1}^{R'} \gamma_{R+1,s}. \tag{28}$$

Consequently, γ has mass on some cross-modal entry of column $R'+1$ if and only if it has mass on some cross-modal entry of row $R+1$.

Step 2 (every optimal plan is modality-preserving). An optimal plan exists, since $\Gamma(\mathbf{a}, \mathbf{b})$ is nonempty and compact and the objective is linear. Let γ be optimal, and suppose, for contradiction, that it places mass on a cross-modal entry. By Eq. equation 28, there exist $r \le R$ and $s \le R'$ with $\gamma_{r,R'+1} > 0$ and $\gamma_{R+1,s} > 0$. Let $\mu = \min(\gamma_{r,R'+1}, \gamma_{R+1,s}) > 0$ and define γ' by

$$\gamma'_{r,R'+1} = \gamma_{r,R'+1} - \mu, \quad \gamma'_{R+1,s} = \gamma_{R+1,s} - \mu, \quad \gamma'_{rs} = \gamma_{rs} + \mu, \quad \gamma'_{R+1,R'+1} = \gamma_{R+1,R'+1} + \mu, \tag{29}$$

with all other entries unchanged. Each of rows $r, R+1$ and columns $s, R'+1$ loses μ on one entry and gains μ on another, and all entries remain non-negative, so $\gamma' \in \Gamma(\mathbf{a}, \mathbf{b})$. By Eq. equation 27 and Condition equation 25,

$$\langle \gamma', D \rangle - \langle \gamma, D \rangle \;=\; \mu\big(D_{rs} + \rho_{text} - 2\sqrt{2}\big) \;<\; 0, \tag{30}$$

which contradicts the optimality of γ. Hence every optimal plan $\hat{\gamma}$ satisfies $\hat{\gamma}_{r,R'+1} = \hat{\gamma}_{R+1,s} = 0$ for all $r \le R$ and $s \le R'$, which proves the first claim of the proposition.

Step 3 (the text block). Let $\hat{\gamma}$ be any optimal plan. Row $R+1$ has total mass α and, by Step 2, no mass in columns $s \le R'$, so $\hat{\gamma}_{R+1,R'+1} = \alpha$. By Eq. equation 27,

$$W_1 \;=\; \langle \hat{\gamma}, D \rangle \;=\; \alpha\, \rho_{text} \;+\; \sum_{r=1}^{R} \sum_{s=1}^{R'} \hat{\gamma}_{rs}\, \|e_r^{ts,*} - e_s^{ts}\|_2. \tag{31}$$

Step 4 (the time-series block). By Step 2, each time-series row and column of $\hat{\gamma}$ keeps its full marginal within the time-series block:

$$\sum_{s=1}^{R'} \hat{\gamma}_{rs} = \frac{1-\alpha}{R} \quad (r \le R), \qquad \sum_{r=1}^{R} \hat{\gamma}_{rs} = \frac{1-\alpha}{R'} \quad (s \le R'). \tag{32}$$

Therefore,

$$\sum_{r=1}^{R} \sum_{s=1}^{R'} \hat{\gamma}_{rs}\, (e_r^{ts,*} - e_s^{ts}) \;=\; \frac{1-\alpha}{R} \sum_{r=1}^{R} e_r^{ts,*} \;-\; \frac{1-\alpha}{R'} \sum_{s=1}^{R'} e_s^{ts} \;=\; (1-\alpha)\, (\bar{e}^{ts,*} - \bar{e}^{ts}), \tag{33}$$

and, by the triangle inequality and $\hat{\gamma} \ge 0$,

$$\sum_{r=1}^{R} \sum_{s=1}^{R'} \hat{\gamma}_{rs}\, \|e_r^{ts,*} - e_s^{ts}\|_2 \;\ge\; \Big\| \sum_{r=1}^{R} \sum_{s=1}^{R'} \hat{\gamma}_{rs}\, (e_r^{ts,*} - e_s^{ts}) \Big\|_2 \;=\; (1-\alpha)\, \|\bar{e}^{ts,*} - \bar{e}^{ts}\|_2. \tag{34}$$

Substituting this bound into Eq. equation 31 yields Eq. equation 26. □

Remarks. (i) *Strictness.* The strict inequality in Condition equation 25 is needed only for the first claim. If equality holds for some pair (r, s), the exchange of Step 2 is cost-neutral, and optimal plans with and without cross-modal mass may coexist; applying the exchange repeatedly then still yields *some* modality-preserving optimal plan, so the bound of Eq. equation 26, which depends only on the optimal value W_1, continues to hold. (ii) *A sufficient condition.* For unit vectors, $\|x - y\|_2^2 = 2 - 2\cos(x, y)$, so each distance is below $\sqrt{2}$ whenever the corresponding cosine similarity is positive. Condition equation 25 therefore holds whenever every pair of time-series tokens across the two sets, and the two text tokens, have positive cosine similarity. (iii) *Entropic plans.* The proposition concerns exact transport. The entropic plan γ_ε of Eq. equation 9 has strictly positive entries and therefore always places some mass across modalities; we quantify this below.

Validity in practice. We checked Condition equation 25 on the Chronos-2 and text token embedding sets TSLM-ICL retrieves with, over all 39 subtasks of the five benchmarks (19,611 queries, every query–candidate pair within a subtask's pool). The condition holds for 99.8% of pairs overall; the lowest benchmark is TSExam with 95.3%, and the lowest subtask is at 92.1%. The pairs where it fails are not the ones that matter for retrieval: it holds for every top-1 demonstration and for all but one of the $\approx$115k query–demonstration pairs selected at $k = 4$ (99.99% at $k = 8$). On

a sample of 100 pairs per subtask, we also solved the exact problem with a network-simplex solver and confirmed that, where the condition holds, the optimal cost equals that of the best plan with no cross-modal mass to within 10^{-15}, as Proposition 1 asserts.

The proposition concerns the exact plan, whereas TSLM-ICL uses the entropic plan γ_ε of Eq. equation 9. Since γ_ε has strictly positive entries, it always places some mass across modalities; the question is how much. At the ε we use, it stays close to the exact plan: in the example of Figure 10, the text point sends 50.0% of the total mass (i.e. all of its mass $\alpha = 0.5$, up to rounding) to the other sample's text point at $\varepsilon \leq 0.05$, and 49.7% at $\varepsilon = 0.1$ (Appendix G.2). Hence the modality-preserving structure of Proposition 1 also describes the plan used in practice, up to an ε-dependent leakage that is negligible at $\varepsilon = 0.03$.

D DATASETS

To evaluate TSLM-ICL, we utilize five public multimodal time-series benchmarks comprising a total of 39 sub-datasets and 19,611 samples. These benchmarks span a diverse range of domains, task types (such as classification, reasoning, and question answering), and temporal structures. This section details the datasets used in our evaluation. We first outline the data provenance and our unification pipeline, followed by a high-level benchmark overview and a detailed sub-dataset breakdown. Finally, we analyze the structural heterogeneity of both the time-series and text modalities across the corpus and provide representative worked examples for each benchmark.

D.1 SOURCES, SUBSAMPLING, AND UNIFICATION

The raw data for our evaluation is sourced from public HuggingFace repositories. The list below summarizes the origin, data splits, and minor preprocessing steps applied to each benchmark. Licensing information is maintained on the respective HuggingFace dataset cards.

- **TSExam:** Sourced from `AutonLab/TimeSeriesExam1` (test split). A category typo in the upstream dataset ("Anolmaly Detection" to "Anomaly Detection") was corrected to ensure accurate sub-dataset grouping.
- **TSR-Bench:** Sourced from `umd-zhou-lab/TSRBench` (full dataset). Similar upstream category typos were corrected to map samples to their proper subtasks.
- **MTBench:** Sourced from 8 repositories under `afeng/MTBench_*` covering finance and weather domains. Forecasting subtasks were excluded from our evaluation, and we use the full datasets for the remaining tasks.
- **Time-MQA:** Sourced from `Time-MQA/TSQA`. Forecasting, imputation, and open-ended QA subtasks were excluded from our evaluation. Additionally, we applied a formatting fix to properly parse negative numbers (correcting text artifacts where minus signs were improperly encoded). The remaining pool was subsampled to a maximum of 1,000 samples per subtask using seed 0.
- **TimeOmni:** Sourced from `anton-hugging/timeomni-1-testbed`. For the retained subtasks we evaluate on the full set, since the release defines no train/test split; the two forecasting subtasks were excluded.

Unification pipeline. To allow a single retrieval pipeline and prompt template to execute seamlessly across all benchmarks and TSLMs, all five datasets were converted into a single, unified JSONL schema. Every sample was standardized into the tuple $x = (\tau, w)$. The time-series modality τ is stored as a list of series, each a 2D array of `[variate][timestep]`. The text modality w retains the primary question and any associated extra text fields (options, context articles, or system prompts). Finally, the ground-truth target for each sample was mapped to a standardized dictionary indicating the expected output format (for example, mapping a multiple-choice target to a single letter like `A`, or a classification target to its specific label string).

D.2 BENCHMARK OVERVIEW

Table 6 summarizes the five multimodal benchmarks used in our evaluation. While every sample in our corpus carries at least one time series and a primary question, the benchmarks differ significantly

in their supplementary textual modalities. Depending on the dataset, the prompt may incorporate explicitly structured MCQ choices, extensive context articles, or rigid system instructions.

Table 6: **Benchmark overview.** Five benchmarks with 39 sub-datasets and 19,611 samples. "Extra Text" denotes text beyond the primary question.

Benchmark	Subtasks	Samples	Task Type(s)	Answer Format	Extra Text
TSExam	5	746	MCQ	Letter	Choices (all samples)
TSR-Bench	15	4,125	MCQ	Letter	Choices / Merged into question
MTBench	10	8,580	Class., MCQ	Label, Letter	Context article (in question)
Time-MQA	3	3,000	Anomaly, Class.	Label	—
TimeOmni	6	3,160	Class.	Text	System prompt formatting

D.3 Per-Sub-Dataset Breakdown

Table 7 details all 39 sub-datasets. Because retrieval operates on a leave-one-out basis within a single sub-dataset, the demonstration pool available for any given query is exactly $N - 1$, where N is the total sample count of its specific sub-dataset.

Table 7: **Sub-Dataset Breakdown.** Total sample count (N), task type, answer format, and time-series shape for all 39 sub-datasets. For MCQ tasks, options are either provided in a structured `choices` field or stated *inline* within the question.

Benchmark / Subtask	N	Task	Answer	Options	Variates (Avg/Max)	Len. (Range)
TSExam (All tasks evaluate MCQ with letter answers; domain unspecified)						
anomaly_detection	108	mcq	letter	2–3	1.28 / 2	1024–1204
causality_analysis	72	mcq	letter	2–3	2.00 / 2	1024
noise_understanding	84	mcq	letter	2–3	1.14 / 2	1024
pattern_recognition	362	mcq	letter	2–4	1.04 / 2	1024
similarity_analysis	120	mcq	letter	2–3	2.00 / 2	1024
TSR-Bench (Diverse domains: Weather, Finance, Physics, ECG, etc.)						
abductive_reasoning	150	mcq	letter	4	2.00 / 2	274–416
anomaly_detection	129	mcq	letter	inline	1.31 / 2	128–308
causal_reasoning	300	mcq	letter	4	5.00 / 5	64–1024
deductive_reasoning	250	mcq	letter	4	4.00 / 4	120
etiological_reasoning	350	mcq	letter	5–6	2.14 / 3	24–883
event_prediction	360	mcq	letter	2	5.00 / 5	24
inductive_reasoning	100	mcq	letter	4	4.50 / 8	24–396
noise_understanding	87	mcq	letter	inline	1.15 / 2	128
numerical_reasoning	400	mcq	letter	2–4	1.88 / 4	20–1007
pattern_recognition	371	mcq	letter	inline	1.04 / 2	128
qualitative_decision	335	mcq	letter	4	12.00 / 12	1000
quantitative_decision	300	mcq	letter	5	1.00 / 1	200–251
similarity_analysis	113	mcq	letter	inline	2.00 / 2	128
temporal_relation_reasoning	160	mcq	letter	4	1.00 / 1	63–723
time_series_forecasting	720	mcq	letter	4	1.00 / 1	35–140
MTBench (All tasks univariate: avg/max variates = 1; domains: Finance, Weather)						
finance_correlation_long	516	class.	label	—	1.00 / 1	114–144
finance_correlation_short	491	class.	label	—	1.00 / 1	275–402
finance_mcqa_long	516	mcq	letter	inline	1.00 / 1	114–144
finance_mcqa_short	491	mcq	letter	inline	1.00 / 1	275–402
finance_trend_long	525	class.	label	—	1.00 / 1	115–1303
finance_trend_short	750	class.	label	—	1.00 / 1	275–402
weather_qa_long	728	mcq	letter	inline	1.00 / 1	336
weather_qa_short	645	mcq	letter	inline	1.00 / 1	168
weather_trend_future	1,959	class.	label	—	1.00 / 1	168
weather_trend_past	1,959	class.	label	—	1.00 / 1	168
Time-MQA (All pools subsampled to maximum $N = 1{,}000$; broad domains: HAR, Health, Web)						
anomaly_detection	1,000	anomaly	label	—	1.00 / 1	8–64
classification_activity	1,000	class.	label	—	1.00 / 1	30
classification_freeze	1,000	class.	label	—	1.00 / 1	9
TimeOmni (Evaluated on In-Distribution (ID) and Out-Of-Distribution (OOD) splits)						
id_causality_discovery	800	class.	text	—	2.00 / 2	56–62
id_decision_making	188	class.	text	—	1.00 / 1	48
id_scenario_understanding	200	class.	text	—	1.00 / 1	12–745
ood_causality_discovery	800	class.	text	—	2.00 / 2	21–62
ood_decision_making	273	class.	text	—	1.00 / 1	48
ood_scenario_understanding	899	class.	text	—	1.00 / 1	10–790

D.4 TEXT-MODALITY STATISTICS

The textual footprint of a sample varies by nearly two orders of magnitude across benchmarks, from a couple of hundred characters in TSExam to tens of thousands in MTBench. This heterogeneity motivates a joint text-embedding representation whose contribution is tuned by the mass parameter α. Table 8 reports the total textual input per sample—the combined length of all distinct text fields. MTBench samples embed a long news or context article *inside* the question, reaching $\sim$89,000 characters; that article is strictly counted once. TimeOmni adds a near-constant $\sim$194-character system prompt that enforces its output format.

Table 8: **Text per sample.** Total characters across distinct text fields (mean / median / max); MT-Bench context is counted once.

Benchmark	Fields Included	Mean	Median	Max
TSExam	Question, Choices	165	158	603
TSR-Bench	Question, Choices	2,472	971	28,589
MTBench	Question (contains Context)	6,006	4,516	88,978
Time-MQA	Question	663	646	854
TimeOmni	Question, System Prompt	1,145	812	3,056

D.5 WORKED EXAMPLES

To ground the structural differences between the benchmark samples, the panels below demonstrate one typical evaluation instance from each dataset. For readability, we summarize each instance's structure rather than reproduce the full prompt text. For fully rendered versions of the serialized prompts (including the text, numeric, and visual representations of the TSExam benchmark), see Appendix E.

