TIMEBRAID: UNIFYING TIME SERIES AND LANGUAGE FOR UNDERSTANDING AND FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present TimeBraid, a series of unified time-series and language models that align pretrained language models and pretrained time-series foundation models through interleaved global residual attention layers. Each model inherits knowledge, instruction following, and reasoning from one side and continuous-signal perception and zero-shot forecasting from the other, fusing both in a shared representation space for understanding and generation. We study the design choices enabling unified modeling: where to align the two representation spaces, how to ground language in temporal structure, how to balance understanding with generation, and how to stabilize joint optimization. The resulting recipe combines a unified prompting scheme for diverse time-series and text tasks, stabilized joint training, and supervision from 2.2M curated series–text pairs and 4.9M instruction-tuning samples. Across benchmarks for time-series perception, understanding, reasoning, and both context-aided and unimodal forecasting, TimeBraid remains competitive with far larger general-purpose models and task-specific counterparts.

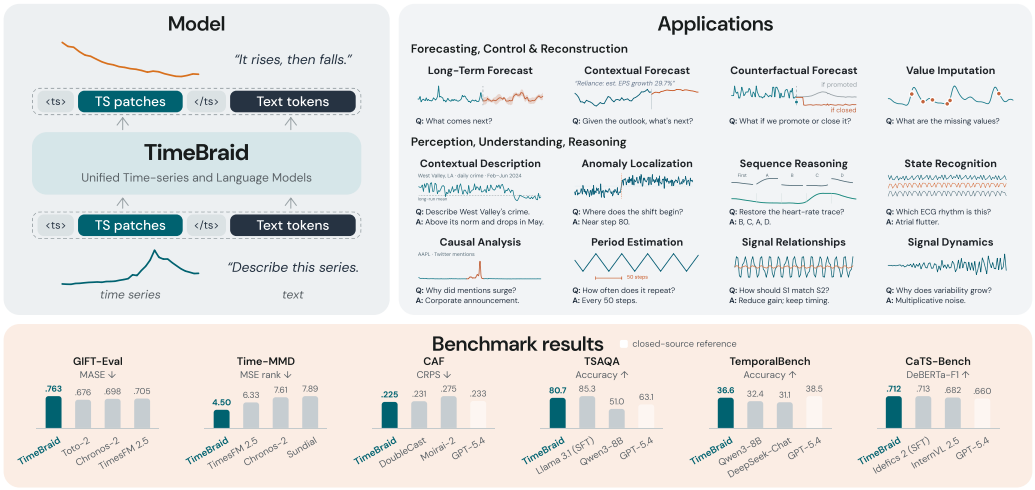


Figure 1: **Overview of TimeBraid.** One model reads and writes interleaved time-series patches and text tokens (left) across diverse tasks (right), compared with the strongest baselines on six benchmarks (bottom). SFT denotes benchmark-specific fine-tuning; closed-source models are shown as references.

1 INTRODUCTION

From early development, continuous interaction with a rich stream of heterogeneous signals shapes human intelligence. Vision, audition, language, and tactile sensation arrive as synchronized time series, allowing information from one modality to contextualize and complement information from others (Smith & Gasser, 2005; Smith et al., 2018; Orhan et al., 2020). Multimodal systems have

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.

largely relied on composed workflows or architectures merging separately developed, modality-specific modules. Unified multimodal models explore a complementary direction: learning shared representations for understanding and generating heterogeneous signals with one set of weights (Team, 2024; Zhou et al., 2025a; Xie et al., 2025; Wang et al., 2024; Chen et al., 2025; Deng et al., 2025; Xu et al., 2025a; An et al., 2026). Emerging evidence suggests that joint multimodal pretraining may foster capabilities difficult to acquire from any modality alone (Tong et al., 2026).

However, time-series data remains largely outside this unification. Numerical records of dynamical systems (e.g., physiology, power grids, markets, and climate) lack the semantic context for identification and interpretation (Goldberger et al., 2000; Hong & Fan, 2016; Tsay & Tsay, 2010; Lam et al., 2023; Shallue & Vanderburg, 2018). Language grounding, however, can supply this context and connect time series with broader world knowledge, motivating their inclusion as an essential modality in unified models.

Existing approaches typically connect time series and language through specialized components, from projection layers to signal tokenizers, added to pretrained language or time-series models (Xie et al., 2024; Guan et al., 2026a; Jin et al., 2024; Wang et al., 2025a). However, such components introduce representational gaps, causing semantic misalignment and degrading pretrained capabilities, and each system covers only part of the full bidirectional interface (Appendix H). Connecting a pretrained language model and a pretrained time-series foundation model in their native representation spaces, so that one model inherits the capabilities of both, remains open.

Such a connection requires large-scale paired pretraining, and the semantic gap between continuous signals and discrete language makes this alignment challenging. We present **TimeBraid**, a series of unified time-series and language models that bridges this gap. Much as the brain integrates information from different senses, TimeBraid is a mixture-of-Transformers (MoTs) that keeps separate pretrained experts for language reasoning, time-series perception, and forecasting, each running its native blocks, and braids them through *global residual attention*. At interleaved layers, text tokens and time-series patches are placed on one chronological axis and attend to each other in a shared space. To learn this bidirectional interface, we curate 2.2M time-series–language pairs for understanding and text-conditioned forecasting and train the model through alignment followed by supervised fine-tuning on 4.9M instruction samples.

Across twelve understanding and forecasting benchmarks (Jing et al., 2026; Cai et al., 2024; Weng et al., 2026; Zhou et al., 2026; Liu et al., 2024a; Wang et al., 2025a; Zheng et al., 2026; Williams et al., 2024; Aksu et al., 2024), TimeBraid (1.2B–6.7B parameters) serves both directions of the series–text interface with one set of weights (Figure 1). *It approaches benchmark specialists in understanding.* TimeBraid-6.7B reaches 80.65% on TSAQA, against 85.26% for LLaMA3.1-8B fine-tuned on TSAQA alone and 63.10% for GPT-5.4, which even TimeBraid-1.2B surpasses. Its CaTS-Bench caption alignment exceeds every frontier model. It leads all open models on TimeSeriesExam and, on out-of-domain TemporalBench, performs on par with GPT-5.4 (36.58% vs. 38.50%). *Text improves and steers its forecasts.* Its three sizes take the top three average MSE ranks across the nine Time-MMD domains, each outperforms every baseline on all three CGTSF datasets, and it attains the lowest CRPS on CAF. It is also the only non-LLM model whose Ctrl-F forecasts follow the stated condition. *It retains foundation-model forecasting.* From numerical histories alone, it stays close to state-of-the-art time-series foundation models on GIFT-Eval and takes the second-best average MSE rank on long-horizon ETT and Weather. Ablations of this new model class identify the shared attention space, stabilized joint optimization, and the alignment stage as its most important design choices. Together, these results establish TimeBraid as a series of powerful and versatile unified models of time and language.