TSExam Example (Similarity Analysis)

Series: 2 variates × 1,024 steps
Question: Are the given two time series likely to have the same underlying distribution?
Choices: A) No, they have different underlying distribution B) Yes, they have the same underlying distribution: Gaussian White Noise
Answer: Letter (A or B)

TSR-Bench Example (Qualitative Decision)

Series: 12 variates × 1,000 steps (12-lead ECG)
Question: ... [1,844 chars of expert-cardiologist ECG-diagnosis instructions] ...
Choices: 4 clinical diagnosis options
Answer: Letter

MTBench Example (Finance MCQA Short)

Series: 1 variate × ∼390 steps
Question: ... [10.6k-char question (expert-analyst instruction + inline news article + options A--D)] ...
Answer: Letter (e.g., C)

Time-MQA Example (Activity Classification)

Series: 1 variate × 30 steps (20 Hz accelerometer)
Question: ... [activity-recognition prompt over six states: walking, jogging, sitting, standing, upstairs, downstairs] ...
Answer: Label (e.g., Jogging)

TimeOmni Example (ID Causality Discovery)

```
Series:  2 variates
Question:  You are given two time series related to river discharge
(m^3/s)...  causal discovery...
System Prompt:  Specifies <think>/<answer> output format
Answer:  Free text (scored by official TimeOmni scorer)
```

E MODELS

We evaluate six frozen TSLMs (Table 1, right), chosen to span the three distinct ways a language model can be given a time series: as *numeric text*, a *rendered plot*, or through a *native time-series encoder*. The retrieval pipeline, prompt wording, decoding, and context budgets are identical across all models; only the serialization of the series differs. All models are used as released.

Numeric text (Qwen3-4B, Time-MQA, TimeOmni). Three models read the series as numeric text. *Qwen3-4B* (Yang et al., 2025a) is a general-purpose LLM with no time-series training, while *Time-MQA-7B* (Kong et al., 2025) and *TimeOmni-1-7B* (Guan et al., 2026) are adapted specifically for time-series QA and reasoning. Each series is written as comma-separated values at 2-decimal precision and inlined into the question in place of its series marker (Figure 5). A length-L, C-variate sample therefore contributes $\sim C\,L$ numbers at $\approx$6 tokens each, making this the modality whose prompt grows fastest with the series and drives the truncation discussed in Appendix E.1.

Rendered plot (Qwen3-VL, Gemma-3). *Qwen3-VL-8B* (Bai et al., 2025) and *Gemma-3-12B* (Team et al., 2025) are general-purpose vision-language models. For these, each series is drawn as a 640×320 line plot over an integer time-step axis and passed to the model's image encoder in place of its marker (Figure 6); a sample with several series therefore becomes several images. Both VL models share the same canvas formatting, and the plot's token cost is roughly constant in L (a few hundred tokens per image), bypassing the heavy token scaling of the text path.

Native encoder (ChatTS). *ChatTS-8B* (Xie et al., 2025a) uses its own dedicated time-series encoder. It passes the raw values to this encoder, which turns a length-L variate into $\approx L/8$ patch embeddings (patch size 8) that are spliced directly into the text stream at a `<ts><ts/>` marker. The surrounding text matches the numeric path, but its per-demonstration cost scales closer to the plot path rather than the text path.

Overlap between training data and benchmarks. Two of the time-series-specialized models were trained on data from the benchmark released with them, and for these two model–benchmark pairs zero-shot accuracy exceeds most ICL configurations. *Time-MQA-7B* is continually pretrained on TSQA (Kong et al., 2025), the corpus from which our Time-MQA benchmark is drawn. The release defines no train/test split, so our subsample of 1,000 samples per subtask (Appendix D.1) can overlap the model's training data. Its zero-shot accuracy on Time-MQA (0.646) is higher than TSLM-ICL at $k = 1$ and $k = 4$ (0.584 and 0.615), and is exceeded only at $k = 8$ (0.708; Table 30). *TimeOmni-1-7B* is evaluated on the TimeOmni testbed, of which half of the subtasks are designated in-distribution with respect to the model's training data by its authors (Guan et al., 2026); these are the ID subtasks of Table 7. Its zero-shot accuracy (0.651) is higher than TSLM-ICL at every budget (0.639, 0.531 and 0.461 at $k = 1, 4, 8$; Table 38). For both pairs the zero-shot score is a partly in-distribution reference, so this behaviour is expected and reflects the training data of these two models rather than the retrieval method. On the other four benchmarks both models improve over zero-shot from $k = 4$ onward, as does every other model on Time-MQA and TimeOmni.

E.1 CONTEXT-WINDOW LIMITS ON THE DEMONSTRATION BUDGET

For long or multivariate samples, the k demonstrations requested by a k-shot prompt can exceed the model's context window, so the *effective* number read, m, falls below k. This follows from the per-value token cost of serializing a time series and applies identically to every method under the same budget.

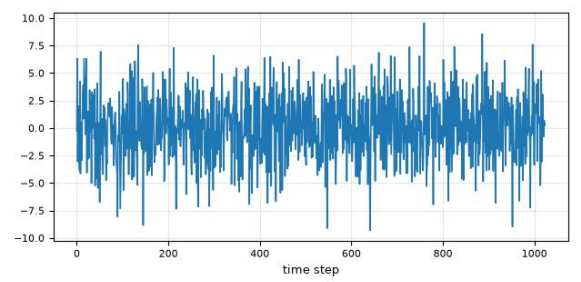
Numeric-text prompt (coloured = values inserted by the serializer)

```
You are given two time series here, where each step is separated
by a comma.  Timeseries1:  -0.26, 6.30, -3.07, 0.53, 1.98, -1.49,
-3.86, 4.24, -4.14, 0.52, -1.30, -2.93, ...  Timeseries2:  0.07,
-0.04, -0.06, 0.19, 0.09, 0.05, 0.08, 0.08, 0.19, 0.05, 0.27, 0.00,
...  Answer the following question based on the time series.  ...
Question:  Are the given two time series likely to have the same
underlying distribution?   A) No, they have different underlying
distribution   B) Yes, they have the same underlying distribution:
Gaussian White Noise
```

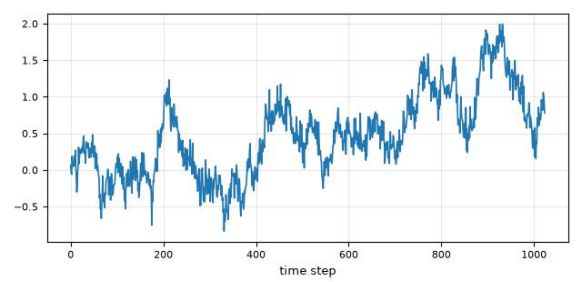
Figure 5: **Numeric-text serialization.** A TSExam similarity sample as the text models receive: series values (coloured) replace each `Timeseries`k`:` marker. Truncated for display.

Vision-language prompt (the same template as Figure 5, with each series rendered to a plot)

```
You are given two time series here, where each step is separated by a
comma.  Timeseries1:
```




```
Timeseries2:
```




```
Answer the following question based on the time series.  ...
Question:  Are the given two time series likely to have the same
underlying distribution?   A) No, they have different underlying
distribution   B) Yes, they have the same underlying distribution:
Gaussian White Noise
```

Figure 6: **Rendered-plot serialization.** The VL prompt for the same sample as Figure 5: the template is unchanged, but each series is a rendered plot.

Budget and overflow policy. Requests are assembled under a fixed budget $\beta = \text{max_model_len} - \text{max_new_tokens} - 64 = 32768 - 4096 - 64 = 28{,}608$ tokens. With tokenizer count $\text{tok}(\cdot)$ and the ranking $(x_1, \ldots, x_k)$ of Eq. equation 9 (most-relevant first; the prompt presents it reversed), the assembler keeps the largest prefix $(x_1, \ldots, x_m)$ with $\text{tok}(I, x_{1:m}, x^*) \le \beta$ and drops the tail; text precision is fixed at 2 decimals and demonstrations are never subsampled. Only when $m = 0$ and the query alone exceeds β is the query series decimated ($512 \to \cdots \to 8$ points). We replay this

Native-encoder prompt (the same template with encoder markers)

```
You are given two time series here, where each step is separated by
a comma.  Timeseries1:  <ts><ts/> Timeseries2:  <ts><ts/> Answer
the following question based on the time series.  ...  Question:
Are the given two time series likely to have the same underlying
distribution?   A) No, they have different underlying distribution
B) Yes, they have the same underlying distribution:  Gaussian White
Noise
```

Figure 7: **Native-encoder serialization.** The ChatTS prompt for the same sample. The template is unchanged, but the raw series are passed to the time-series encoder, which injects patch embeddings directly at the <ts><ts/> markers.

assembly offline for our main experiments across all five benchmarks, six TSLMs, and $k \in \{1, 4, 8\}$; the recovered m matches the values logged during the scored runs (Table 9).

Table 9: **Effective demonstrations m at $k = 8$.** Mean m per model and benchmark after budget-driven dropping. **Bold** marks TSExam text models, which retain about two demonstrations.

Model	TSExam	TSR-Bench	MTBench	Time-MQA	TimeOmni
ChatTS-8B	8.00	6.39	7.85	8.00	8.00
Qwen3-VL-8B	8.00	5.83	7.79	8.00	8.00
Gemma-3-12B	8.00	6.86	7.85	8.00	8.00
Qwen3-4B	**2.01**	5.86	6.82	8.00	7.71
TimeMQA-7B	**2.01**	5.86	6.82	8.00	7.71
TimeOmni-1-7B	**2.01**	5.86	6.82	8.00	7.71

The effective count is set by the marginal token cost per demonstration ν (Table 10): numeric-text serialization emits ≈ 6 tokens per value, so $\nu \sim 6\,C\bar{L}$ for C variates of mean length $\bar{L}$ and $m \approx \lfloor \beta/\nu \rfloor$, whereas rendered plots and TS encoders are near-constant in length. Truncation concentrates in the text models on TSExam ($m \approx 2$ at both k=4, 8) and in all models on TSR-Bench (the heaviest text cells reach $m = 0$); elsewhere $m \approx 8$. On affected cells, k-curves should be read against m; fixed-k comparisons *between* methods share β and are unaffected.

Table 10: **Marginal tokens per demonstration ν.** Set by the serialization (numeric text, rendered plot, or TS encoder); text models differ only by tokenizer.

Model	Series input	TSExam	TSR-Bench	MTBench	Time-MQA	TimeOmni
Qwen3-4B	Numeric text	6377	2150	3144	346	1229
TimeMQA-7B	Numeric text	6377	2316	3144	346	1229
TimeOmni-1-7B	Numeric text	6377	2316	3144	346	1229
Qwen3-VL-8B	Rendered plot	1336	2733	2267	1004	1362
Gemma-3-12B	Rendered plot	534	1707	1738	461	641
ChatTS-8B	TS encoder	1678	1313	1721	228	442

F Retrieval Methods

This section describes the retrieval methods compared in Table 2, both as originally proposed and as adapted to multimodal time-series ICL. Following Section 3.1, each method is specified by a representation E, a comparison $\mathrm{sim}\big(E(x^*), E(x_i)\big)$ and a selection rule (Table 11).

Shared protocol. TimeRAG and TS-RAG were proposed for retrieval-augmented *forecasting*. Each searches a dedicated knowledge base of historical windows and passes the retrieved windows, or their continuations, to the forecaster as additional context. Neither was designed for multimodal inputs or for question answering. We therefore port only their *retrieval* components, the representation and the similarity, and place every method in one ICL protocol. The candidate pool of a query

x^* is its leave-one-out sub-dataset $P = \{(x_i, y_i)\}_{i=1}^{N-1}$, which replaces each method's own knowledge base. A retrieved item is a complete demonstration (x_i, y_i), series, text and answer, rendered in the TSLM's input format of Eq. equation 2. Every scoring method ranks P by its similarity to x^*, keeps the top k, and orders them by increasing relevance, so the most similar demonstration immediately precedes the query. All methods share the same k, the same prompt budget and the same truncation rule (Appendix E.1). Each method's $N \times N$ similarity matrix is computed once per sub-dataset and cached (Appendix B.2), so all six TSLMs see identical demonstrations for a given method, sub-dataset and k. Methods therefore differ only in which demonstrations they select.

Table 11: **Retrieval methods decomposed into representation, comparison and selection.** τ_1 is the first variate, $z(\cdot)$ per-variate z-scoring, f_{text} the pooled MiniLM sentence encoder, and h^{REG} the summary token of the time-series encoder. Every method except Random selects the top k.

Method	Representation $E(x)$	Comparison	Modalities
Random	—	— (uniform sample)	—
Text	$f_{\text{text}}(w)$ (MiniLM)	cosine	text
TimeRAG	raw first variate τ_1	$-\text{DTW}/M$	first variate
TS-RAG	$\frac{1}{C}\sum_c h^{\text{REG}}\big(z(\tau_c)\big)$ (Chronos-Bolt)	cosine	all variates, pooled
TSLM-ICL	token set A, Eq. equation 8 (Chronos-2)	$-$entropic OT	all variates + text

F.1 RANDOM

Random selection is the standard reference for in-context learning (Brown et al., 2020). It measures how much a TSLM gains from demonstrations of the task alone, independent of their relevance to the query. We draw S uniformly without replacement from P,

$$S \sim \text{Unif}\Big\{S' \subseteq P : |S'| = \min(k, |P|)\Big\}, \tag{35}$$

and seed the generator with a hash of the query's identifier. The draw is therefore deterministic, identical across TSLMs and across reruns, and independent between queries. Random has no relevance score, so demonstrations are placed in the order drawn. The gap between Random and a scoring method isolates the contribution of relevance-based selection, since both show the TSLM demonstrations from the same sub-dataset in the same format.

F.2 TEXT

Text is the retriever of text-only ICL, applied to the textual half of a multimodal sample. We follow KATE (Liu et al., 2022), which selects the nearest neighbors of the query in the embedding space of a pretrained sentence encoder and improves consistently over random demonstrations in NLP. Each sample's question and textual context w are embedded by the Sentence-BERT (Reimers & Gurevych, 2019) model `all-MiniLM-L6-v2` (Wang et al., 2020) into a single mean-pooled vector $f_{\text{text}}(w) \in \mathbb{R}^{384}$, and demonstrations are scored by

$$\text{sim}_{\text{text}}(x^*, x_i) \;=\; \frac{\langle f_{\text{text}}(w^*), f_{\text{text}}(w_i)\rangle}{\|f_{\text{text}}(w^*)\|_2 \, \|f_{\text{text}}(w_i)\|_2}. \tag{36}$$

The encoder and input string are exactly those of the text point in TSLM-ICL (Eq. equation 8), so Text coincides with the limit $\alpha \to 1$ of our method. It ignores the time series entirely. It is therefore expected to perform well when a sub-dataset's questions vary informatively, and poorly when samples share one question template and differ only in their series, in which case its ranking is close to arbitrary.

F.3 TIMERAG

TimeRAG (Yang et al., 2025b) augments an LLM forecaster with retrieved history. Its knowledge base is built by slicing historical series with a sliding window and clustering the slices with K-means, keeping one representative per cluster. At inference, the query window is matched to the knowledge base by dynamic time warping (DTW) (Sakoe & Chiba, 1978), and the retrieved sequences are serialized into the LLM prompt as reference examples. Given univariate series $\tau \in \mathbb{R}^L$

and $\tau' \in \mathbb{R}^{L'}$, a warping path $\pi = \big((p_1, q_1), \ldots, (p_M, q_M)\big)$ is a monotone, contiguous alignment from $(1, 1)$ to (L, L'). TimeRAG scores a candidate by the DTW cost of the optimal path, normalized by that path's length M,

$$\mathrm{dist}_{\text{TimeRAG}}(\tau, \tau') \;=\; \frac{1}{M}\sqrt{\sum_{\ell=1}^{M} \big(\tau_{p_\ell} - \tau'_{q_\ell}\big)^2}, \qquad \pi \;=\; \arg\min_{\pi'} \sum_{(p,q)\in\pi'} (\tau_p - \tau'_q)^2. \tag{37}$$

DTW compares raw values under an elastic alignment in time. It tolerates shifts and local speed changes, but it is defined only for univariate sequences and costs $O(LL')$ per pair.

Adaptation. We port TimeRAG's scoring rule (Eq. equation 37) exactly: the optimal path is recovered explicitly so that M is the true path length, no Sakoe–Chiba band is imposed, and values are not z-scored, as in the original method. The sliding-window and K-means knowledge base is replaced by the leave-one-out pool, because the unit of retrieval here is a complete question–answer sample rather than a series fragment. TimeRAG defines no rule for multivariate inputs, and aligning variate c of the query with a different variate c' of a candidate is not meaningful when variates are unordered and heterogeneous. We therefore compare the first variates only, $\mathrm{sim}(x^*, x_i) = -\mathrm{dist}_{\text{TimeRAG}}(\tau^*_1, \tau_{i,1})$, which leaves the method well defined for any C and for unequal lengths $L_1 \neq L'_1$. Its similarity ignores the text and all remaining variates. This is the source of the univariate row in Table 1 and of the flat variate curve in Figure 8.

F.4 TS-RAG

TS-RAG (Ning et al., 2026) replaces handcrafted distances with a frozen time-series foundation model. It encodes every window of a multi-domain knowledge base with the Chronos-Bolt encoder (Ansari et al., 2024), retrieves the top-k nearest windows to the query by the distance between their embeddings, and passes the retrieved windows' future horizons to an Adaptive Retrieval Mixer that blends them into the forecaster's representation. Chronos-Bolt splits an instance-normalized series into patches of length 16, encodes them with a T5 encoder, and appends a `[REG]` token. TS-RAG takes the final hidden state at the `[REG]` position, $h^{\text{REG}}(\tau_c) \in \mathbb{R}^{768}$, as the representation of a window. Because the representation is a single fixed-size vector, comparison reduces to one inner product and the entire pool can be searched with a nearest-neighbor index.

Adaptation. We keep TS-RAG's encoder (`amazon/chronos-bolt-base`) and its `[REG]` representation, and replace the knowledge base by the leave-one-out pool. The retrieved futures and the Adaptive Retrieval Mixer have no counterpart in ICL with a frozen TSLM: the full demonstration (x_i, y_i) plays that role. Chronos-Bolt is univariate, so each variate is z-scored and encoded independently, and a sample is summarized by the mean of its per-variate summary tokens,

$$\bar{h}(x) \;=\; \frac{1}{C}\sum_{c=1}^{C} h^{\text{REG}}\big(z(\tau_c)\big), \qquad \mathrm{sim}_{\text{TS-RAG}}(x^*, x_i) \;=\; \frac{\langle \bar{h}(x^*), \bar{h}(x_i)\rangle}{\|\bar{h}(x^*)\|_2\, \|\bar{h}(x_i)\|_2}. \tag{38}$$

Mean pooling is the natural single-vector extension to $C > 1$ variates. It is defined for any number of variates of any lengths and reduces to the original representation when $C = 1$. We use cosine similarity, as for Text, rather than the Euclidean index of the original implementation. The sample representation is thus a pooled mean $\bar{e}$ in the sense of Eq. equation 3. It uses every variate, unlike TimeRAG, but discards their dispersion (Remark 1) and the text. Appendix G.1 replaces Chronos-Bolt with Chronos-2 to separate the effect of the encoder from that of the comparison.

F.5 TSLM-ICL

Our method (Section 4.4) differs from the two time-series baselines along all three axes. *Representation:* Chronos-2 (Ansari et al., 2025) encodes all variates of a sample jointly, with group attention across variates, and every token is kept: the patch tokens of each variate together with its `[REG]` and future tokens. The pooled MiniLM embedding of the text is added to the same set as one extra point, in a disjoint subspace (Eq. equation 8). *Comparison:* two samples are compared by entropic optimal transport between their token sets (Eq. equation 9), which is defined for sets of different sizes and retains their dispersion (Remark 2). *Selection:* top-k with increasing-relevance ordering,

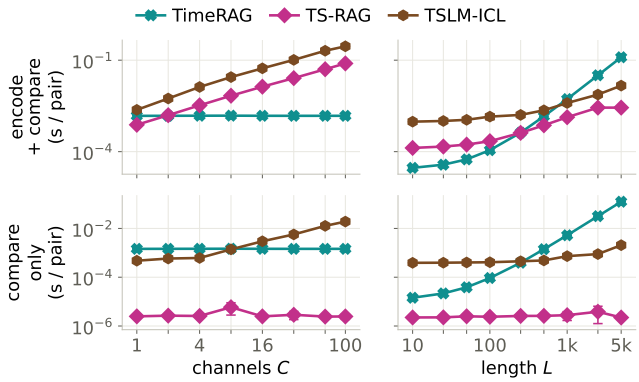


Figure 8: **Retrieval cost.** Seconds per pair against variates C and length L, with encoding (top) and without (bottom); mean ± 1 std over three passes, log–log axes.

as for the baselines. The limits of the text mass α relate it to the other methods: $\alpha \to 1$ recovers Text, and replacing transport by the distance between set means recovers a pooled retriever in the style of TS-RAG. Hyperparameters are listed in Table 5.

F.6 Retrieval Cost

Figure 8 times one similarity computation as the input grows, split into a per-sample *encode* stage, paid once per sample and cached, and a per-pair *compare* stage. With encoding, cost grows with the number of variates for both learned retrievers. At $L = 512$, TSLM-ICL rises from 2.4 ms per pair at $C = 1$ to 0.29 s at $C = 100$, about $3.6\times$ TS-RAG (78 ms). Most of this is the Chronos-2 forward pass over all variates; the transport accounts for 19 ms. TimeRAG stays at 1.5 ms, since it reads only the first variate. Along the length axis ($C = 1$) the ordering reverses. DTW is quadratic in L and becomes the most expensive method beyond $L \approx 1{,}000$, reaching 0.13 s per pair at $L = 5{,}000$, against 15 ms for TSLM-ICL and 2.8 ms for TS-RAG. For comparison alone, TS-RAG costs one inner product ($\approx 2.5\,\mu$s, independent of C and L), while the Sinkhorn solve of TSLM-ICL costs 0.4 ms on short univariate series and 19 ms at $C = 100$. All similarities are computed once per sub-dataset and cached (Appendix B.2), so these costs are paid offline rather than at inference time.

G Additional Experiments

G.1 TS-RAG with the Chronos-2 Encoder

TS-RAG and TSLM-ICL differ in both their time-series encoder and retrieval mechanism: TS-RAG uses a pooled Chronos representation with cosine similarity, whereas TSLM-ICL compares sets of temporal and textual tokens via optimal transport. To isolate these factors, we evaluate TS-RAG with Chronos-2 (C-2) while keeping its retrieval procedure unchanged in Table 12. Chronos-2 improves TS-RAG by 1.4, 1.4, and 1.7 points for $k = 1, 4, 8$, respectively, indicating that part of the gap stems from the encoder. However, with the encoder fixed, TSLM-ICL still outperforms TS-RAG (C-2) by 2.2, 1.9, and 1.2 points on average, with the largest gains on TSR-Bench and the smallest on Time-MQA. Thus, these results emphasize that the gains of TSLM-ICL arise not only from the stronger encoder, but also from its optimal-transport-based comparison of temporal and textual embeddings.

G.2 Transport: the Sinkhorn Regularization

Figure 9 varies the entropic regularization ε, which controls how sharply the plan concentrates its mass. As $\varepsilon \to \infty$ the plan tends to the product of its marginals, $\gamma_\varepsilon \to \mathbf{a}\mathbf{b}^\top$, and the score tends to the mean pairwise distance, which no longer rewards matching tokens. As $\varepsilon \to 0$ the plan tends to the exact optimal plan of Eq. equation 5. Over the range the float32 solver supports, $\varepsilon \in \{0.03, 0.05, 0.1\}$, accuracy is nearly flat. On TSExam all three values lie within 0.006 of each other at every k, less than one standard error (≈ 0.008). On TSR-Bench, $\varepsilon = 0.03$ is best at $k = 1$

Table 12: **TS-RAG with its original Chronos-Bolt encoder and with Chronos-2 (C-2), against TSLM-ICL.** Mean accuracy over six TSLMs and the subtask cells shared by these three methods, so values can differ slightly from Table 2. The *Average* weights every subtask equally; best values within each k block are in **bold**.

	Method	TSExam	TSR-Bench	Time-MQA	Average
$k=1$	TS-RAG	0.551	0.433	0.577	0.477
	TS-RAG (C-2)	0.590	0.439	0.587	0.491
	TSLM-ICL	**0.608**	**0.466**	**0.588**	**0.513**
$k=4$	TS-RAG	0.593	0.474	0.665	0.524
	TS-RAG (C-2)	0.642	0.475	0.681	0.538
	TSLM-ICL	**0.649**	**0.501**	**0.683**	**0.557**
$k=8$	TS-RAG	0.596	0.481	0.708	0.536
	TS-RAG (C-2)	0.642	0.489	0.723	0.553
	TSLM-ICL	**0.649**	**0.505**	**0.728**	**0.565**

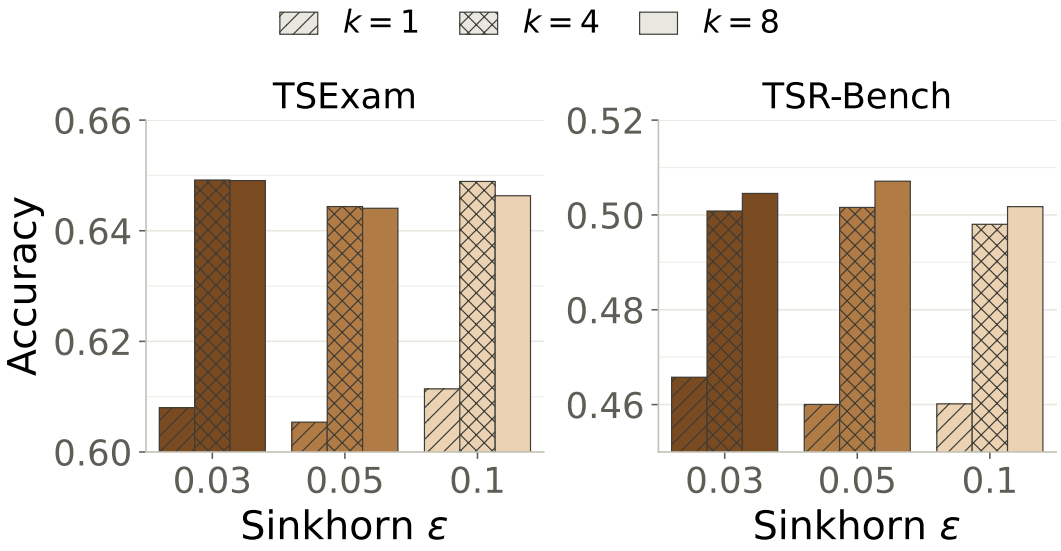


Figure 9: **Sinkhorn regularization ε.** Accuracy of TSLM-ICL as the entropic regularization with top-k and text mass $\alpha = 0.5$ fixed. Bars show the mean over shared cells for $k \in \{1, 4, 8\}$; $\varepsilon = 0.03$ is ours. Values below 0.025 are numerically unstable in float32. Axes are truncated per dataset.

(0.466 against 0.460 for both larger values), and the three are within 0.005 at $k = 4$ and $k = 8$. TSLM-ICL is therefore robust to this hyperparameter. We use $\varepsilon = 0.03$, the smallest value that is numerically stable in float32 (Appendix B.3), since it gives the plan closest to exact transport at no cost in accuracy.

Figure 10 shows what ε changes in the plan itself, for a pair of samples with three and two variates (23×18 tokens). We summarize the plan by its number of *effective pairs*, $\exp H(\gamma)$, which ranges from about 22 for a permutation-like plan to 374 for the product plan, and by the mean fraction of each token's mass sent to its single best match (the *top-1 share*). Decreasing ε from 0.1 to 0.03 halves the effective pairs ($169 \to 86$) and raises the top-1 share from 27% to 48%, so each token is matched to a few similar tokens rather than spread over the whole demonstration. The transport cost changes much less, from 0.798 to 0.776, because the additional mass lands on nearby tokens. This explains why the plan sharpens while accuracy stays flat: retrieval depends only on how candidates are ranked, and a sharper plan of nearly the same cost rarely changes that ranking. At every ε the text point sends essentially all of its mass $\alpha = 0.5$ to the other sample's text point (49.7% of the total mass at $\varepsilon = 0.1$, 50.0% below), so the regularization leaves the text-to-text coupling almost unchanged. This agrees with Proposition 1.

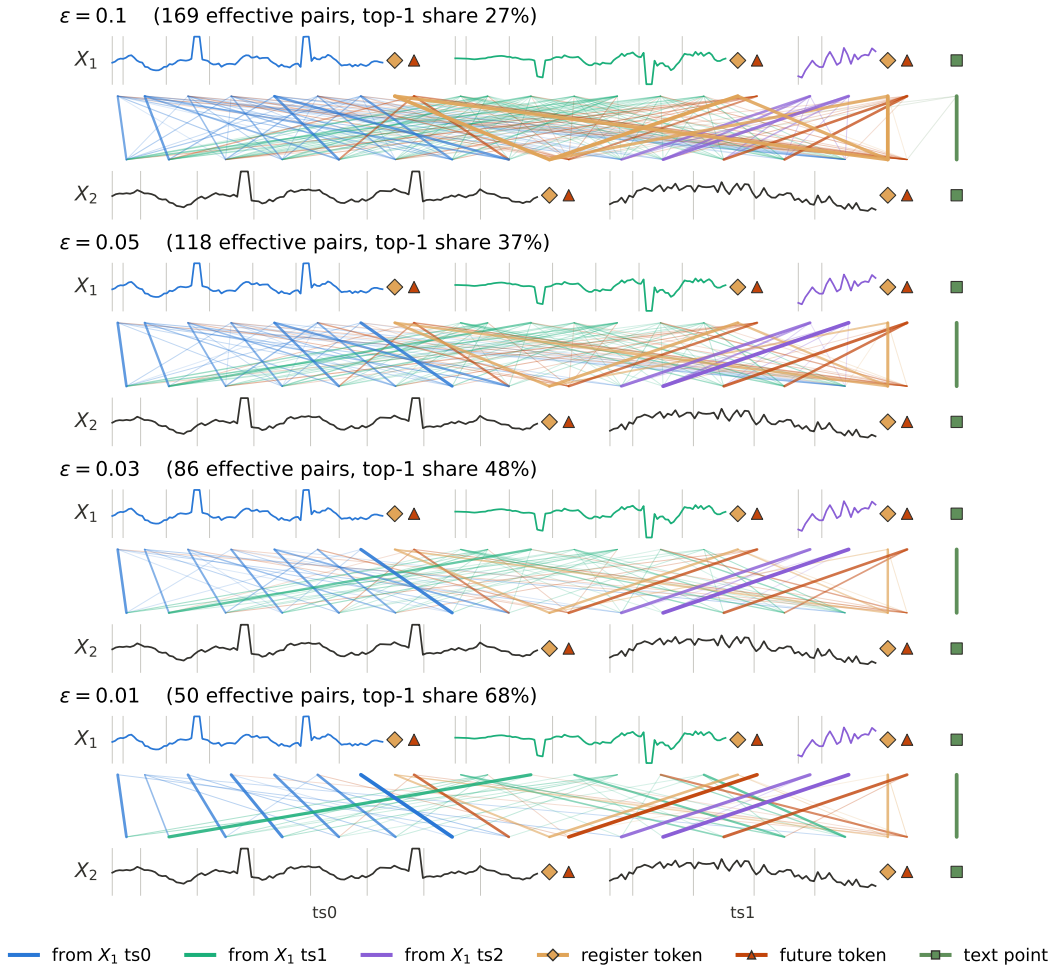


Figure 10: **Lower ε yields a sparser, more focused coupling.** Each row is the transport plan between the same two token embedding sets at a different ε, each holding the Chronos-2 tokens of every variate and the pooled text point (■, mass $\alpha = 0.5$) of Eq. equation 8. Strands show sufficiently large token-to-token mass, scaled against the largest series-to-series cell. The text point keeps essentially all of its mass within its own modality at every ε, as Proposition 1 requires of the exact plan. Lower ε concentrates mass on fewer matches. Plans use float64; $\varepsilon = 0.01$ is illustrative only.