2 PRELIMINARIES

We formulate time-series and language tasks in a unified format by interleaving segments of the two modalities. Formally, an example is a segment sequence $S = (s_1, \ldots, s_M)$ with an unconstrained number and order of segments. Each segment is either a *text segment* (a token sequence) or a *time-series segment* (a univariate value sequence $(x_1, \ldots, x_n) \in \mathbb{R}^n$, with a multivariate series entering as one segment per variable), and is designated as *context* or *target*. Context segments are fully observed. Text s_{text} carries instructions, questions, and background, and the time-series segment $s_{\text{ts},C}$ carries observed measurements. Target segments are model outputs. A text target is a

language response to be generated, and a time-series target $s_{\mathrm{ts},T} = (x_1, \ldots, x_h, x_{h+1}, \ldots, x_{h+f})$ repeats the observed history $(x_1, \ldots, x_h)$ of its context segment and extends it by f future values to be generated. Every task thereby becomes one conditional generation problem. The model generates each target segment conditioned on everything before it, so an example's likelihood factorizes as $\prod_{i \in \mathcal{I}_{\mathrm{tgt}}} p_\theta(s_i \mid s_{<i})$ over the target segments $\mathcal{I}_{\mathrm{tgt}}$, and within a time-series target only the f future values are generated and scored. Forecasting, contextual forecasting, captioning, question answering, and multivariate understanding are all instances of this template.

3 METHOD

TimeBraid unifies time series and language using the native representations of a pretrained language model and a pretrained time-series foundation model (Figure 2). It is designed to (i) serve as a general-purpose model for time-series and language understanding and generation, allowing flexible combinations of both modalities as inputs and outputs without a fixed task format; and (ii) reuse pretrained representations from both models rather than learn them from scratch.

3.1 UNIFIED TIME-SERIES AND LANGUAGE MODELING

Unified Time-series and Language Prompting. The prompting scheme defines the serialization $S \mapsto u_{1:T}$, mapping a segment sequence (Section 2) to T mixed text tokens and time-series patches in the chronological order of the original segments. Each example is rendered in the language model's native chat template, with every time-series segment wrapped in `<ts></ts>`. A context segment is preceded by a statistics block giving its length, mean, and standard deviation, while a target segment carries no statistics. Text is tokenized as in the language model, and each series is normalized and cut into patches of P values as in the time-series foundation model. Context segments are normalized by their own statistics. A target segment is normalized by the statistics of its observed history and then by rolling RevIN (Kim et al., 2021), so normalization never looks ahead and follows non-stationary futures. The transcripts below show a multivariate understanding and a univariate forecasting example.

Multivariate

User: Equity: `<stats>len=94, mean=103, std=4</stats><ts>` `</ts>`. Stress: `<stats>len=90, mean=19, std=7</stats><ts>` `</ts>`. How do they relate?

Assistant: Stress leads each equity drawdown.

Univariate

User: Here is the last 96 hours of server CPU load: `<stats>len=96, mean=42.1, std=8.3</stats><ts>` `</ts>`. A deployment just shipped, forecast the next 96 hours.

Assistant: `<ts>` `</ts>`

Tri-Expert Architecture. TimeBraid's core is a Mixture-of-Transformers (MoT) architecture unifying time-series perception, reasoning, and forecasting. The perception expert encodes context time-series segments; the reasoning expert handles text inputs and outputs, including instructions and language responses; the forecasting expert generates target time-series segments from time-series and text contexts. We initialize the experts with their corresponding pretrained models, e.g., large language models and time-series foundation models. Note that while the perception and forecasting experts process different inputs with different normalization schemes, they can share the same underlying model (dual-tower architecture). One may also consider using different backbones (tri-tower architecture) to further disentangle the representations used for understanding and forecasting, analogous to unified vision models that decouple understanding from generation representations (Chen et al., 2025). The dual- and tri-tower variants have similar language and time-series alignment-loss curves (Figure 3E), so we use the shared time-series backbone.

Global Residual Attention. Each time-series layer is paired with one language layer, and the pairs are evenly interleaved across depth. When the time-series tower is shallower, the remaining language layers process only their own tokens. The towers interact through global residual attention layers at the paired positions. At each paired layer, tower $r \in \{\mathrm{L}, \mathrm{T}\}$ first applies its native block to its own tokens, producing hidden states $H_r \in \mathbb{R}^{n_r \times d_r}$ with n_r tokens and native width d_r. The

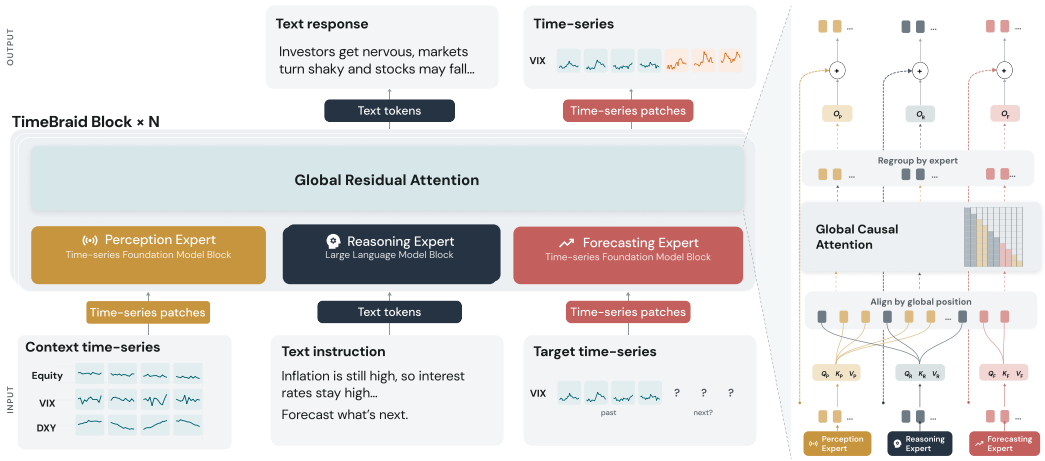


Figure 2: **TimeBraid architecture.** Each expert runs its native blocks on its own tokens. Global residual attention aligns all tokens by global position, attends causally across modalities, and regroups the outputs by expert as residual updates (right).

global residual attention layers project them into a shared attention space of width d for self-attention on the global mixed sequence,

$$Q_r = \psi_r^Q\big(\phi_r(H_r)\,W_r^Q\big), \qquad K_r = \psi_r^K\big(\phi_r(H_r)\,W_r^K\big), \qquad V_r = \phi_r(H_r)\,W_r^V, \tag{1}$$

where $W_r^Q, W_r^K, W_r^V \in \mathbb{R}^{d_r \times d}$ map each stream to the shared width, ϕ_r is a stream-specific RMSNorm on hidden states, and ψ_r^Q, ψ_r^K are per-head RMSNorms on queries and keys (not on values). The projected tokens of both streams are reordered into single global sequences $\tilde{Q}, \tilde{K}, \tilde{V}$ by their timeline positions τ and attended jointly under a causal mask,

$$O = \mathrm{Attention}\big(\mathrm{RoPE}_\tau(\tilde{Q}),\, \mathrm{RoPE}_\tau(\tilde{K}),\, \tilde{V}\big), \qquad H_r \leftarrow H_r + O_r\,W_r^O, \tag{2}$$

where O_r contains stream r's rows of O, routed back to their original positions, and the output projections $W_r^O \in \mathbb{R}^{d \times d_r}$ are zero-initialized so each global layer starts as an identity map. Text and patch tokens thus attend to all earlier tokens of both modalities, exchanging cross-modal information bidirectionally at the token level rather than through a pooled summary.