G.3 Where OT and Pooling Retrieve Differently

Aggregate accuracy differences between OT and mean pooling (Section 6) do not show whether the demonstrations OT selects are better, because the two comparisons select the same demonstrations for many queries and the remaining differences are confounded with generation noise. We therefore compare them query by query. For each of the 117,666 (model, query) pairs of the component analysis (six TSLMs, 39 accuracy-scored subtasks) at each $k \in \{1, 4, 8\}$, we pair the TSLM-ICL retrieval with the retrieval of the mean-pooled cosine or Euclidean variant, and record (i) the *disagreement* $1 - |S_{\mathrm{OT}} \cap S_{\mathrm{pool}}|/k$ between the two sets of demonstrations shown in the prompt, and (ii) the accuracy difference $\Delta\mathrm{acc} = \mathrm{acc}_{\mathrm{OT}} - \mathrm{acc}_{\mathrm{pool}} \in \{-1, 0, 1\}$. Queries for which both variants produce the *same prompt* (identical demonstrations in identical order) serve as a control. OT and mean pooling often select different demonstrations: they produce the same prompt for 59% of queries at $k = 1$, but only for 10% at $k = 4$ and 5% at $k = 8$. *Whenever the selections differ, the demonstrations chosen by OT lead to higher accuracy on average.* This holds consistently: for univariate and multivariate

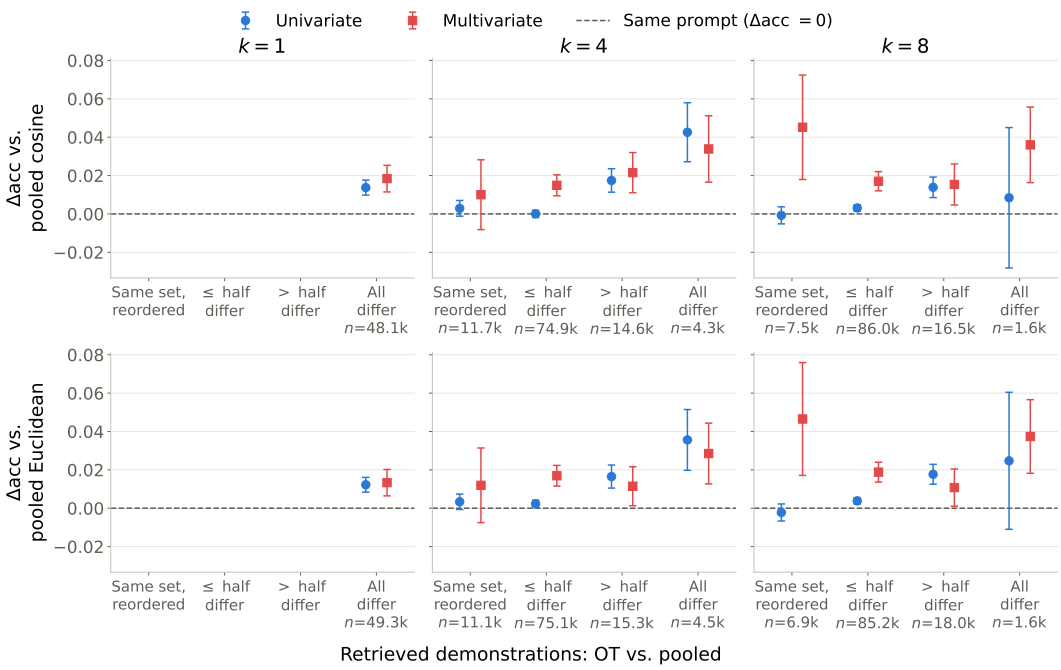


Figure 11: **When OT and pooling select different demonstrations, OT is more accurate.** Mean per-query accuracy difference $\Delta\mathrm{acc}$ between TSLM-ICL and mean pooling (top: cosine, bottom: Euclidean), binned by how much the two selections differ. The dashed line marks zero, the value expected when both select the same prompt. n is the number of (model, query) pairs per bin; at $k = 1$, only the disjoint case exists. Error bars are ± 1 SE, clustered by query.

subtasks at every budget, against both pooled cosine and pooled Euclidean; across the six TSLMs. Moreover, the more the two selections differ, the larger the advantage of OT (Figure 11).

H Extended Results

Tables 14–43 break Table 2 down to one table per model and dataset, with the dataset's subtasks as columns. Each table follows the rules of Table 2, re-applied to that model and dataset alone: within a k block all methods are scored on the same subtask cells, the *Average* column is the mean over those cells, zero-shot is a single reference row on the subtasks common to all blocks, and the best value per column within a block is in **bold**. Each caption gives the number of common subtask cells in each k block. A dagger (†) marks a method not yet run on every subtask of that model and dataset at some k; it is shown on the other methods' common cells where it has them, does not compete for bold, and has no *Average*. For Time-MQA-7B on Time-MQA and TimeOmni-1-7B on TimeOmni, zero-shot accuracy exceeds most ICL configurations because both models were trained on data from these benchmarks (Appendix E).

Paired differences. The standard errors of single-method means are nearly identical across methods (e.g. ≈ 0.008 on TSExam), because all methods are graded on the same cells and $p(1-p)$ varies little over the observed accuracies. Comparing two such intervals overstates the noise in their difference, since both methods answer the same items and an item's difficulty is shared. Table 13 therefore reports paired differences. For a cell j (model, dataset, subtask, k) with n_j items, let δ_{uj} be the score of TSLM-ICL minus that of the baseline on item u, with mean $\bar{\delta}_j$ and sample variance s_j^2. Over the J common cells of a table entry,

$$\Delta = \frac{1}{J}\sum_{j=1}^{J}\bar{\delta}_j, \qquad \mathrm{SE}(\Delta) = \frac{1}{J}\sqrt{\sum_{j=1}^{J}\frac{s_j^2}{n_j}}. \tag{39}$$

Table 13: **Paired accuracy difference, TSLM-ICL minus each baseline.** Mean over the same common cells as Table 2, so each value is the difference of two entries of that table. $\pm$ is one standard error from grading noise, paired by item within each cell. Differences within 1.96 SE of zero (not significant at the 5% level, uncorrected) are in grey.

	TSLM-ICL −	TSExam	TSR-Bench	Time-MQA	MTBench	TimeOmni	Average
$k=1$	Random	$+0.109 \pm 0.009$	$+0.093 \pm 0.004$	$+0.128 \pm 0.004$	$+0.034 \pm 0.002$	$+0.016 \pm 0.005$	$+0.071 \pm 0.002$
	Text	$+0.103 \pm 0.009$	$+0.083 \pm 0.004$	$+0.054 \pm 0.004$	$+0.015 \pm 0.002$	$+0.003 \pm 0.004$	$+0.053 \pm 0.002$
	TimeRAG	$+0.050 \pm 0.008$	$+0.041 \pm 0.004$	$+0.003 \pm 0.003$	$+0.007 \pm 0.002$	$+0.004 \pm 0.004$	$+0.025 \pm 0.002$
	TS-RAG	$+0.057 \pm 0.008$	$+0.033 \pm 0.004$	$+0.011 \pm 0.003$	$+0.006 \pm 0.002$	$+0.007 \pm 0.004$	$+0.024 \pm 0.002$
$k=4$	Random	$+0.128 \pm 0.009$	$+0.113 \pm 0.004$	$+0.159 \pm 0.004$	$+0.039 \pm 0.002$	$+0.038 \pm 0.005$	$+0.088 \pm 0.002$
	Text	$+0.127 \pm 0.009$	$+0.085 \pm 0.004$	$+0.101 \pm 0.004$	$+0.014 \pm 0.002$	$+0.021 \pm 0.005$	$+0.064 \pm 0.002$
	TimeRAG	$+0.056 \pm 0.009$	$+0.048 \pm 0.004$	$+0.023 \pm 0.003$	$+0.001 \pm 0.002$	$+0.013 \pm 0.005$	$+0.030 \pm 0.002$
	TS-RAG	$+0.057 \pm 0.008$	$+0.027 \pm 0.004$	$+0.019 \pm 0.003$	$+0.002 \pm 0.002$	$+0.001 \pm 0.005$	$+0.020 \pm 0.002$
$k=8$	Random	$+0.123 \pm 0.009$	$+0.106 \pm 0.004$	$+0.177 \pm 0.004$	$+0.037 \pm 0.002$	$+0.043 \pm 0.005$	$+0.086 \pm 0.002$
	Text	$+0.122 \pm 0.009$	$+0.083 \pm 0.004$	$+0.119 \pm 0.004$	$+0.012 \pm 0.002$	$+0.027 \pm 0.005$	$+0.064 \pm 0.002$
	TimeRAG	$+0.048 \pm 0.009$	$+0.039 \pm 0.004$	$+0.028 \pm 0.003$	$+0.000 \pm 0.002$	$+0.012 \pm 0.005$	$+0.025 \pm 0.002$
	TS-RAG	$+0.053 \pm 0.008$	$+0.024 \pm 0.004$	$+0.020 \pm 0.003$	-0.002 ± 0.002	$+0.004 \pm 0.004$	$+0.018 \pm 0.002$

Δ weights cells equally, so it is exactly the difference of the corresponding entries of Table 2. The SE treats the cells as fixed and measures item-level grading noise only, the same convention as the per-method SE. Each s_j^2 is computed on per-item differences, so the difficulty shared by the two methods cancels.

TSLM-ICL's average gain is significant against every baseline at every k, with $\mathrm{SE} = 0.002$ throughout. It is also significant on TSExam and TSR-Bench in every entry, and against Random in every entry. Of the 60 per-dataset entries, 10 are not significant, all with $|\Delta| \leq 0.007$. Nine lie on MTBench and TimeOmni, mostly against the two time-series baselines (TimeRAG and TS-RAG on MTBench at $k \in \{4, 8\}$, TS-RAG on TimeOmni at every k). The tenth is TimeRAG on Time-MQA at $k = 1$. None is significantly negative. These are the same datasets on which Table 2 shows the smallest margins.

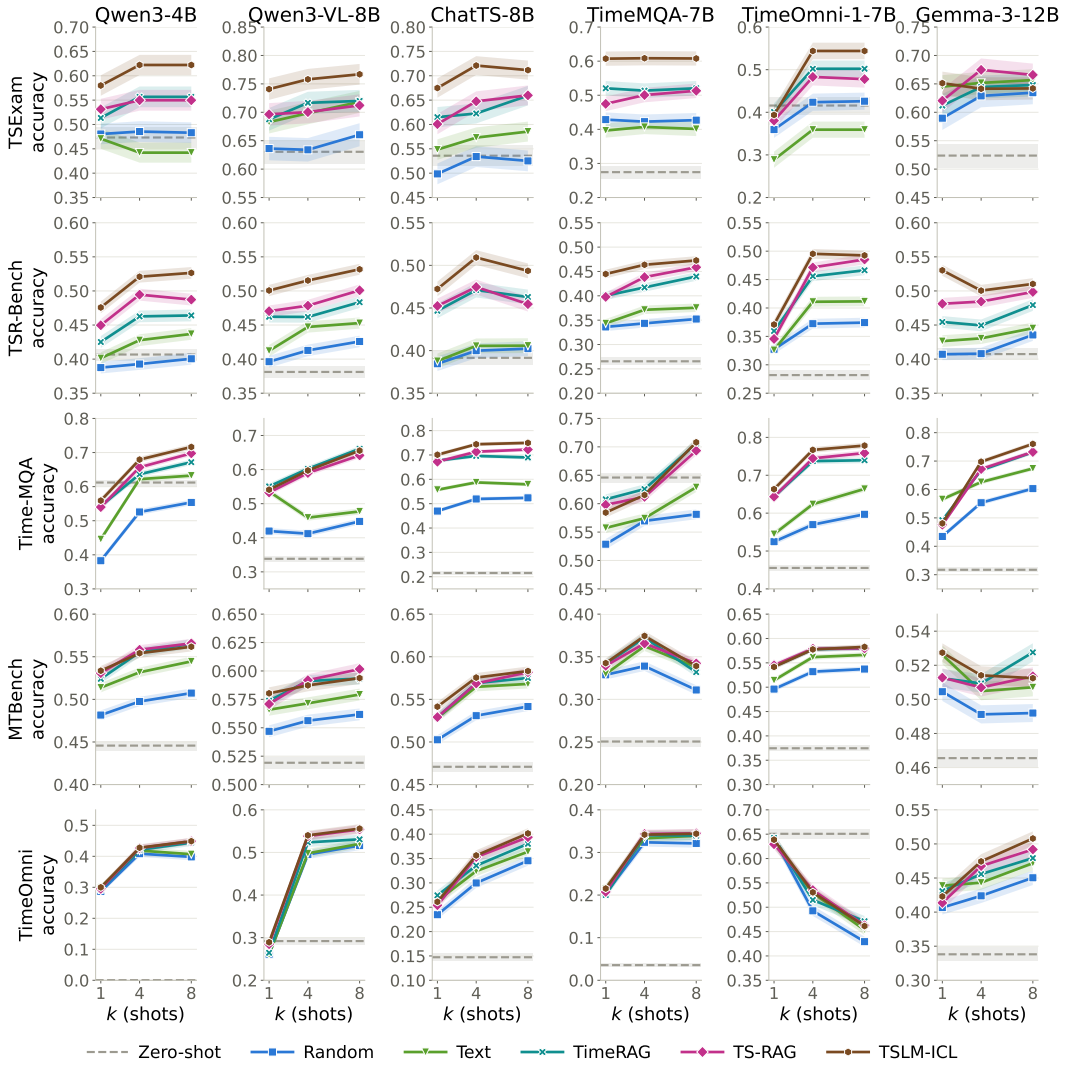


Figure 12: **Accuracy against k, per dataset.** The breakdown of Figure 3: rows are datasets and columns are models. Bands show ± 1 SE, dashed lines show matching zero-shot accuracy, and panels use separate y-ranges.

H.1 QWEN3-4B

Table 14: **Qwen3-4B on TSExam.** Common subtask cells per block: k=1: 5, k=4: 5, k=8: 5.

	Method	Pattern recognition	Noise	Similarity	Anomaly detection	Causality	Average
	Zero-shot	0.472	0.548	0.550	0.380	0.417	0.473
$k=1$	Random	0.456	0.583	**0.567**	0.380	0.417	0.480
	Text	0.489	**0.655**	0.483	0.352	0.375	0.471
	TimeRAG	0.475	0.488	0.525	**0.537**	0.542	0.513
	TS-RAG	0.528	0.559	0.542	0.528	0.500	0.531
	TSLM-ICL	**0.539**	**0.655**	0.558	**0.537**	**0.611**	**0.580**
$k=4$	Random	0.481	0.559	**0.558**	0.398	0.431	0.485
	Text	0.453	0.631	0.483	0.269	0.375	0.442
	TimeRAG	0.561	0.619	0.525	0.537	0.542	0.557
	TS-RAG	0.577	0.583	0.542	0.546	0.500	0.550
	TSLM-ICL	**0.677**	**0.691**	**0.558**	**0.574**	**0.611**	**0.622**
$k=8$	Random	0.481	0.548	**0.558**	0.398	0.431	0.483
	Text	0.453	0.631	0.483	0.269	0.375	0.442
	TimeRAG	0.561	0.619	0.525	0.537	0.542	0.557
	TS-RAG	0.577	0.583	0.542	0.546	0.500	0.550
	TSLM-ICL	**0.677**	**0.691**	**0.558**	**0.574**	**0.611**	**0.622**

Table 15: **Qwen3-4B on Time-MQA.** Common subtask cells per block: k=1: 3, k=4: 3, k=8: 3.