Disentangled Positional Embeddings. We use three positional embeddings: (1) the language tower applies rotary embeddings (RoPE) inside its native blocks, on the patch-expanded positions, (2) the time-series tower uses native positions locally, restarting in each segment, (3) the residual attention applies RoPE on the shared global ordered positions τ (Eq. 2), where both modalities are placed on one sequential axis. Algorithm 1 in Appendix C gives full forward-pass pseudocode.

3.2 Training and Inference

Joint Training Loss. Text targets use standard next-token cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$. Time-series targets use autoregressive Multi-Patch Prediction (MPP), predicting every target patch from its preceding context. The time-series loss combines point regression for the central trajectory with quantile regression for the predictive distribution and uncertainty (Appendix C.1). The total objective is $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \alpha\,\mathcal{L}_{\mathrm{TS}}$.

Stabilizing the Joint Optimization. Language and time-series losses differ in optimization behavior, making naive joint training ineffective. We observe that on strongly non-stationary segments, where the future departs from history statistics, regression loss $\mathcal{L}_{\mathrm{TS}}$ can spike by orders of magnitude, disrupting language-side learning through shared backbone updates. We use two techniques. First, *robust normalization*: during the causal RevIN normalization, each raw target value

x is transformed as $y = \operatorname{asinh}\big((x-\mu)/\sigma\big)$, where μ and σ are the rolling RevIN history statistics, producing the standardized targets in Eq. 4. The map is linear near zero and logarithmic in the tails, compressing extreme targets without clipping them. Second, *loss capping*: each response's time-series loss ℓ is rescaled by the detached factor $s = \min\big(1,\, c/\,\mathrm{stop_gradient}(\ell)\big)$, where c is a fixed cap threshold (Appendix C). This caps the value and gradient contribution of outlier responses and leaves in-range responses unchanged.

Text Strength Modulation at Inference Time. Text can be decisive (a verified maintenance schedule) or weak (financial-series commentary); some histories forecast themselves while others hinge on stated events. We therefore modulate text conditioning at inference time. Predicting $s_{\mathrm{ts},T}$ from $s_{\mathrm{ts},C}$ alone or jointly with s_{text} gives time-series-only $\hat{y}_{\mathrm{ts}}$ and joint $\hat{y}_{\mathrm{text+ts}}$, interpolated by $\lambda \in [0,1]$,

$$\hat{y}_\lambda = (1-\lambda)\,\hat{y}_{\mathrm{ts}} + \lambda\,\hat{y}_{\mathrm{text+ts}}. \tag{3}$$

Small λ favors the series' global temporal regularities; large λ keeps forecasts sensitive to the text within the horizon it describes. One trained model thus adapts across domains and inputs of varying certainty.

3.3 Data Recipe and Training Setup

Stage 1: Alignment. Stage 1 trains time-series understanding and text-conditioned forecasting on 2.23M curated pairs (Appendix A), split 3:7 between the two. For understanding, VLMs caption real series from plots, statistics, and domain context, while attribute-first synthesis (Xie et al., 2024) and structural causal models yield multivariate descriptions exact by construction. For forecasting, a general branch annotates realized futures of real series with domain semantics, and a control branch derives several futures per history with level shifts, ramps, pulses, and similar operators, each stated as a textual condition. LLMs rewrite some conditions into domain-grounded and genre-varied text.

Stage 2: Supervised Fine-Tuning. Stage 2 fine-tunes on about 4.9M instruction samples from public datasets. Forecasting sources such as CAF-7M, FinMultiTime (Xu et al., 2025b), Time-MMD, and CGTSF account for about 55% of the training mixture. QA, reasoning, and captioning draw on OpenTSLM (Langer et al., 2025), TSAQA, Time-MQA (Kong et al., 2025), and ChatTS. Unimodal (12%) and retained alignment data (8%) make up the rest (Appendix B).

Training Setup. We train three scales: TimeBraid-1.2B, TimeBraid-2.5B, and TimeBraid-6.7B pair Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B language towers (Yang et al., 2025) with TimesFM-2.5-derived time-series towers (Das et al., 2023). Both stages update all language, time-series, and global-attention parameters under the joint objective. We use AdamW with a constant learning rate of 2×10^{-5} after 500 warmup steps and a global batch size of 128. The time-series loss weight is $\alpha = 0.5$. Alignment uses a context length of 2,560 for one epoch, while supervised fine-tuning uses 4,096 for 30k steps. Appendix C gives the full training settings.

4 Experiments

We compare TimeBraid with general-purpose, specialized time-series, and unified models. Appendices B.3, E, and F document data overlap, protocols, and full results. Bold and underline mark the top two distinct values among comparable models shown, excluding closed-source references; ties share a rank. Dashes indicate unavailable results.

4.1 Time-Series Understanding

Perception, Question Answering, and Reasoning. TSAQA tests perception of a given series, from anomaly detection to temporal ordering. TimeSeriesExam and TemporalBench test time-series concepts and temporal reasoning. TimeBraid-6.7B reaches 80.65% on TSAQA, close to the 85.26% of LLaMA3.1-8B fine-tuned on TSAQA alone (Jing et al., 2026) (Table 1). GPT-5.4 reaches 63.10%, below even TimeBraid-1.2B. TimeBraid-6.7B nearly matches the fine-tuned model on data transformation, and most of the remaining gap comes from temporal ordering. It leads all open models on TimeSeriesExam and TemporalBench. Even on out-of-domain TemporalBench,

TimeBraid-6.7B comes within two points of GPT-5.4. Understanding also improves with the language tower. TimeBraid-1.2B and 2.5B share the same time-series tower and differ mainly in the language tower, and their TimeSeriesExam accuracy rises from 50.13% to 60.05%.

Table 1: CaTS-Bench human-rewritten caption scores ($[0, 1]$) and QA accuracy (%); higher is better. TSAQA selects AD-TF (anomaly detection), DT-MC (data transformation), TR-PZ (temporal ordering), and Overall; full subtests are in Appendix F.

Model	CaTS-Bench		TSAQA				TimeSeriesExam	TemporalBench
	DeBERTa F1	Numeric	AD-TF	DT-MC	TR-PZ	Overall	Acc. (%)	Acc. (%)
Closed-source Reference Models								
GPT-5.4	0.660	0.778	53.32	54.94	51.02	63.10	67.83	38.50
GPT-4.1	0.672	0.757	55.85	51.13	45.77	62.82	67.89	36.91
GPT-4o	0.685	0.739	54.32	53.24	45.61	60.73	55.96	32.30
Gemini-2.5-Flash	0.694	0.802	52.08	60.17	60.84	65.08	53.89	34.61
Open-source Large Language Models								
Qwen3-8B	0.670	0.678	50.60	34.46	21.93	51.04	47.45	32.43
LLaMA3.1-8B	0.667	0.697	54.92	36.56	6.80	44.93	35.52	25.88
Benchmark-Specific Fine-Tuned Models								
LLaMA3.1-8B	–	–	**91.02**	**86.62**	**67.68**	**85.26**	–	–
Ministral-8B	–	–	71.56	71.61	56.82	74.74	–	–
Gemma3-1B	–	–	83.10	45.23	42.92	69.70	–	–
Idefics 2	**0.713**	0.748	–	–	–	–	–	–
QwenVL	0.678	0.669	–	–	–	–	–	–
LLaVA v1.6	0.644	0.589	–	–	–	–	–	–
Time-Series Language Models								
ChatTS	0.614	0.648	49.42	30.29	33.59	49.83	50.72	24.90
TimeOmni-1-7B	0.634	**0.769**	50.98	28.39	30.93	44.40	35.25	25.67
Unified Models								
ChatTime-7B	0.371	0.077	50.20	25.15	27.87	38.38	41.94	28.45
TimeOmni-VL	0.598	0.370	50.82	23.55	15.52	39.27	49.06	32.32
TimeBraid-1.2B (Ours)	0.706	0.666	82.58	77.92	45.04	76.61	50.13	34.17
TimeBraid-2.5B (Ours)	0.708	0.670	83.23	82.92	48.36	78.31	60.05	35.74
TimeBraid-6.7B (Ours)	0.712	0.684	83.43	86.39	49.39	80.65	**63.14**	**36.58**