	Method	Anomaly detection	Activity classification	Freeze classification	Average
	Zero-shot	0.552	0.390	0.894	0.612
$k=1$	Random	0.565	0.295	0.288	0.383
	Text	0.603	0.357	0.377	0.446
	TimeRAG	0.616	0.451	0.562	0.543
	TS-RAG	0.608	0.455	0.557	0.540
	TSLM-ICL	**0.619**	**0.487**	**0.571**	**0.559**
$k=4$	Random	0.585	0.275	0.717	0.526
	Text	**0.624**	0.404	**0.838**	0.622
	TimeRAG	0.590	0.513	0.806	0.636
	TS-RAG	0.617	0.554	0.799	0.657
	TSLM-ICL	0.613	**0.619**	0.806	**0.679**
$k=8$	Random	0.602	0.310	0.749	0.554
	Text	0.645	0.391	**0.861**	0.632
	TimeRAG	0.646	0.548	0.821	0.672
	TS-RAG	**0.669**	0.611	0.813	0.698
	TSLM-ICL	0.658	**0.669**	0.822	**0.716**

Table 16: **Qwen3-4B on TSR-Bench.** Common subtask cells per block: k=1: 15, k=4: 15, k=8: 15.

	Method	Deductive	Inductive	Abductive	Causal	Etiological	Temporal relation	Pattern recognition	Anomaly detection	Noise	Similarity	Numerical	Qualitative decision	Quantitative decision	Event prediction	Forecasting	Average
	Zero-shot	0.324	0.210	0.567	0.233	0.131	0.275	0.507	0.442	0.529	0.602	0.430	0.337	0.310	0.750	0.454	0.407
$k=1$	Random	0.224	0.240	0.580	0.237	0.177	**0.300**	0.515	0.465	0.586	0.549	0.360	0.337	0.253	0.733	**0.257**	0.388
	Text	0.172	0.380	0.527	0.253	0.080	0.212	0.520	0.450	**0.655**	0.681	0.420	**0.346**	0.467	0.611	0.246	0.401
	TimeRAG	0.320	0.350	0.573	0.250	0.269	0.237	0.563	0.395	0.609	0.566	0.400	**0.346**	0.480	**0.767**	0.250	0.425
	TS-RAG	**0.332**	0.360	**0.587**	0.263	0.443	0.231	0.574	0.496	0.632	0.593	0.430	**0.346**	0.477	0.739	0.242	0.450
	TSLM-ICL	0.268	**0.390**	0.547	**0.267**	**0.451**	0.219	**0.588**	**0.550**	0.632	**0.708**	**0.453**	**0.346**	**0.730**	0.739	0.249	**0.476**
$k=4$	Random	0.228	0.260	0.580	0.213	0.220	0.250	0.461	0.403	0.563	0.673	0.383	0.337	0.300	0.750	0.271	0.393
	Text	0.168	0.430	0.527	0.260	0.120	0.237	0.496	0.512	0.632	0.673	0.495	**0.346**	0.500	0.761	0.260	0.428
	TimeRAG	0.316	0.270	0.573	0.243	0.363	0.231	0.585	0.566	0.713	0.664	0.458	**0.346**	0.573	**0.764**	**0.275**	0.463
	TS-RAG	**0.332**	0.410	**0.587**	0.270	0.540	0.231	0.612	**0.667**	0.678	0.646	0.512	**0.346**	0.570	0.744	0.272	0.495
	TSLM-ICL	0.268	**0.440**	0.547	**0.303**	**0.589**	**0.287**	**0.663**	0.605	**0.724**	**0.690**	**0.527**	**0.346**	**0.820**	0.736	0.267	**0.521**
$k=8$	Random	0.228	0.330	0.580	0.243	0.237	0.244	0.466	0.419	0.529	0.646	0.372	**0.346**	0.320	0.764	**0.283**	0.401
	Text	0.168	0.400	0.527	0.263	0.163	**0.263**	0.491	0.519	**0.736**	0.664	0.512	**0.346**	0.490	0.750	0.263	0.437
	TimeRAG	0.316	0.360	0.573	0.243	0.346	0.231	0.577	0.496	0.667	0.664	0.502	**0.346**	0.610	0.761	0.268	0.464
	TS-RAG	**0.332**	0.350	**0.587**	0.260	0.531	0.175	**0.644**	0.550	0.667	0.708	0.537	**0.346**	0.567	0.772	0.282	0.487
	TSLM-ICL	0.268	**0.430**	0.547	**0.290**	**0.606**	**0.263**	0.615	**0.636**	0.701	**0.726**	**0.570**	**0.346**	**0.830**	**0.797**	0.274	**0.526**

Table 17: **Qwen3-4B on MTBench.** Common subtask cells per block: k=1: 10, k=4: 10, k=8: 10.

	Method	Finance trend (short)	Finance trend (long)	Finance corr. (short)	Finance corr. (long)	Finance QA (short)	Finance QA (long)	Weather trend (past)	Weather trend (future)	Weather QA (short)	Weather QA (long)	Average
	Zero-shot	0.288	0.301	0.360	0.296	0.882	0.845	0.304	0.213	0.474	0.493	0.446
$k=1$	Random	0.536	0.297	0.310	**0.283**	0.862	**0.845**	0.377	0.329	0.473	0.505	0.482
	Text	**0.604**	**0.419**	0.295	**0.283**	0.825	0.826	0.447	0.420	**0.504**	0.515	0.514
	TimeRAG	0.565	0.326	**0.322**	0.252	0.866	0.837	**0.672**	0.460	0.457	0.483	0.524
	TS-RAG	0.543	0.366	0.320	0.262	0.864	0.826	0.668	0.484	0.491	0.485	0.531
	TSLM-ICL	0.589	0.362	**0.322**	0.252	**0.868**	0.810	0.582	**0.523**	0.501	**0.527**	**0.534**
$k=4$	Random	0.495	0.364	0.301	0.262	0.888	0.855	0.397	0.418	0.501	0.496	0.498
	Text	**0.561**	**0.425**	0.314	**0.291**	0.870	0.864	0.475	0.469	0.501	**0.548**	0.532
	TimeRAG	0.528	0.381	0.324	0.271	0.888	0.876	0.735	0.514	0.512	0.525	0.555
	TS-RAG	0.476	0.358	0.332	0.275	**0.896**	**0.882**	**0.754**	**0.564**	**0.524**	0.521	**0.558**
	TSLM-ICL	0.537	0.383	**0.352**	0.287	0.880	0.861	0.633	0.545	0.522	0.541	0.554
$k=8$	Random	0.468	0.362	0.301	0.269	0.892	0.862	0.427	0.443	0.530	0.518	0.507
	Text	**0.561**	**0.432**	0.316	0.298	0.876	0.868	0.505	0.505	0.533	0.551	0.545
	TimeRAG	0.515	0.385	0.342	0.279	0.894	0.878	0.749	0.542	0.530	0.538	0.565
	TS-RAG	0.464	0.362	0.340	0.262	**0.902**	**0.893**	**0.770**	**0.587**	0.535	0.541	**0.566**
	TSLM-ICL	0.524	0.366	**0.344**	**0.300**	0.878	0.884	0.644	0.565	**0.549**	**0.565**	0.562

Table 18: **Qwen3-4B on TimeOmni.** Common subtask cells per block: k=1: 6, k=4: 6, k=8: 6. *Zero-shot on this dataset is limited by output format as much as by reasoning:* TimeOmni's official scorer credits an answer only inside `<answer>` tags, and at k=0 this model emits them on 0.2% of items (71%–93% with demonstrations), so answers given in prose score 0. Part of the gain from k=0 to k>0 is therefore format compliance rather than accuracy.

	Method	Scenario (ID)	Causality (ID)	Decision (ID)	Scenario (OOD)	Causality (OOD)	Decision (OOD)	Average
	Zero-shot	0.000	0.000	0.000	0.002	0.001	0.000	0.001
$k=1$	Random	0.470	0.310	0.021	0.487	**0.321**	0.110	0.287
	Text	**0.500**	0.296	0.043	0.486	0.296	**0.176**	**0.300**
	TimeRAG	0.485	0.321	**0.059**	0.484	0.290	0.117	0.293
	TS-RAG	0.475	**0.344**	0.053	0.472	0.318	0.110	0.295
	TSLM-ICL	0.480	0.341	0.048	**0.490**	0.319	0.121	**0.300**
$k=4$	Random	**0.535**	0.321	0.308	**0.507**	0.336	0.443	0.409
	Text	0.495	0.328	0.325	0.479	0.335	**0.549**	0.418
	TimeRAG	0.505	0.331	**0.367**	0.477	0.329	0.516	0.421
	TS-RAG	0.470	0.362	**0.367**	0.478	**0.388**	0.494	0.427
	TSLM-ICL	0.485	**0.366**	0.362	0.483	0.376	0.494	**0.428**
$k=8$	Random	0.385	0.340	0.356	0.393	0.344	0.571	0.398
	Text	0.425	0.375	0.335	0.392	0.318	0.597	0.407
	TimeRAG	**0.485**	0.362	0.404	0.439	0.362	0.612	0.444
	TS-RAG	0.445	0.374	0.425	0.435	**0.398**	**0.615**	**0.449**
	TSLM-ICL	0.420	**0.404**	**0.431**	**0.452**	0.383	0.604	**0.449**

H.2 QWEN3-VL-8B

Table 19: **Qwen3-VL-8B on TSExam.** Common subtask cells per block: k=1: 5, k=4: 5, k=8: 5.

	Method	Pattern recognition	Noise	Similarity	Anomaly detection	Causality	Average
	Zero-shot	0.663	0.643	0.717	0.574	0.556	0.630
$k=1$	Random	0.657	0.643	0.700	0.583	0.597	0.636
	Text	0.727	0.691	0.692	0.694	**0.611**	0.683
	TimeRAG	0.729	0.655	0.725	0.787	0.542	0.688
	TS-RAG	0.710	0.679	0.742	0.768	0.583	0.696
	TSLM-ICL	**0.782**	**0.750**	**0.783**	**0.806**	0.583	**0.741**
$k=4$	Random	0.657	0.643	0.717	0.583	0.569	0.634
	Text	0.751	0.762	0.775	0.676	0.528	0.698
	TimeRAG	0.754	0.714	0.758	0.759	**0.597**	0.717
	TS-RAG	0.765	0.714	0.717	0.750	0.556	0.700
	TSLM-ICL	**0.796**	**0.821**	**0.817**	**0.787**	0.569	**0.758**
$k=8$	Random	0.668	0.702	0.700	0.620	**0.611**	0.660
	Text	0.776	0.762	**0.808**	0.676	0.569	0.718
	TimeRAG	0.773	0.702	0.758	0.768	0.597	0.720
	TS-RAG	0.765	0.714	0.725	0.787	0.569	0.712
	TSLM-ICL	**0.804**	**0.821**	0.783	**0.815**	**0.611**	**0.767**

Table 20: **Qwen3-VL-8B on TimeMQA.** Common subtask cells per block: k=1: 3, k=4: 3, k=8: 3.

	Method	Anomaly detection	Activity classification	Freeze classification	Average
	Zero-shot	0.515	0.390	0.110	0.338
$k=1$	Random	0.524	0.318	0.417	0.420
	Text	0.546	**0.426**	0.635	0.536
	TimeRAG	0.586	0.378	**0.688**	**0.551**
	TS-RAG	**0.589**	0.366	0.643	0.533
	TSLM-ICL	0.586	0.390	0.647	0.541
$k=4$	Random	0.532	0.320	0.384	0.412
	Text	0.586	0.386	0.405	0.459
	TimeRAG	**0.674**	0.423	**0.710**	**0.602**
	TS-RAG	0.663	0.439	0.668	0.590
	TSLM-ICL	0.670	**0.458**	0.664	0.597
$k=8$	Random	0.532	0.355	0.457	0.448
	Text	0.587	0.406	0.439	0.477
	TimeRAG	**0.742**	0.489	**0.752**	**0.661**
	TS-RAG	0.701	0.495	0.729	0.642
	TSLM-ICL	0.709	**0.539**	0.719	0.656

Table 21: **Qwen3-VL-8B on TSR-Bench.** Common subtask cells per block: k=1: 15, k=4: 15, k=8: 15.

	Method	Deductive	Inductive	Abductive	Causal	Etiological	Temporal relation	Pattern recognition	Anomaly detection	Noise	Similarity	Numerical	Qualitative decision	Quantitative decision	Event prediction	Forecasting	Average
	Zero-shot	0.284	0.160	0.533	0.347	0.197	0.306	0.577	0.667	0.506	0.646	0.198	0.304	0.343	0.236	0.414	0.381
$k=1$	Random	0.220	0.250	0.500	0.297	0.214	0.237	0.580	0.612	0.540	0.699	0.185	0.272	0.380	0.606	0.351	0.396
	Text	0.180	0.270	0.560	0.280	0.257	**0.312**	0.582	0.589	0.644	0.708	0.203	0.242	0.450	0.519	0.386	0.412
	TimeRAG	**0.292**	0.250	0.600	0.233	0.391	0.281	0.658	0.744	0.690	0.681	0.215	0.278	0.537	0.664	0.418	0.462
	TS-RAG	0.272	0.270	**0.627**	0.273	0.557	0.250	0.650	**0.760**	0.690	**0.743**	0.158	**0.298**	0.520	0.664	0.324	0.470
	TSLM-ICL	0.232	**0.280**	0.533	**0.343**	**0.611**	0.175	**0.717**	0.736	**0.724**	**0.743**	**0.253**	0.295	**0.747**	**0.672**	**0.447**	**0.501**
$k=4$	Random	0.236	0.190	0.500	0.333	0.286	0.263	0.590	0.605	0.598	0.735	0.188	0.272	0.390	0.600	0.408	0.413
	Text	0.176	**0.310**	0.560	0.333	0.169	**0.338**	0.666	0.705	0.713	0.796	0.195	0.242	0.507	0.517	**0.482**	0.447
	TimeRAG	0.316	0.230	0.600	0.287	0.400	0.225	0.625	0.698	0.586	0.788	0.207	0.278	0.687	0.567	0.435	0.462
	TS-RAG	**0.320**	0.260	**0.627**	0.327	0.571	0.231	0.615	0.698	0.609	0.761	0.245	**0.298**	0.627	0.578	0.410	0.478
	TSLM-ICL	0.256	0.260	0.533	**0.360**	**0.629**	0.219	**0.682**	**0.760**	**0.736**	**0.832**	**0.258**	0.295	**0.840**	**0.606**	0.465	**0.515**
$k=8$	Random	0.236	0.190	0.500	0.350	0.289	**0.275**	0.593	0.682	0.598	0.761	0.253	0.272	0.383	0.603	0.404	0.426
	Text	0.176	0.270	0.560	0.370	0.174	0.256	**0.701**	0.690	0.782	0.796	0.307	0.242	0.537	0.458	0.474	0.453
	TimeRAG	0.316	0.290	0.600	0.353	0.363	0.244	0.639	0.682	0.655	0.796	0.310	0.278	0.670	0.586	0.469	0.483
	TS-RAG	**0.320**	**0.360**	**0.627**	0.330	0.546	0.263	0.644	0.736	0.621	0.770	0.340	**0.298**	0.630	0.622	0.407	0.501
	TSLM-ICL	0.256	0.290	0.533	**0.380**	**0.566**	0.263	0.685	**0.798**	**0.839**	**0.823**	**0.362**	0.275	**0.797**	**0.631**	**0.479**	**0.532**

Table 22: **Qwen3-VL-8B on MTBench.** Common subtask cells per block: k=1: 10, k=4: 10, k=8: 10.

	Method	Finance trend (short)	Finance trend (long)	Finance corr. (short)	Finance corr. (long)	Finance QA (short)	Finance QA (long)	Weather trend (past)	Weather trend (future)	Weather QA (short)	Weather QA (long)	Average
	Zero-shot	0.288	0.293	0.348	0.300	0.935	0.944	0.548	0.370	0.581	0.584	0.519
$k=1$	Random	0.507	0.360	0.332	0.302	0.949	0.959	0.522	0.386	0.589	0.562	0.547
	Text	**0.547**	**0.408**	0.330	**0.312**	0.953	0.957	0.557	0.418	**0.595**	0.582	0.566
	TimeRAG	0.503	0.350	**0.354**	0.304	0.951	0.967	**0.656**	0.481	0.594	0.577	0.574
	TS-RAG	0.496	0.375	0.330	0.302	0.953	0.959	0.650	0.484	0.578	0.581	0.571
	TSLM-ICL	0.540	0.387	0.340	0.306	**0.957**	**0.969**	0.630	**0.502**	0.588	**0.584**	**0.580**
$k=4$	Random	0.503	0.375	0.350	0.298	0.955	0.959	0.562	0.378	0.595	0.585	0.556
	Text	0.520	0.392	0.338	**0.330**	0.961	0.963	0.583	0.413	0.617	0.598	0.572
	TimeRAG	**0.557**	0.377	0.354	0.295	0.939	0.963	**0.705**	0.475	**0.642**	0.604	0.591
	TS-RAG	0.497	0.385	0.340	0.316	0.959	**0.965**	0.702	**0.518**	0.614	**0.622**	**0.592**
	TSLM-ICL	0.521	**0.400**	**0.356**	0.293	**0.967**	0.959	0.638	0.496	0.631	0.611	0.587
$k=8$	Random	0.504	0.391	0.346	0.310	0.963	**0.971**	0.543	0.384	0.612	0.593	0.562
	Text	**0.527**	0.396	0.344	0.318	0.961	0.967	0.587	0.453	0.612	0.628	0.579
	TimeRAG	0.509	0.392	0.346	0.308	0.933	0.963	**0.720**	0.503	**0.643**	0.615	0.593
	TS-RAG	0.507	**0.415**	0.340	**0.324**	0.955	0.965	0.715	**0.543**	0.612	**0.640**	**0.602**
	TSLM-ICL	0.509	0.398	**0.360**	0.310	**0.967**	0.959	0.638	0.528	0.636	0.630	0.594

Table 23: **Qwen3-VL-8B on TimeOmni.** Common subtask cells per block: k=1: 6, k=4: 6, k=8: 6.

	Method	Scenario (ID)	Causality (ID)	Decision (ID)	Scenario (OOD)	Causality (OOD)	Decision (OOD)	Average
	Zero-shot	0.625	0.188	0.021	0.710	0.194	0.015	0.292
$k=1$	Random	0.590	0.152	**0.005**	0.611	0.186	**0.022**	0.261
	Text	0.595	0.189	**0.005**	0.582	0.175	0.015	0.260
	TimeRAG	0.605	0.204	0.000	0.594	0.177	0.007	0.265
	TS-RAG	**0.620**	0.235	**0.005**	0.624	0.220	0.004	0.285
	TSLM-ICL	**0.620**	**0.242**	0.000	**0.642**	**0.224**	0.007	**0.289**
$k=4$	Random	**0.660**	0.326	0.394	0.635	0.347	0.608	0.495
	Text	0.650	0.321	0.383	0.631	0.344	0.659	0.498
	TimeRAG	0.610	0.351	0.521	0.626	**0.366**	0.667	0.524
	TS-RAG	0.630	**0.374**	**0.543**	0.664	0.338	**0.681**	0.538
	TSLM-ICL	0.645	0.361	0.511	**0.685**	0.361	0.678	**0.540**
$k=8$	Random	0.620	0.339	0.457	0.646	0.335	0.700	0.516
	Text	0.655	0.334	0.410	0.658	0.346	**0.718**	0.520
	TimeRAG	0.610	0.370	0.489	0.640	**0.376**	0.700	0.531
	TS-RAG	**0.675**	**0.384**	0.548	**0.674**	0.355	0.689	0.554
	TSLM-ICL	0.660	0.364	**0.569**	0.671	0.357	0.714	**0.556**

H.3 CHATTS-8B

Table 24: **ChatTS-8B on TSExam.** Common subtask cells per block: k=1: 5, k=4: 5, k=8: 5.

	Method	Pattern recognition	Noise	Similarity	Anomaly detection	Causality	Average
	Zero-shot	0.514	0.595	0.658	0.426	0.486	0.536
$k=1$	Random	0.544	0.476	0.583	0.417	0.472	0.499
	Text	0.508	0.536	0.675	0.426	0.597	0.548
	TimeRAG	0.597	**0.631**	0.667	0.528	0.653	0.615
	TS-RAG	0.608	0.536	0.675	0.518	0.667	0.601
	TSLM-ICL	**0.682**	**0.631**	**0.825**	**0.556**	**0.681**	**0.675**
$k=4$	Random	0.533	0.559	0.550	0.500	0.528	0.534
	Text	0.544	0.631	0.625	0.454	0.611	0.573
	TimeRAG	0.655	0.607	0.592	0.537	0.722	0.623
	TS-RAG	0.685	0.607	0.617	0.620	0.708	0.648
	TSLM-ICL	**0.754**	**0.750**	**0.717**	**0.648**	**0.736**	**0.721**
$k=8$	Random	0.514	0.559	0.608	0.472	0.472	0.525
	Text	0.566	**0.786**	0.575	0.472	0.528	0.585
	TimeRAG	0.677	0.643	0.633	0.593	**0.750**	0.659
	TS-RAG	0.693	0.655	0.650	0.593	0.708	0.660
	TSLM-ICL	**0.771**	0.738	**0.767**	**0.657**	0.625	**0.712**

Table 25: **ChatTS-8B on Time-MQA.** Common subtask cells per block: k=1: 3, k=4: 3, k=8: 3.

	Method	Anomaly detection	Activity classification	Freeze classification	Average
	Zero-shot	0.495	0.043	0.108	0.215
$k=1$	Random	0.511	0.204	0.694	0.470
	Text	0.538	0.280	0.853	0.557
	TimeRAG	**0.635**	0.530	0.865	0.677
	TS-RAG	0.610	0.553	0.854	0.672
	TSLM-ICL	0.622	**0.610**	**0.871**	**0.701**
$k=4$	Random	0.555	0.220	0.783	0.519
	Text	0.572	0.299	**0.891**	0.587
	TimeRAG	0.649	0.569	0.870	0.696
	TS-RAG	0.661	0.612	0.867	0.713
	TSLM-ICL	**0.677**	**0.682**	0.873	**0.744**
$k=8$	Random	0.546	0.229	0.797	0.524
	Text	0.595	0.269	0.874	0.579
	TimeRAG	0.649	0.546	0.874	0.690
	TS-RAG	0.672	0.614	**0.881**	0.722
	TSLM-ICL	**0.696**	**0.678**	0.875	**0.750**

Table 26: **ChatTS-8B on TSR-Bench.** Common subtask cells per block: k=1: 15, k=4: 15, k=8: 15.

	Method	Deductive	Inductive	Abductive	Causal	Etiological	Temporal relation	Pattern recognition	Anomaly detection	Noise	Similarity	Numerical	Qualitative decision	Quantitative decision	Event prediction	Forecasting	Average
	Zero-shot	0.316	0.280	0.713	0.313	0.194	0.269	0.491	0.380	0.506	0.575	0.305	0.227	0.230	0.703	0.371	0.391
$k=1$	Random	0.220	0.270	0.593	0.283	0.191	0.256	0.523	0.488	0.540	0.664	0.350	0.179	0.243	0.711	0.256	0.385
	Text	0.184	0.410	0.647	0.280	0.146	0.256	0.512	0.527	0.506	0.566	**0.458**	0.194	0.357	0.500	**0.275**	0.388
	TimeRAG	0.308	**0.430**	0.673	0.220	0.366	**0.269**	0.553	0.612	**0.621**	0.637	0.398	0.164	0.453	0.728	0.269	0.447
	TS-RAG	**0.332**	0.330	0.573	0.263	0.469	0.237	0.585	**0.705**	0.575	**0.681**	**0.458**	**0.221**	0.387	**0.731**	0.239	0.452
	TSLM-ICL	0.256	0.400	**0.680**	**0.290**	**0.529**	0.231	**0.633**	0.698	0.575	0.646	0.450	0.161	**0.573**	0.697	0.265	**0.472**
$k=4$	Random	0.212	0.320	0.593	0.277	0.254	0.275	0.517	0.473	0.621	0.664	0.385	0.188	0.330	0.622	0.267	0.400
	Text	0.192	0.330	0.647	**0.353**	0.146	0.263	0.523	0.504	0.621	0.584	**0.505**	**0.203**	0.487	0.439	0.287	0.406
	TimeRAG	**0.320**	0.380	0.673	0.317	0.369	0.256	0.612	0.698	**0.644**	0.673	0.427	0.170	0.620	0.622	0.285	0.471
	TS-RAG	0.288	0.390	0.573	0.310	0.486	0.275	0.596	**0.744**	**0.644**	0.646	0.480	0.197	0.577	**0.658**	0.256	0.475
	TSLM-ICL	0.240	**0.470**	**0.680**	0.343	**0.551**	**0.306**	**0.628**	0.651	**0.644**	**0.708**	0.500	0.158	**0.817**	0.644	**0.297**	**0.509**
$k=8$	Random	0.212	0.280	0.593	0.277	0.209	**0.300**	0.526	0.543	0.575	0.664	0.435	**0.224**	0.373	0.561	0.264	0.402
	Text	0.192	0.330	0.647	**0.373**	0.137	0.244	0.534	0.512	0.575	0.531	**0.578**	0.197	0.457	0.503	0.279	0.406
	TimeRAG	**0.320**	0.420	0.673	0.310	0.334	0.219	0.593	0.674	**0.667**	0.611	0.485	0.185	0.647	0.525	0.281	0.463
	TS-RAG	0.288	0.360	0.573	0.310	0.446	0.206	**0.620**	**0.690**	0.609	0.611	0.555	0.173	0.520	**0.606**	0.251	0.455
	TSLM-ICL	0.240	**0.480**	**0.680**	0.353	**0.534**	0.250	0.596	0.612	0.598	**0.708**	0.555	0.161	**0.733**	0.603	**0.300**	**0.494**

Table 27: **ChatTS-8B on MTBench.** Common subtask cells per block: k=1: 10, k=4: 10, k=8: 10.

	Method	Finance trend (short)	Finance trend (long)	Finance corr. (short)	Finance corr. (long)	Finance QA (short)	Finance QA (long)	Weather trend (past)	Weather trend (future)	Weather QA (short)	Weather QA (long)	Average
	Zero-shot	0.227	0.293	0.271	0.256	0.876	0.828	0.446	0.413	0.555	0.545	0.471
$k=1$	Random	0.477	0.383	0.293	0.271	0.912	0.864	0.379	0.245	**0.606**	0.593	0.503
	Text	0.544	**0.446**	0.301	**0.308**	0.916	0.843	0.444	0.283	**0.606**	0.591	0.528
	TimeRAG	0.476	0.343	0.330	0.264	0.888	**0.880**	0.596	0.337	**0.606**	0.569	0.529
	TS-RAG	0.469	0.377	0.312	0.271	0.902	0.851	**0.609**	0.341	0.574	0.587	0.529
	TSLM-ICL	**0.549**	0.411	**0.334**	0.281	**0.921**	0.870	0.529	**0.346**	0.571	**0.603**	**0.541**
$k=4$	Random	0.427	0.368	0.312	0.273	0.933	0.889	0.503	0.447	0.577	0.581	0.531
	Text	**0.531**	**0.417**	0.301	**0.316**	0.935	**0.922**	0.562	0.456	**0.609**	**0.598**	0.565
	TimeRAG	0.481	0.368	**0.334**	0.250	**0.939**	0.907	0.714	0.503	0.605	0.591	0.569
	TS-RAG	0.429	0.366	0.314	0.293	0.925	0.907	**0.751**	**0.521**	0.588	0.593	0.569
	TSLM-ICL	0.504	0.398	**0.334**	0.306	0.937	**0.922**	0.650	0.508	0.602	0.595	**0.576**
$k=8$	Random	0.519	0.381	0.307	0.275	0.933	0.909	0.503	0.400	0.597	0.592	0.542
	Text	0.517	**0.444**	0.297	**0.324**	0.935	0.924	0.581	0.450	0.602	**0.607**	0.568
	TimeRAG	**0.531**	0.366	0.295	0.258	**0.943**	0.909	0.738	0.489	**0.619**	0.604	0.575
	TS-RAG	0.491	0.387	0.330	0.293	0.929	0.913	**0.756**	**0.522**	0.592	0.606	0.582
	TSLM-ICL	0.508	0.417	**0.354**	0.306	0.941	**0.928**	0.665	0.520	0.592	0.600	**0.583**

Table 28: **ChatTS-8B on TimeOmni.** Common subtask cells per block: k=1: 6, k=4: 6, k=8: 6. *Zero-shot on this dataset is limited by output format as much as by reasoning:* TimeOmni's official scorer credits an answer only inside `<answer>` tags, and at k=0 this model emits them on 41.9% of items (59%–64% with demonstrations), so answers given in prose score 0. Part of the gain from k=0 to k>0 is therefore format compliance rather than accuracy.

	Method	Scenario (ID)	Causality (ID)	Decision (ID)	Scenario (OOD)	Causality (OOD)	Decision (OOD)	Average
	Zero-shot	0.330	0.069	0.064	0.294	0.080	0.048	0.147
$k=1$	Random	**0.395**	0.046	0.213	0.396	0.065	0.293	0.235
	Text	0.390	0.065	0.239	0.376	0.074	**0.436**	0.263
	TimeRAG	0.355	0.074	**0.325**	0.408	0.075	0.410	**0.274**
	TS-RAG	0.355	0.074	0.261	0.402	0.064	0.370	0.254
	TSLM-ICL	0.380	**0.079**	0.245	**0.409**	**0.087**	0.366	0.261
$k=4$	Random	0.450	0.126	0.261	0.457	0.128	0.377	0.300
	Text	**0.465**	0.140	0.266	0.469	0.142	0.462	0.324
	TimeRAG	0.445	0.169	0.314	0.443	0.170	**0.472**	0.335
	TS-RAG	0.430	0.201	**0.404**	0.477	0.180	0.425	0.353
	TSLM-ICL	0.410	**0.231**	0.346	**0.478**	**0.215**	0.458	**0.356**
$k=8$	Random	0.465	0.204	0.223	0.508	0.240	0.432	0.345
	Text	**0.530**	0.226	0.239	0.503	0.239	**0.447**	0.364
	TimeRAG	0.510	0.265	0.266	0.511	0.301	0.429	0.380
	TS-RAG	0.515	0.279	**0.351**	0.522	0.305	0.392	0.394
	TSLM-ICL	**0.530**	**0.294**	0.335	**0.536**	**0.307**	0.407	**0.402**

H.4 TIMEMQA-7B

Table 29: **TimeMQA-7B on TSExam.** Common subtask cells per block: k=1: 5, k=4: 5, k=8: 5.