Contextual Description. CaTS-Bench asks a model to describe a series given its background. It scores agreement with human-rewritten captions and the accuracy of the numbers each caption cites. All three TimeBraid sizes agree with the human captions better than every closed-source model, and even TimeBraid-1.2B exceeds the best of them, Gemini-2.5-Flash (0.706 vs. 0.694 DeBERTa F1). Numeric fidelity trails the closed-source models, because each patch compresses 32 values into one embedding and does not fully preserve exact values (Appendix I). TimeBraid-6.7B reaches 0.712, on par with Idefics 2 fine-tuned on CaTS-Bench (0.713). Every TimeBraid size also leads the other time-series language and unified models on all caption-alignment metrics (Table 12).

4.2 Time-Series Forecasting

Contextual Forecasting. Time-MMD and CGTSF pair real-world series with text about their domain (Table 2). The three TimeBraid sizes take the top three average MSE ranks across the nine Time-MMD domains (4.11, 4.67, and 5.33, against 5.56 for TimesFM-2.5). TimeBraid has the lowest MSE in four domains and falls outside the top three only in Agriculture. On CGTSF, every TimeBraid size beats every baseline on all three datasets, and TimeBraid-6.7B lowers MSE by up to 19% against the best baseline (Table 16).

Controlled Forecasting. Our Ctrl-F benchmark tests whether text can steer a forecast (Table 3). Each held-out GIFT-Eval history (Aksu et al., 2024) comes with three different futures, and a text condition names the intended one. The history is the same for all three, so a model that ignores the text stays at the 33.33% chance level. Every time-series foundation model and multimodal forecaster scores exactly 33.33%. TimeBraid-6.7B reaches 43.33% and has the lowest Ctrl-F MSE among open models. CiK needs background knowledge beyond the series, and there frontier LLMs lead while TimeBraid matches the strongest time-series foundation model. CAF gives text that describes the coming scenario, and there TimeBraid-6.7B has the lowest CRPS of all models, below GPT-5.4 (0.225 vs. 0.233). Unlike understanding, controlled forecasting tracks the time-series tower.

Table 2: Real-world contextual forecasting. Time-MMD reports nine-domain MSE over four horizons; CGTSF reports training-standardized MSE over four history lengths. Text-conditioned models receive paired text; PatchTST* adds a text branch.

Model	Time-MMD (MSE↓)									CGTSF (MSE↓)		
	Agri.	Clim.	Econ.	Ener.	Env.	Health	Sec.	Soc.	Traf.	MSPG	LEU	PTF
Time-Series Foundation Models												
TimesFM-2.5	0.127	**0.859**	0.017	0.239	0.569	**0.665**	114.8	0.845	0.152	0.585	0.777	0.243
Chronos-2	**0.085**	0.988	**0.015**	0.262	0.566	1.217	110.3	1.110	0.201	0.649	0.715	0.230
Moirai	0.102	0.949	0.020	0.277	0.581	1.280	109.8	0.859	0.163	1.000	0.681	0.284
Sundial	0.102	0.865	0.022	0.266	0.561	1.240	111.7	0.856	0.160	0.763	0.683	0.242
TimeMoE	0.107	0.935	0.021	0.277	0.579	1.229	110.4	0.801	0.167	1.138	0.698	0.239
Multimodal Forecasting Models												
Aurora	0.109	1.428	0.040	0.367	0.597	2.000	120.7	1.155	0.322	0.789	0.757	0.336
MIGAS-1.5	0.086	0.983	**0.015**	0.242	0.671	1.302	109.9	1.110	0.200	0.757	0.887	0.386
Time-LLM	0.098	1.320	0.031	0.279	0.519	1.488	114.9	1.023	0.223	0.583	0.699	0.163
GPT4MTS	0.093	1.262	0.027	0.266	0.518	1.519	111.2	1.014	0.249	0.567	0.694	0.180
Time-VLM	0.093	1.282	0.020	0.253	0.494	1.482	113.0	1.068	0.229	0.818	0.725	0.175
PatchTST*	0.093	1.267	0.019	0.269	**0.492**	1.556	111.7	1.110	0.216	0.602	0.706	0.179
Unified Models												
ChatTime	0.130	2.279	0.071	0.365	1.099	2.516	144.3	1.227	0.505	2.727	1.015	0.359
TimeBraid-1.2B (Ours)	0.116	0.899	0.025	0.256	0.516	1.002	109.0	0.808	0.148	0.472	0.642	0.136
TimeBraid-2.5B (Ours)	0.137	0.871	0.025	0.235	0.507	0.956	108.9	0.839	**0.145**	0.467	0.644	**0.133**
TimeBraid-6.7B (Ours)	0.144	0.938	0.018	**0.231**	0.507	0.925	**108.6**	**0.799**	0.153	**0.459**	**0.641**	0.138

TimeBraid-1.2B and 2.5B share the same time-series tower and tie on CAF and Ctrl-F Top-1, although the 2.5B language tower is nearly three times larger. TimeBraid-6.7B has a larger time-series tower and improves on both.

Table 3: Controlled forecasting: Ctrl-F normalized MSE and correct-sibling Top-1; CAF pooled normalized CRPS; CiK official-weight RCRPS.

Model	Ctrl-F		CAF	CiK
	MSE↓	Top-1 (%)↑	CRPS↓	RCRPS↓
Closed-source Reference Models				
GPT-5.4	5.188	51.33	0.233	0.145
GPT-4o	5.447	60.00	0.234	0.257
Gemini-2.5-Flash	6.624	56.00	0.247	0.110
Open-source Large Language Models				
DeepSeek-Chat	6.405	**59.33**	0.267	**0.225**
Time-Series Foundation Models				
Moirai-2	8.425	33.33	0.275	0.288
Chronos-2	8.597	33.33	0.277	0.295
TimesFM-2.5	8.784	33.33	0.281	0.295
Sundial	8.829	33.33	0.331	0.314
Time-MoE-200M	9.077	33.33	0.414	0.327
Multimodal Forecasting Models				
Aurora	9.091	33.33	0.460	0.313
MIGAS-1.5	8.604	33.33	0.346	0.351
Unified Models				
ChatTime	9.165	33.33	0.345	0.344
TimeOmni-VL	10.387	32.00	0.489	0.655
TimeBraid-1.2B (Ours)	6.826	40.67	0.235	0.301
TimeBraid-2.5B (Ours)	6.972	40.67	0.235	0.289
TimeBraid-6.7B (Ours)	**6.232**	43.33	**0.225**	0.294

Table 4: GIFT-Eval (Aksu et al., 2024) normalized MASE and CRPS (lower is better).