	Method	Pattern recognition	Noise	Similarity	Anomaly detection	Causality	Average
	Zero-shot	0.367	0.321	0.308	0.194	0.181	0.274
$k=1$	Random	0.425	0.464	0.475	0.417	0.361	0.428
	Text	0.398	0.452	0.433	0.296	0.403	0.397
	TimeRAG	0.536	0.559	0.442	0.481	0.583	0.520
	TS-RAG	0.514	0.441	0.475	0.528	0.417	0.475
	TSLM-ICL	**0.638**	**0.571**	**0.642**	**0.546**	**0.639**	**0.607**
$k=4$	Random	0.442	0.464	0.475	0.370	0.361	0.423
	Text	0.392	0.476	0.417	0.333	0.417	0.407
	TimeRAG	0.541	0.524	0.433	0.472	0.597	0.514
	TS-RAG	0.536	0.488	0.483	**0.593**	0.403	0.501
	TSLM-ICL	**0.602**	**0.607**	**0.667**	0.556	**0.611**	**0.609**
$k=8$	Random	0.453	0.488	0.467	0.352	0.375	0.427
	Text	0.392	0.452	0.425	0.306	0.431	0.401
	TimeRAG	0.536	0.524	0.450	0.509	0.583	0.520
	TS-RAG	0.547	0.500	0.500	**0.574**	0.444	0.513
	TSLM-ICL	**0.591**	**0.595**	**0.658**	0.556	**0.639**	**0.608**

Table 30: **TimeMQA-7B on TimeMQA.** Common subtask cells per block: k=1: 3, k=4: 3, k=8: 3.

	Method	Anomaly detection	Activity classification	Freeze classification	Average
	Zero-shot	0.571	0.477	0.890	0.646
$k=1$	Random	0.573	**0.529**	0.484	0.529
	Text	**0.704**	0.485	0.483	0.557
	TimeRAG	0.552	0.522	**0.749**	**0.608**
	TS-RAG	0.562	0.485	0.747	0.598
	TSLM-ICL	0.566	0.459	0.728	0.584
$k=4$	Random	0.591	0.447	0.671	0.570
	Text	**0.609**	0.358	0.756	0.574
	TimeRAG	0.578	**0.478**	**0.821**	**0.626**
	TS-RAG	0.575	0.462	0.799	0.612
	TSLM-ICL	0.600	0.458	0.788	0.615
$k=8$	Random	0.582	0.416	0.746	0.581
	Text	0.686	0.452	0.749	0.629
	TimeRAG	**0.724**	0.541	**0.852**	0.706
	TS-RAG	0.693	0.572	0.815	0.693
	TSLM-ICL	0.708	**0.585**	0.831	**0.708**

Table 31: **TimeMQA-7B on TSR-Bench.** Common subtask cells per block: k=1: 15, k=4: 15, k=8: 15.

	Method	Deductive	Inductive	Abductive	Causal	Etiological	Temporal relation	Pattern recognition	Anomaly detection	Noise	Similarity	Numerical	Qualitative decision	Quantitative decision	Event prediction	Forecasting	Average
	Zero-shot	0.220	0.180	0.453	0.203	0.189	0.188	0.288	0.147	0.483	0.416	0.142	0.325	0.187	0.311	0.250	0.266
$k=1$	Random	0.204	0.250	0.413	0.240	0.269	0.219	0.420	0.357	0.460	0.469	0.263	**0.352**	0.243	0.625	0.258	0.336
	Text	0.172	**0.350**	0.447	0.220	0.243	**0.256**	0.434	0.442	0.437	0.443	0.318	0.313	0.423	0.406	0.250	0.344
	TimeRAG	**0.324**	0.330	0.467	0.183	0.331	0.250	0.544	0.504	**0.575**	**0.584**	0.250	0.319	0.470	**0.636**	0.235	0.400
	TS-RAG	**0.324**	0.310	0.387	0.227	0.437	0.237	0.531	0.504	0.494	0.478	**0.328**	0.316	0.503	0.606	**0.285**	0.398
	TSLM-ICL	0.252	**0.350**	**0.533**	**0.253**	**0.477**	0.250	**0.631**	**0.535**	0.552	**0.584**	**0.328**	0.328	**0.750**	0.600	0.249	**0.445**
$k=4$	Random	0.192	0.350	0.373	0.200	0.209	**0.312**	0.420	0.403	0.437	0.531	0.315	0.319	0.290	0.508	0.292	0.343
	Text	0.180	0.380	0.427	0.207	0.174	0.256	0.491	0.473	0.575	0.557	**0.460**	0.322	0.347	0.436	0.285	0.371
	TimeRAG	**0.328**	0.390	0.440	0.210	0.263	0.200	0.571	0.566	0.598	0.566	0.343	0.328	0.617	0.536	0.301	0.417
	TS-RAG	0.292	0.420	**0.447**	0.200	0.337	0.275	0.598	0.620	0.552	**0.637**	0.410	0.328	0.593	**0.561**	**0.304**	0.438
	TSLM-ICL	0.252	**0.500**	0.440	**0.247**	**0.389**	0.269	**0.617**	**0.636**	**0.632**	0.557	0.445	**0.364**	**0.790**	0.531	0.287	**0.464**
$k=8$	Random	0.184	0.360	0.427	**0.203**	0.220	0.263	0.437	0.434	0.425	0.504	0.350	0.325	0.307	0.561	0.285	0.352
	Text	0.176	0.320	0.427	0.183	0.134	**0.300**	0.507	0.434	0.598	0.557	0.453	0.322	0.477	0.453	0.292	0.375
	TimeRAG	**0.332**	**0.460**	0.413	0.190	0.286	0.237	0.558	0.574	0.575	0.611	0.367	0.322	0.760	**0.614**	**0.296**	0.440
	TS-RAG	0.300	0.450	0.420	0.197	0.394	0.294	0.604	**0.651**	0.598	**0.646**	0.438	0.316	0.670	0.606	**0.296**	0.459
	TSLM-ICL	0.268	0.430	**0.440**	0.200	**0.434**	0.275	**0.628**	0.597	**0.678**	0.637	**0.455**	**0.358**	**0.813**	0.600	0.275	**0.473**

Table 32: **TimeMQA-7B on MTBench.** Common subtask cells per block: k=1: 10, k=4: 10, k=8: 10.

	Method	Finance trend (short)	Finance trend (long)	Finance corr. (short)	Finance corr. (long)	Finance QA (short)	Finance QA (long)	Weather trend (past)	Weather trend (future)	Weather QA (short)	Weather QA (long)	Average
	Zero-shot	0.248	0.215	0.204	0.194	0.279	0.382	0.132	0.182	0.340	0.331	0.251
$k=1$	Random	0.261	0.200	0.151	**0.167**	0.440	**0.392**	0.326	0.499	0.419	0.434	0.329
	Text	**0.283**	**0.227**	0.163	0.151	0.417	0.357	0.372	0.495	0.383	**0.451**	0.330
	TimeRAG	0.272	0.200	0.169	0.143	0.440	0.378	0.429	**0.519**	**0.425**	0.438	0.341
	TS-RAG	0.273	0.192	0.165	0.153	0.426	0.374	**0.458**	0.518	0.398	0.437	0.339
	TSLM-ICL	0.281	0.221	**0.194**	0.147	**0.470**	0.337	0.415	0.516	0.400	0.444	**0.343**
$k=4$	Random	0.341	0.286	0.232	**0.236**	0.399	0.424	0.359	0.510	0.285	0.316	0.339
	Text	**0.453**	**0.330**	0.269	0.231	0.415	0.434	0.389	0.507	0.301	0.297	0.363
	TimeRAG	0.397	0.318	0.251	0.221	0.411	0.434	0.518	**0.540**	**0.316**	0.319	0.372
	TS-RAG	0.357	0.326	0.253	0.207	**0.426**	0.397	**0.546**	0.528	0.301	0.316	0.366
	TSLM-ICL	0.428	0.303	**0.281**	0.227	0.422	**0.450**	0.474	0.525	0.296	**0.339**	**0.374**
$k=8$	Random	0.345	0.278	0.244	0.233	0.385	0.312	0.344	0.533	0.191	**0.246**	0.311
	Text	**0.435**	**0.333**	0.259	**0.273**	0.399	**0.360**	0.399	0.540	0.153	0.225	0.338
	TimeRAG	0.404	0.278	0.232	0.229	0.383	0.277	0.564	**0.560**	0.184	0.209	0.332
	TS-RAG	0.359	0.297	0.283	0.211	**0.417**	0.298	**0.586**	0.553	0.178	0.239	**0.342**
	TSLM-ICL	0.393	0.322	**0.285**	0.234	0.369	0.341	0.484	0.552	**0.203**	0.207	0.339

Table 33: **TimeMQA-7B on TimeOmni.** Common subtask cells per block: k=1: 6, k=4: 6, k=8: 6. *Zero-shot on this dataset is limited by output format as much as by reasoning:* TimeOmni's official scorer credits an answer only inside `<answer>` tags, and at k=0 this model emits them on 16.9% of items (75%–100% with demonstrations), so answers given in prose score 0. Part of the gain from k=0 to k>0 is therefore format compliance rather than accuracy.

	Method	Scenario (ID)	Causality (ID)	Decision (ID)	Scenario (OOD)	Causality (OOD)	Decision (OOD)	Average
	Zero-shot	0.055	0.058	0.016	0.031	0.050	0.004	0.036
$k=1$	Random	0.235	0.134	0.223	0.268	0.146	0.216	0.204
	Text	0.250	0.149	0.229	0.247	0.138	**0.278**	**0.215**
	TimeRAG	0.235	0.135	**0.250**	0.230	0.135	0.216	0.200
	TS-RAG	0.240	0.136	0.213	**0.290**	**0.152**	0.212	0.207
	TSLM-ICL	**0.290**	**0.152**	0.186	0.274	0.147	0.242	**0.215**
$k=4$	Random	0.300	0.339	0.250	0.379	0.352	0.322	0.324
	Text	**0.400**	0.331	0.207	0.367	0.352	0.341	0.333
	TimeRAG	0.375	0.340	**0.261**	**0.387**	0.350	0.330	0.340
	TS-RAG	0.340	0.360	0.229	0.369	**0.393**	**0.370**	**0.343**
	TSLM-ICL	0.340	**0.371**	0.245	0.357	0.386	0.352	0.342
$k=8$	Random	0.355	0.346	0.213	**0.387**	0.355	0.271	0.321
	Text	0.420	0.364	0.213	0.377	0.336	0.326	0.339
	TimeRAG	0.385	0.344	0.229	0.369	0.370	0.341	0.340
	TS-RAG	**0.425**	**0.376**	0.202	0.369	0.352	**0.344**	**0.345**
	TSLM-ICL	0.375	0.367	**0.250**	0.338	**0.400**	0.333	0.344

H.5 TIMEOMNI-1-7B

Table 34: **TimeOmni-1-7B on TSExam.** Common subtask cells per block: k=1: 5, k=4: 5, k=8: 5.

	Method	Pattern recognition	Noise	Similarity	Anomaly detection	Causality	Average
	Zero-shot	0.541	0.583	0.325	0.407	0.222	0.416
$k=1$	Random	0.448	0.524	0.358	0.287	0.181	0.359
	Text	0.362	0.381	0.192	0.287	**0.222**	0.289
	TimeRAG	0.483	0.524	0.358	**0.417**	**0.222**	**0.401**
	TS-RAG	0.503	0.417	**0.408**	0.380	0.194	0.380
	TSLM-ICL	**0.508**	**0.548**	0.283	0.407	**0.222**	0.394
$k=4$	Random	0.541	0.583	0.350	0.407	**0.236**	0.424
	Text	0.434	0.643	0.192	0.306	0.222	0.359
	TimeRAG	0.674	**0.702**	0.292	0.620	0.222	0.502
	TS-RAG	0.657	0.571	**0.408**	0.583	0.194	0.483
	TSLM-ICL	**0.765**	0.691	0.375	**0.667**	0.222	**0.544**
$k=8$	Random	0.541	0.595	0.350	0.407	**0.236**	0.426
	Text	0.434	0.643	0.192	0.306	0.222	0.359
	TimeRAG	0.674	**0.702**	0.292	0.620	0.222	0.502
	TS-RAG	0.652	0.559	**0.400**	0.583	0.194	0.478
	TSLM-ICL	**0.765**	0.691	0.375	**0.667**	0.222	**0.544**

Table 35: **TimeOmni-1-7B on TimeMQA.** Common subtask cells per block: k=1: 3, k=4: 3, k=8: 3.

	Method	Anomaly detection	Activity classification	Freeze classification	Average
	Zero-shot	0.521	0.085	0.760	0.455
$k=1$	Random	0.539	0.276	0.759	0.525
	Text	0.514	0.300	0.820	0.545
	TimeRAG	**0.566**	0.553	0.813	0.644
	TS-RAG	0.552	0.567	0.811	0.643
	TSLM-ICL	0.563	**0.604**	**0.823**	**0.663**
$k=4$	Random	0.649	0.285	0.775	0.570
	Text	0.702	0.356	0.813	0.624
	TimeRAG	0.764	0.581	0.867	0.737
	TS-RAG	0.762	0.614	0.858	0.745
	TSLM-ICL	**0.767**	**0.656**	**0.878**	**0.767**
$k=8$	Random	0.662	0.342	0.787	0.597
	Text	0.735	0.372	**0.885**	0.664
	TimeRAG	0.774	0.584	0.861	0.740
	TS-RAG	0.785	0.645	0.846	0.759
	TSLM-ICL	**0.786**	**0.687**	0.862	**0.778**

Table 36: **TimeOmni-1-7B on TSR-Bench.** Common subtask cells per block: k=1: 15, k=4: 15, k=8: 15.

	Method	Deductive	Inductive	Abductive	Causal	Etiological	Temporal relation	Pattern recognition	Anomaly detection	Noise	Similarity	Numerical	Qualitative decision	Quantitative decision	Event prediction	Forecasting	Average
	Zero-shot	0.156	0.190	0.267	0.123	0.206	0.125	0.472	0.240	0.448	0.496	0.092	0.298	0.197	0.672	0.246	0.282
$k=1$	Random	0.180	0.300	0.467	0.153	0.163	**0.231**	0.485	0.279	0.460	0.522	0.200	**0.298**	0.233	0.706	**0.232**	0.327
	Text	0.168	0.280	**0.513**	0.153	0.097	0.206	0.491	0.271	0.471	0.513	0.170	**0.298**	0.380	0.675	0.201	0.326
	TimeRAG	0.192	**0.340**	0.407	**0.187**	0.266	0.206	**0.526**	0.318	**0.494**	**0.628**	**0.205**	**0.298**	0.393	**0.728**	0.204	0.359
	TS-RAG	**0.244**	0.300	0.393	0.163	0.349	0.200	0.493	0.341	**0.494**	0.443	0.182	**0.298**	0.367	0.686	0.229	0.346
	TSLM-ICL	0.200	0.330	0.447	0.160	**0.391**	0.225	0.515	**0.357**	0.414	0.575	0.195	**0.298**	**0.553**	0.686	0.217	**0.371**
$k=4$	Random	0.224	0.290	0.467	0.130	0.206	**0.281**	0.542	0.380	0.552	0.611	0.297	**0.298**	0.247	0.728	0.333	0.372
	Text	0.172	0.490	**0.513**	0.160	0.140	0.244	0.542	0.434	0.667	0.593	**0.505**	**0.298**	0.503	0.556	**0.347**	0.411
	TimeRAG	0.320	0.440	0.387	0.137	0.403	0.269	0.639	0.597	0.690	**0.690**	0.378	**0.298**	0.500	**0.744**	**0.347**	0.456
	TS-RAG	**0.332**	**0.520**	0.393	0.160	0.509	0.263	**0.679**	**0.612**	0.667	0.637	0.460	**0.298**	0.510	0.719	0.311	0.471
	TSLM-ICL	0.268	0.460	0.380	**0.173**	**0.591**	**0.281**	0.658	0.574	**0.713**	0.673	0.485	**0.298**	**0.833**	0.711	0.332	**0.495**
$k=8$	Random	0.224	0.270	0.467	0.143	0.229	0.250	0.544	0.380	0.552	0.593	0.320	**0.298**	0.313	0.700	0.331	0.374
	Text	0.172	0.460	**0.513**	**0.153**	0.120	0.275	0.547	0.419	0.701	0.584	0.487	**0.298**	0.507	0.589	**0.349**	0.412
	TimeRAG	0.320	0.460	0.387	0.130	0.403	0.269	0.674	0.543	0.724	**0.708**	0.393	**0.298**	0.600	**0.742**	0.346	0.466
	TS-RAG	**0.332**	0.480	0.393	0.143	0.546	**0.281**	**0.695**	**0.620**	0.690	0.673	**0.510**	**0.298**	0.587	0.719	0.312	0.485
	TSLM-ICL	0.268	**0.490**	0.380	0.127	**0.620**	0.225	0.641	0.535	**0.736**	0.655	0.500	**0.298**	**0.853**	0.725	0.335	**0.493**

Table 37: **TimeOmni-1-7B on MTBench.** Common subtask cells per block: k=1: 10, k=4: 10, k=8: 10.