Model	MASE↓	CRPS↓
Task-Specific (Supervised)		
DeepAR	1.343	0.853
TiDE	1.091	0.772
N-BEATS	0.938	0.816
PatchTST	0.849	0.587
iTransformer	0.893	0.620
Time-Series Foundation Models		
TimesFM-2.5	0.705	0.490
TabPFN-TS	0.771	0.544
Chronos-2	0.698	0.485
Moirai-2	0.728	0.516
Sundial Base	0.750	0.559
TiRex	0.716	0.488
Toto-2.0 FnF	**0.676**	**0.463**
Timer-S1	0.693	0.485
TTM-R3-PT	0.724	0.520
Kairos-50M	0.742	0.548
TiRex-2-Pretrained	0.678	0.467
Moirai-1.1-R-large	0.875	0.599
Unified Models		
Chronicle	1.053	0.754
TimeBraid-2.5B (Ours)	0.763	0.546

Unimodal Forecasting. Without text, TimeBraid keeps the forecasting ability of its TimesFM-2.5 backbone (Tables 4 and 20). GIFT-Eval covers many domains and forecast lengths. There TimeBraid-2.5B stays within 0.06 MASE of TimesFM-2.5 and ahead of every supervised baseline. On long-horizon ETT and Weather, it takes the second-best average MSE rank among ten models. Chronicle learns a joint series–language space from scratch. TimeBraid-2.5B, built on pretrained towers, outperforms it by a wide margin (0.763 vs. 1.053 MASE).

4.3 Ablation Studies

We examine which architectural and optimization choices support cross-modal alignment (Figure 3) and whether alignment improves subsequent supervised fine-tuning (Figure 4). Appendix G adds language retention and inference-time text guidance.

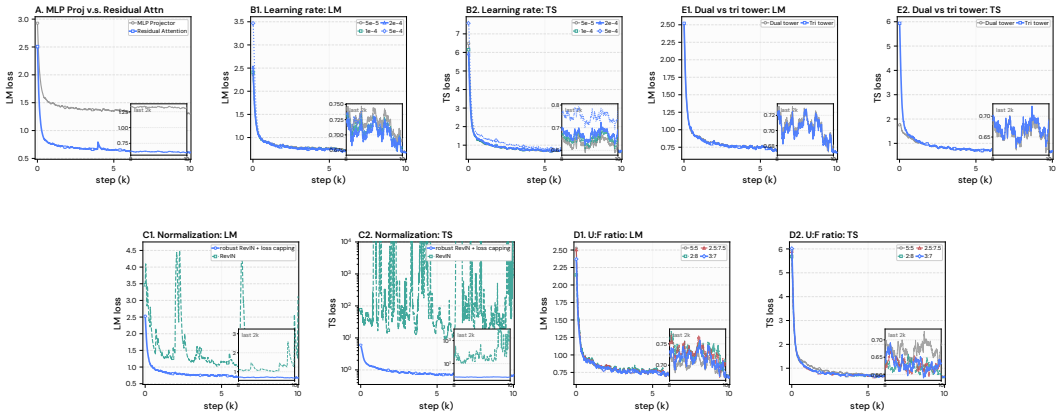


Figure 3: Design ablations: training-loss trajectories from 10k-step runs differing only in the ablated factor: **(A)** cross-modal interface, **(B)** shared learning rate, **(C)** plain vs. robust RevIN, **(D)** understanding-to-forecasting data ratio, **(E)** dual- vs. tri-tower. Insets zoom into the last 2k steps; lower is better.

RQ1: Interaction Space. Panel A tests whether the language representation is a suitable interaction space for temporal patches. We compare the MLP connector used to project another modality into the language space (Liu et al., 2023; Xie et al., 2024) with residual attention in a shared space (Liang et al., 2024). With the MLP connector, language loss plateaus early at roughly twice the level reached by residual attention, and the gap persists throughout training. Projected patches often retain only global statistics such as mean and standard deviation. The lower loss with residual attention supports aligning the pretrained towers through a shared interaction space, consistent with prior findings on cross-modal alignment limits (Yashwante & Yu, 2026).

RQ2: Expert Separation. Panel E compares the dual- and tri-tower variants of Section 3.1. Their language and time-series alignment-loss trajectories are nearly identical throughout the 10k-step comparison, so we retain the shared time-series tower. Appendix G.1 gives the full comparison.

RQ3: Optimization Stability. Time-series regression is sensitive to scale: when a non-stationary future departs sharply from its history statistics, history-based normalization can produce extreme targets. Panel C shows the resulting instability under plain causal RevIN (Kim et al., 2021). Time-series loss repeatedly spikes by orders of magnitude (C2), and the shared updates also disrupt language-side optimization (C1). Robust normalization and loss capping remove these recurrent spikes and stabilize both objectives. Controlling extreme regression updates is therefore important for joint training, even when the language objective itself is well behaved.

RQ4: Balancing the Two Modalities. Panels B and D show that the two objectives respond differently to the learning rate and data mixture. Across the shared learning-rate sweep, language loss remains stable (B1), whereas time-series loss degrades at the largest rate (B2). Changing the understanding-to-forecasting data ratio shows the same asymmetry: the forecasting-heavy mixtures

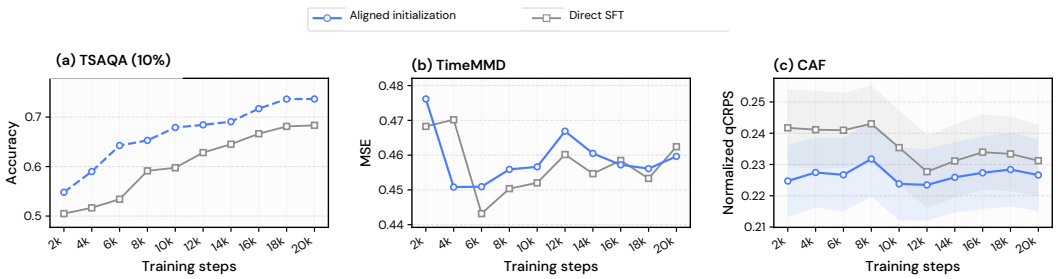


Figure 4: SFT trajectories from aligned vs. pretrained towers: **(a)** TSAQA accuracy (10% test subset, higher is better), **(b)** Time-MMD MSE and **(c)** CAF normalized qCRPS (lower is better for both).

from 3:7 to 2:8 reach similar time-series losses, below the balanced 5:5 mixture (D2), while language loss remains largely unchanged (D1). We therefore choose 3:7, which retains the largest understanding share among the tested mixtures with the lower forecasting loss. Together with Panel C, these results motivate robust normalization and loss capping, a moderate learning rate, and a forecasting-heavy mixture to support the more sensitive time-series objective.

RQ5: Necessity of the Alignment Stage. Paired-data alignment before instruction tuning is conventional but costly (Liu et al., 2023; Deng et al., 2025). Under the same SFT recipe (Figure 4), aligned initialization leads direct pretrained-tower initialization on TSAQA at every checkpoint, especially early. Time-MMD curves cross and end at nearly identical MSE: its text weakly constrains the future (RQ7 in Appendix G.3), so SFT's numeric-history skill largely covers it. On CAF, where text specifies the future, alignment maintains lower qCRPS at every checkpoint and direct SFT never closes the gap. We retain alignment for its lasting understanding head start and its value when forecasts must draw on language.