	Method	Finance trend (short)	Finance trend (long)	Finance corr. (short)	Finance corr. (long)	Finance QA (short)	Finance QA (long)	Weather trend (past)	Weather trend (future)	Weather QA (short)	Weather QA (long)	Average
	Zero-shot	0.333	0.307	0.230	0.254	0.644	0.641	0.419	0.208	0.343	0.367	0.375
$k=1$	Random	0.496	0.276	0.255	0.262	0.860	0.837	0.399	0.436	0.558	**0.582**	0.496
	Text	**0.560**	**0.347**	0.248	**0.296**	0.839	0.843	0.478	0.455	0.527	0.547	0.514
	TimeRAG	0.536	0.299	**0.287**	0.248	0.851	**0.849**	0.720	0.518	**0.563**	0.555	0.543
	TS-RAG	0.509	0.324	0.283	0.254	**0.862**	0.839	**0.743**	**0.561**	0.535	0.540	**0.545**
	TSLM-ICL	0.544	0.331	0.285	0.281	0.855	**0.849**	0.628	0.553	0.544	0.545	0.542
$k=4$	Random	0.479	0.335	0.279	0.285	0.900	0.889	0.472	0.455	0.592	0.632	0.532
	Text	**0.587**	**0.423**	0.275	**0.316**	0.900	0.889	0.528	0.470	0.594	0.636	0.562
	TimeRAG	0.540	0.354	0.307	0.269	**0.910**	**0.897**	0.742	0.525	**0.612**	0.621	0.578
	TS-RAG	0.495	0.364	0.303	0.289	0.906	0.891	**0.753**	**0.566**	0.594	0.629	**0.579**
	TSLM-ICL	0.552	0.375	**0.312**	0.291	**0.910**	0.893	0.651	0.538	0.608	**0.646**	0.578
$k=8$	Random	0.449	0.331	0.289	0.277	0.900	0.889	0.501	0.488	0.600	0.647	0.537
	Text	**0.568**	**0.400**	0.275	0.314	0.892	0.895	0.545	0.518	0.605	**0.651**	0.566
	TimeRAG	0.521	0.337	**0.305**	0.271	0.908	0.899	0.739	0.555	**0.625**	0.639	0.580
	TS-RAG	0.472	0.333	**0.305**	0.285	**0.916**	**0.903**	**0.761**	**0.576**	0.606	0.640	0.580
	TSLM-ICL	0.524	0.364	**0.305**	**0.322**	**0.916**	0.899	0.673	0.553	0.619	**0.651**	**0.583**

Table 38: **TimeOmni-1-7B on TimeOmni.** Common subtask cells per block: k=1: 6, k=4: 6, k=8: 6.

	Method	Scenario (ID)	Causality (ID)	Decision (ID)	Scenario (OOD)	Causality (OOD)	Decision (OOD)	Average
	Zero-shot	0.830	0.674	0.399	0.791	0.625	0.586	0.651
$k=1$	Random	**0.815**	**0.661**	0.367	0.745	0.614	**0.641**	0.641
	Text	0.785	0.646	0.388	0.741	0.604	0.626	0.632
	TimeRAG	0.795	**0.661**	**0.410**	**0.773**	**0.626**	0.590	**0.642**
	TS-RAG	0.765	0.631	0.404	0.740	0.603	0.634	0.629
	TSLM-ICL	0.805	0.654	0.388	0.760	0.610	0.615	0.639
$k=4$	Random	0.590	0.583	0.255	0.590	0.565	0.374	0.493
	Text	0.605	0.596	**0.330**	0.592	0.559	**0.494**	0.529
	TimeRAG	0.600	0.579	0.319	0.601	0.562	0.429	0.515
	TS-RAG	**0.615**	**0.611**	0.298	**0.627**	0.565	**0.494**	**0.535**
	TSLM-ICL	0.595	0.608	0.308	0.612	**0.570**	0.491	0.531
$k=8$	Random	**0.615**	0.394	0.245	0.580	0.399	0.344	0.429
	Text	0.585	0.388	0.271	0.590	0.379	**0.509**	0.454
	TimeRAG	0.595	0.411	**0.356**	0.602	0.405	0.458	**0.471**
	TS-RAG	0.585	0.401	0.314	**0.611**	0.390	0.476	0.463
	TSLM-ICL	0.610	**0.420**	0.287	0.604	**0.407**	0.440	0.461

H.6 GEMMA-3-12B

Table 39: **Gemma-3-12B on TSExam.** Common subtask cells per block: k=1: 5, k=4: 5, k=8: 5.

	Method	Pattern recognition	Noise	Similarity	Anomaly detection	Causality	Average
	Zero-shot	0.575	0.583	0.642	0.472	0.347	0.524
$k=1$	Random	0.577	0.702	0.650	0.574	0.444	0.590
	Text	0.652	**0.750**	**0.750**	0.574	0.500	0.645
	TimeRAG	0.619	0.619	0.733	**0.657**	0.431	0.612
	TS-RAG	0.613	0.667	0.725	0.639	0.458	0.620
	TSLM-ICL	**0.677**	**0.750**	0.658	0.630	**0.542**	**0.651**
$k=4$	Random	0.619	**0.774**	0.692	0.602	0.458	0.629
	Text	0.713	**0.774**	**0.758**	0.639	0.375	0.652
	TimeRAG	0.713	0.702	0.692	**0.685**	0.431	0.645
	TS-RAG	0.674	0.714	0.717	**0.685**	**0.583**	**0.675**
	TSLM-ICL	**0.729**	0.750	0.658	0.611	0.458	0.641
$k=8$	Random	0.649	0.750	0.708	0.565	**0.500**	0.634
	Text	0.738	**0.786**	**0.742**	**0.685**	0.333	0.657
	TimeRAG	0.732	0.714	0.692	**0.685**	0.417	0.648
	TS-RAG	0.699	0.774	0.708	0.676	0.472	**0.666**
	TSLM-ICL	**0.743**	**0.786**	0.658	0.676	0.347	0.642

Table 40: **Gemma-3-12B on Time-MQA.** Common subtask cells per block: k=1: 3, k=4: 3, k=8: 3.

	Method	Anomaly detection	Activity classification	Freeze classification	Average
	Zero-shot	0.455	0.390	0.107	0.317
$k=1$	Random	0.496	0.307	0.501	0.435
	Text	**0.580**	**0.433**	0.684	**0.566**
	TimeRAG	0.487	0.247	0.742	0.492
	TS-RAG	0.466	0.227	0.735	0.476
	TSLM-ICL	0.468	0.228	**0.746**	0.481
$k=4$	Random	0.556	0.341	0.763	0.553
	Text	0.651	0.400	0.827	0.626
	TimeRAG	0.708	0.432	0.863	0.668
	TS-RAG	0.705	0.458	0.853	0.672
	TSLM-ICL	**0.716**	**0.512**	**0.865**	**0.698**
$k=8$	Random	0.588	0.391	0.830	0.603
	Text	0.689	0.440	**0.892**	0.674
	TimeRAG	0.750	0.565	0.879	0.731
	TS-RAG	0.736	0.598	0.862	0.732
	TSLM-ICL	**0.752**	**0.649**	0.880	**0.760**

Table 41: **Gemma-3-12B on TSR-Bench.** Common subtask cells per block: k=1: 15, k=4: 15, k=8: 15.

	Method	Deductive	Inductive	Abductive	Causal	Etiological	Temporal relation	Pattern recognition	Anomaly detection	Noise	Similarity	Numerical	Qualitative decision	Quantitative decision	Event prediction	Forecasting	Average
	Zero-shot	0.280	0.360	0.480	0.433	0.274	0.231	0.631	0.589	0.621	0.628	0.318	0.307	0.340	0.222	0.396	0.407
$k=1$	Random	0.224	0.290	0.460	0.320	0.306	0.225	0.569	0.589	0.621	0.673	0.315	**0.290**	0.360	0.558	**0.308**	0.407
	Text	0.168	**0.450**	0.453	0.323	0.209	**0.294**	0.596	0.651	0.678	0.690	0.438	0.251	0.473	0.439	0.283	0.426
	TimeRAG	0.320	0.310	0.400	0.303	0.397	0.256	0.704	0.667	0.713	0.637	0.357	0.281	0.577	0.606	0.290	0.454
	TS-RAG	**0.332**	0.370	**0.533**	0.307	0.589	0.175	0.674	0.744	0.678	0.673	**0.448**	0.242	0.567	0.597	0.290	0.481
	TSLM-ICL	0.272	0.360	0.447	**0.330**	**0.677**	0.219	**0.782**	**0.822**	**0.793**	**0.770**	0.435	0.263	**0.843**	**0.642**	0.300	**0.530**
$k=4$	Random	0.232	0.260	0.460	0.330	0.337	0.294	0.590	0.558	0.632	0.664	0.375	0.269	0.337	0.453	**0.329**	0.408
	Text	0.184	0.280	0.453	0.340	0.237	0.250	0.660	**0.698**	**0.713**	**0.823**	0.435	0.254	0.473	0.364	0.293	0.430
	TimeRAG	0.312	0.220	0.400	0.320	0.411	0.312	0.623	0.690	0.667	0.708	0.438	0.275	0.610	0.481	0.276	0.449
	TS-RAG	**0.324**	0.340	**0.533**	0.347	0.511	**0.319**	0.663	**0.698**	0.621	0.708	**0.497**	**0.292**	0.617	0.497	0.299	0.484
	TSLM-ICL	0.260	**0.360**	0.447	**0.353**	**0.534**	0.275	**0.682**	0.682	**0.713**	0.814	0.468	0.287	**0.830**	**0.528**	0.275	**0.500**
$k=8$	Random	0.232	**0.330**	0.460	0.370	0.354	0.300	0.590	0.581	0.632	0.699	0.410	0.263	0.403	0.581	**0.326**	0.435
	Text	0.184	0.290	0.453	**0.373**	0.283	0.269	**0.709**	0.636	0.724	**0.805**	0.500	0.266	0.470	0.408	0.310	0.445
	TimeRAG	0.312	0.290	0.400	0.333	0.397	0.300	0.671	**0.713**	0.678	0.761	0.490	0.275	0.637	**0.661**	0.271	0.479
	TS-RAG	**0.324**	0.310	**0.533**	0.370	**0.569**	**0.306**	0.674	0.628	0.644	0.752	0.500	**0.301**	0.610	0.636	0.321	0.499
	TSLM-ICL	0.260	**0.330**	0.447	0.360	**0.569**	0.294	**0.709**	0.659	**0.770**	0.726	**0.520**	0.284	**0.833**	0.625	0.269	**0.510**

Table 42: **Gemma-3-12B on MTBench.** Common subtask cells per block: k=1: 10, k=4: 10, k=8: 10.

	Method	Finance trend (short)	Finance trend (long)	Finance corr. (short)	Finance corr. (long)	Finance QA (short)	Finance QA (long)	Weather trend (past)	Weather trend (future)	Weather QA (short)	Weather QA (long)	Average
	Zero-shot	0.171	0.238	0.342	0.285	0.876	0.847	0.307	0.485	0.526	0.578	0.466
$k=1$	Random	0.409	0.312	0.328	0.271	0.874	0.876	**0.403**	0.394	0.561	0.617	0.504
	Text	0.464	**0.379**	0.322	**0.312**	0.878	0.878	0.399	**0.452**	0.561	0.617	0.526
	TimeRAG	0.440	0.324	0.330	0.283	0.855	0.882	0.402	0.423	0.557	**0.629**	0.512
	TS-RAG	0.429	0.322	0.332	0.295	**0.884**	**0.884**	0.398	0.403	0.553	0.628	0.513
	TSLM-ICL	**0.476**	0.343	**0.354**	0.298	0.880	0.882	0.400	0.440	**0.583**	0.618	**0.527**
$k=4$	Random	0.260	0.301	0.295	0.279	0.884	0.874	0.402	0.425	0.574	0.618	0.491
	Text	0.271	**0.307**	0.346	0.295	0.884	0.893	0.416	0.441	0.569	0.626	0.505
	TimeRAG	**0.307**	0.280	0.346	0.279	0.864	0.866	**0.459**	**0.459**	**0.602**	0.632	0.509
	TS-RAG	0.277	0.284	0.314	0.308	0.888	0.893	0.452	0.431	0.585	**0.637**	0.507
	TSLM-ICL	0.289	0.274	**0.350**	**0.310**	**0.892**	**0.899**	0.450	0.458	0.588	0.630	**0.514**
$k=8$	Random	0.319	0.225	0.303	0.273	**0.894**	0.878	0.397	0.422	0.585	0.624	0.492
	Text	0.339	0.253	0.328	**0.302**	0.888	0.905	0.422	0.413	0.589	0.632	0.507
	TimeRAG	**0.435**	0.242	0.326	0.281	0.890	0.876	**0.509**	**0.459**	**0.616**	0.643	**0.528**
	TS-RAG	0.319	0.263	0.310	0.298	0.892	**0.907**	0.478	0.441	0.583	**0.646**	0.514
	TSLM-ICL	0.331	**0.267**	**0.342**	0.283	0.884	0.899	0.462	0.438	0.586	0.633	0.512

Table 43: **Gemma-3-12B on TimeOmni.** Common subtask cells per block: k=1: 6, k=4: 6, k=8: 6.

	Method	Scenario (ID)	Causality (ID)	Decision (ID)	Scenario (OOD)	Causality (OOD)	Decision (OOD)	Average
	Zero-shot	0.510	0.329	0.160	0.518	0.333	0.179	0.338
$k=1$	Random	0.605	0.302	0.261	0.621	0.333	0.319	0.407
	Text	**0.630**	**0.343**	0.319	0.610	0.310	**0.421**	**0.439**
	TimeRAG	0.620	0.331	**0.340**	0.628	0.320	0.344	0.431
	TS-RAG	0.610	0.324	0.266	**0.631**	0.333	0.319	0.414
	TSLM-ICL	0.625	0.325	0.287	0.626	**0.341**	0.333	0.423
$k=4$	Random	0.615	0.268	0.340	0.602	0.297	0.421	0.424
	Text	0.650	0.265	0.340	0.598	0.292	0.513	0.443
	TimeRAG	0.600	0.274	0.394	0.641	**0.321**	0.505	0.456
	TS-RAG	0.645	0.270	**0.447**	0.638	0.292	0.513	0.468
	TSLM-ICL	**0.675**	**0.302**	0.383	**0.652**	0.306	**0.527**	**0.474**
$k=8$	Random	0.595	0.278	0.356	0.602	0.307	0.564	0.450
	Text	0.660	0.264	0.388	0.626	0.310	0.582	0.472
	TimeRAG	0.600	0.289	0.436	0.605	**0.319**	0.626	0.479
	TS-RAG	0.645	0.290	**0.457**	0.643	0.279	0.637	0.492
	TSLM-ICL	**0.685**	**0.296**	0.425	**0.647**	0.301	**0.692**	**0.508**