5 Related Work

Unified multimodal models understand and generate several modalities with one set of weights (Team, 2024; Zhou et al., 2025a; Deng et al., 2025), and mixture-of-Transformers designs give each modality its own weights under shared attention (Liang et al., 2024). TimeBraid brings this program to time series, braiding two *independently pretrained* models of different width and depth through a separate shared attention space. Time-series foundation models forecast zero-shot across domains from numerical histories (Das et al., 2023; Ansari et al., 2024; Woo et al., 2024; Liu et al., 2025b). TimeBraid inherits this competence and opens it to language. Series–language models typically move one modality into the other's host, projecting series into an LLM (Xie et al., 2024; Guan et al., 2026a) or adapting a language or vision model for forecasting (Jin et al., 2024; Zhong et al., 2025). Bidirectional models represent values as LLM tokens (Wang et al., 2025a) or plots (Guan et al., 2026b), or learn a joint space from scratch (Quinlan et al., 2026). TimeBraid keeps both pretrained hosts native within one model and scales their joint training to large-scale time-series–language data. Appendix H gives the full discussion.

6 Conclusion

We presented TimeBraid, a unified time-series and language model that aligns pretrained language and time-series foundation models in their native representations through tri-expert global residual attention. With dual-stage training on curated alignment pairs and a broad instruction mixture, one set of weights understands and generates both modalities, forecasting at the level of dedicated time-series foundation models while remaining competitive with far larger general-purpose models on understanding and context-aided forecasting. Our ablations distill this model class's key design choices: where cross-modal interaction should occur, whether perception and forecasting need separate experts, and how strongly forecasts should draw on text at inference. We hope TimeBraid advances unified models that treat time-series as an essential modality, understood and generated in its native representation.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic training data. Vision–language models annotate time-series morphology and context, and language models verbalize structured evidence for multivariate examples and annotate or rewrite forecasting conditions (Appendix A). We also used generative AI tools to implement parts of the training and evaluation code. We have not used generative AI tools to design the model or the experiments or to interpret results, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to assist in drafting and polishing the paper and to check formatting. We have reviewed all AI-assisted work. All model-generated annotations pass factual-consistency, style, and format filters, AI-assisted code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available time-series datasets and benchmarks under their respective licenses and involves no human subjects or personally identifiable information. Training annotations are generated by language and vision–language models from these public series and filtered for factual consistency (Appendix A). We will release models and data subject to the licenses and distribution terms of their sources.

REPRODUCIBILITY STATEMENT

Appendices A and B describe data construction and the training mixture. Appendix C specifies model configurations, training hyperparameters, and the forward-pass algorithm. Appendices E and F provide benchmark protocols and detailed results, and Appendix G reports additional ablations. Section 4.3 records the settings for the design comparisons. Our code, model checkpoints, and datasets will be made publicly available upon acceptance, subject to the licenses and distribution terms of their sources.

REFERENCES

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

Siyu An, Junru Lu, Junnan Dong, Qiufeng Wang, Yinghui Li, Weizhi Fei, Zichao Yu, Zheng Yuan, Biao Liu, Haopeng Wang, et al. Toward native multimodal modeling: A roadmap. *arXiv preprint arXiv:2605.25343*, 2026.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. Timeseriesexam: A time series understanding exam. *arXiv preprint arXiv:2410.14752*, 2024.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–20, 2025.

Jialin Chen, Ziyu Zhao, Gaukhar Nurbek, Aosong Feng, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. Trace: Grounding time series in context for multimodal embedding and retrieval. *Advances in Neural Information Processing Systems*, 38:2440–2472, 2026.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, et al. This time is different: An observability perspective on time series foundation models. *Advances in neural information processing systems*, 38:50907–50951, 2026.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Yueyang Ding, Haopeng Zhang, Rui Dai, Yi Wang, Tianyu Zong, Kaikui Liu, and Xiangxiang Chu. Llatisa: Towards difficulty-stratified time series reasoning from visual perception to semantics. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32677–32717, 2026.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181, 2024.

Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *International Conference on Learning Representations*, volume 2025, pp. 100218–100231, 2025.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

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

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

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 *International Conference on Learning Representations*, volume 2026, pp. 152139–152170, 2026a.

Tong Guan, Sheng Pan, Johan Barthelemy, Zhao Li, Yujun Cai, Cesare Alippi, Ming Jin, and Shirui Pan. Timeomni-vl: Unified models for time series understanding and generation. *arXiv preprint arXiv:2602.17149*, 2026b.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

Sheikh Asif Imran, Mohammad Nur Hossain Khan, Subrata Biswas, and Bashima Islam. Llasa: A multimodal llm for human activity analysis through wearable and smartphone sensors. *arXiv e-prints*, pp. arXiv–2406, 2024.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

Baoyu Jing, Sanhorn Chen, Lecheng Zheng, Boyu Liu, Zihao Li, Jiaru Zou, Tianxin Wei, Zhining Liu, Zhichen Zeng, Ruizhong Qiu, et al. Tsaqa: Time series analysis question and answering benchmark. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*, pp. 944–979, 2026.

Jaeik Kim, Woojin Kim, Jihwan Hong, Yejoon Lee, Sieun Hyeon, Mintaek Lim, Yunseok Han, Dogeun Kim, Hoeun Lee, Hyunggeun Kim, et al. Dynin-omni: Omnimodal unified large diffusion language model. *arXiv preprint arXiv:2604.00007*, 2026.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A Xu, Winnie Chow, Martin Maritsch, Robert Jakob, Ning Wang, Juncheng Liu, Aradhana Verma, et al. Opentslm: Time-series language models for reasoning over multivariate medical text-and time-series data. *arXiv preprint arXiv:2510.02410*, 2025.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933, 2024a.

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*, pp. 4095–4106, 2024b.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 83982–84006, 2025a.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025b.

Yong Liu, Xingjian Su, Shiyu Wang, Haoran Zhang, Haixuan Liu, Yuxuan Wang, Zhou Ye, Yang Xiang, Jianmin Wang, and Mingsheng Long. Timer-s1: A billion-scale time series foundation model with serial scaling. *arXiv preprint arXiv:2603.04791*, 2026.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Emin Orhan, Vaibhav Gupta, and Brenden M Lake. Self-supervised learning through the eyes of a child. *Advances in neural information processing systems*, 33:9960–9971, 2020.

Paul Quinlan, Jeremy Levasseur, Qingguo Li, and Xiaodan Zhu. Chronicle: A multimodal foundation model for joint language and time series understanding. *arXiv preprint arXiv:2605.20268*, 2026.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Christopher J Shallue and Andrew Vanderburg. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155(2):94, 2018.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *International conference on learning representations*, volume 2025, pp. 34635–34667, 2025.

Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005.

Linda B Smith, Swapnaa Jayaraman, Elizabeth Clerkin, and Chen Yu. The developing infant creates a curriculum for statistical learning. *Trends in cognitive sciences*, 22(4):325–336, 2018.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37:60162–60191, 2024.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026.

Ruey S Tsay and Ruey S Tsay. *Analysis of financial time series*, volume 3. wiley New York, 2010.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025a.

Siyuan Wang, Peng Chen, Yihang Wang, Wanghui Qiu, Guo Guo, Bin Yang, and Yang Shu. Unlocking the value of text: Event-driven reasoning and multi-level alignment for time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 31182–31210, 2026.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang Li, and Zhongyu Wei. Itformer: Bridging time series and natural language for multi-modal qa with large-scale multitask dataset. *arXiv preprint arXiv:2506.20093*, 2025b.

Muyan Weng, Defu Cao, Wei Yang, Yashaswi Sharma, and Yan Liu. Temporalbench: A benchmark for evaluating llm-based agents on contextual and event-informed time series tasks. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 9997–10008, 2026.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, et al. Context is key: A benchmark for forecasting with essential textual information. *arXiv preprint arXiv:2410.18959*, 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Congxi Xiao, Jingbo Zhou, Yixiong Xiao, Xinjiang Lu, Le Zhang, and Hui Xiong. Timefound: a foundation model for time series forecasting. *arXiv preprint arXiv:2503.04118*, 2025.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *Advances in Neural Information Processing Systems*, 38:47490–47518, 2026.

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. *arXiv preprint arXiv:2412.03104*, 2024.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025a.

Wenyan Xu, Dawei Xiang, Yue Liu, Xiyu Wang, Yanxiang Ma, Liang Zhang, Shu Hu, Chang Xu, and Jiaheng Zhang. Finmultitime: A four-modal bilingual dataset for financial time-series analysis. *arXiv preprint arXiv:2506.05019*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yiyuan Yang, Zichuan Liu, Lei Song, Kai Ying, Stephen Wang, Joshua Thomas Bamford, Svitlana Vyetrenko, Jiang Bian, and Qingsong Wen. Time-ra: Towards time series reasoning for anomaly diagnosis with llm feedback. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11591–11616, 2026.

Pratham Yashwante and Rose Yu. Time series, vision, and language: Exploring the limits of alignment in contrastive representation spaces. *arXiv preprint arXiv:2602.19367*, 2026.

Fangxu Yu, Hongyu Zhao, and Tianyi Zhou. Ts-reasoner: Aligning time series foundation models with llm reasoning. *arXiv preprint arXiv:2510.03519*, 2025.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9637–9662, 2024.

Vincent Zhihao Zheng, Étienne Marcotte, Arjun Ashok, Andrew Robert Williams, Lijun Sun, Alexandre Drouin, and Valentina Zantedeschi. Overcoming the modality gap in context-aided forecasting. *arXiv preprint arXiv:2603.12451*, 2026.

Siru Zhong, Yuxuan Liang, Weilin Ruan, Ming Jin, Huan Li, and Qingsong Wen. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. In *AI for Time Series*, pp. 173–203. CRC Press, 2025.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations*, volume 2025, pp. 6446–6469, 2025a.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Luca Zhou, Pratham Yashwante, Marshall Fisher, Alessio Sampieri, Zihao Zhou, Fabio Galasso, and Rose Yu. Cats-bench: Can language models describe time series? In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34479–34519, 2026.

Xin Zhou, Weiqing Wang, Francisco J Baldán, Wray Buntine, and Christoph Bergmeir. Motime: A dataset suite for multimodal time series forecasting. *arXiv preprint arXiv:2505.15072*, 2025b.

APPENDIX CONTENTS

A Data Curation Pipeline 17

A.1 Univariate Understanding 17

A.2 Multivariate Understanding 17

A.3 Univariate Forecasting 18

B Training Data 18

B.1 Alignment Corpus Composition 18

B.2 Supervised Fine-Tuning Sources 19

B.3 Overlap with Evaluation Benchmarks 20

C Implementation Details 21

C.1 Loss Definition 21

C.2 Model Configuration 21

C.3 Training Hyperparameters 22

D Prompt Template Examples 23

E Evaluation Protocols 25

F Detailed Experimental Results 28

F.1 Time-Series Understanding 28

F.2 Time-Series Forecasting 31

G Additional Ablations 37

G.1 RQ2 (Extended): Expert Separation 37

G.2 RQ6: Language Ability Retention 37

G.3 RQ7: Text-Guidance Sensitivity 38

H Related Work 39

I Limitations 40

J Success and failure cases 41

A Data Curation Pipeline

Figure 5 presents our data curation pipelines for univariate understanding, multivariate understanding, and univariate forecasting. Each pipeline pairs time series with language through a capability-specific annotation process, followed by shared filtering for factual consistency, style, and format.

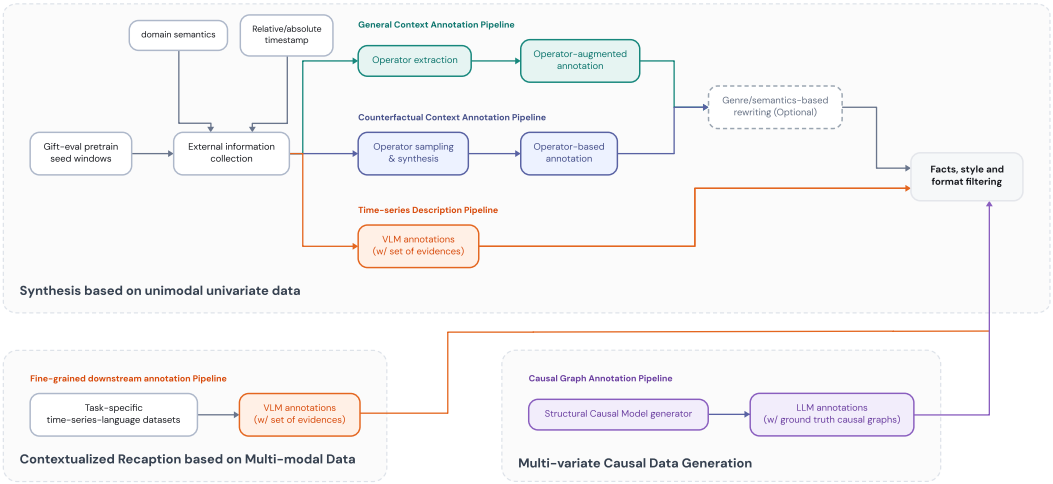


Figure 5: Alignment data curation for univariate understanding, multivariate understanding, and univariate forecasting.

A.1 Univariate Understanding

Univariate understanding data teaches the model to recognize temporal patterns and express them in language. We use vision-language models as the primary annotators because their exposure to charts and visual data equips them to perceive global morphology, such as trends, seasonality, and regime changes, together with local events such as peaks, troughs, and anomalies. Their annotations distill this temporal knowledge into textual supervision. For real series, each evidence pack combines the observed trace with deterministic statistics, extrema, trend segments, change points, periodic patterns, and temporal landmarks; the context-rich route additionally supplies domain semantics and calendar information. Separately, we adopt the attribute-first synthesis scheme of ChatTS (Xie et al., 2024), in which sampled temporal attributes determine both the generated series and its programmable description. We also rewrite procedurally generated exam prompts on pattern recognition, noise, similarity, anomalies, and Granger-causal structure into concise descriptive-answer supervision. All outputs pass factual, stylistic, and formatting filters.

A.2 Multivariate Understanding

Real multivariate time series rarely carry detailed, verifiable annotations of interactions across variables. We retain the cleaned multivariate descriptions from ChatTS (Xie et al., 2024) for broad descriptive coverage and construct graph-grounded relational supervision with temporal structural equation models. The source generator samples sparse or random lagged graphs over 3 to 50 observed variables with maximum lags from 2 to 50 steps, then simulates their multivariate trajectories. Each evidence pack records the graph-authorized variable roles, lag offsets, intervention metadata when present, temporal anchors, and canonical evidence for direct parent-to-target relations, short directed chains, localized shock follow-ons, historical reference windows from the same SCM, and local parent/child motifs. A text-only LLM converts this evidence into natural descriptions under automatic grounding checks. The active mixture contains 100,000 SCM descriptions, including 74,210 positive relation examples and 25,790 comparison controls whose channels are not directly adjacent to the target.

A.3 Univariate Forecasting

Paired data that connects an observed history, a textual condition, and its corresponding future is scarce in natural corpora. We construct this supervision through the general-context and counterfactual-context pipelines in Figure 5. The general branch annotates the realized suffix of a real window with its domain semantics and temporal information. The control branch produces three sibling continuations from one history: a baseline and two controlled variants, each composing up to three transformations drawn from level shifts, ramps, slope changes, pulses, seasonal-amplitude changes, saturations, shock decays, and analog replays. It renders the applied transformations as natural-language conditions, so the shared history can lead to distinct text-specified futures. The model input contains the observed history and the natural-language continuation condition, while time-series loss applies only to the held-out suffix. Optional genre and semantic rewrites diversify the conditions while preserving their connection to the target future. Both pipelines apply the shared factual, stylistic, and formatting filters.

B Training Data

Stage 1 contains 2,229,500 alignment examples, grouped into 12 families along the three curation routes of Appendix A. Stage 2 uses a final inventory of 4,881,583 samples across five composition categories, with alignment retention restricted to its recipe-defined quota.

B.1 Alignment Corpus Composition

Table 5 lists the 12 alignment families. The understanding and forecasting halves are fixed at 3:7 by construction (668,850 and 1,560,650 rows). The understanding data covers both univariate and multivariate cases: morphology captions describe the basic structures of a single series (trends, seasonality, regimes, salient features, inter-series relationships); context-rich captions describe real series together with their real-world context; programmable captions in the style of ChatTS (Xie et al., 2024) are generated attribute-first, so each description matches its series precisely by construction; multivariate examples are generated from structural causal models with known interaction structure; and templated programmable tasks (Cai et al., 2024) cover the same content from diverse task perspectives.

The forecasting data focuses on controllability and is built with future-aware annotation: annotations are produced with access to the future, while future numerical values are never included in the model input. The forecasting control set synthesizes several distinct futures from one shared history using programmable operators and captions the condition that leads to each future, so the model learns to follow stated conditions when forecasting. The general forecasting set annotates realized futures of real series, aligning context-conditioned forecasts with natural real-world evolution. We rewrite some forecasting examples in different tones and genres to make their presentation more diverse and realistic. Each forecasting branch appears in three variants: a base variant with horizon at most 512, a rewrite of the same rows into domain-grounded real-world context with horizon at most 32, and a partial genre rewrite under the same horizon limit. Control rows are organized into three-sibling groups that share one history: one neutral continuation and two operator-conditioned futures. Appendix A details the construction of each family.

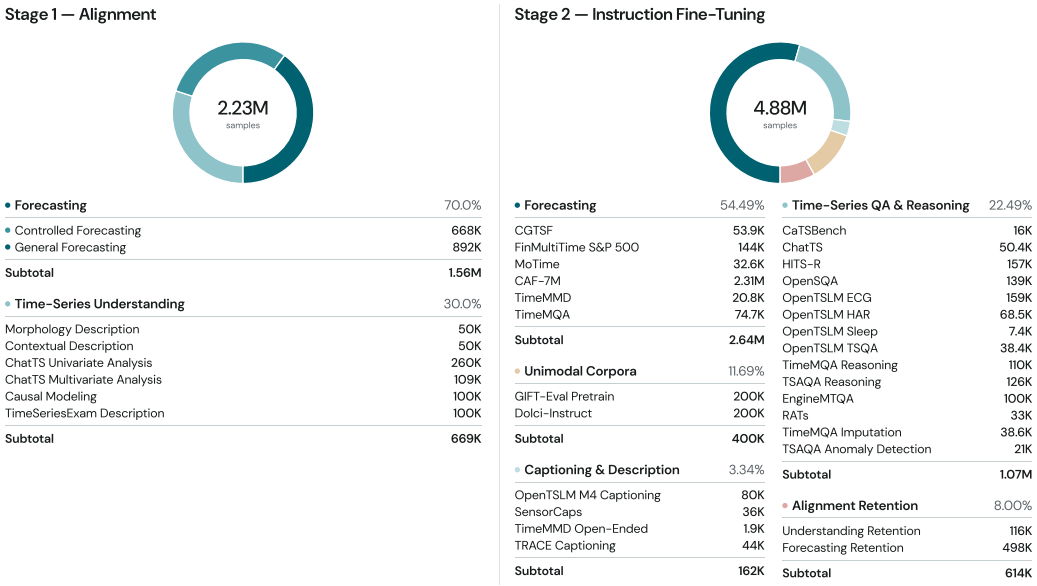


Figure 6: Two-stage training recipe. Stage 2 counts show the final sample inventory, while percentages show training-time sampling shares.

Table 5: Alignment corpus: 12 families, 2,229,500 examples. H is the forecast horizon. Rewrite variants reuse the base rows and change only the user-visible text.

Family	Curation route	n	Share
Understanding 668,850 rows, 30.0%			
Morphology captions	morphology-grounded VLM annotation	50,000	2.2%
Context-rich captions	context-grounded VLM annotation	50,000	2.2%
ChatTS (Xie et al., 2024) univariate	attribute-programmed synthesis	260,000	11.7%
ChatTS (Xie et al., 2024) multivariate	attribute-programmed synthesis	108,850	4.9%
Multivariate SCM	SCM-grounded relation synthesis	100,000	4.5%
TimeSeriesExam descriptions	procedural task synthesis	100,000	4.5%
Forecasting, control branch 668,291 rows, 30.0%			
Control, $H \leq 512$	operator-controlled future synthesis	338,953	15.2%
Control, $H \leq 32$	domain-grounded rewriting	164,918	7.4%
Control, $H \leq 32$	genre-conditioned rewriting	164,420	7.4%
Forecasting, general branch 892,359 rows, 40.0%			
General, $H \leq 512$	realized-future annotation	528,507	23.7%
General, $H \leq 32$	domain-grounded rewriting	255,291	11.5%
General, $H \leq 32$	genre-conditioned rewriting	108,561	4.9%

B.2 Supervised Fine-Tuning Sources

The second stage uses a heterogeneous mixture spanning time-series question answering and reasoning, captioning and description, contextual and unimodal long-horizon forecasting, and domain-specific applications in energy, climate, healthcare, finance, sensing, and industrial monitoring. Table 6 reports the final sample inventory and training-time sampling shares, including the understanding and forecasting quotas retained from alignment.

Table 6: Final Stage-2 sample inventory and training-time sampling shares. References identify the source datasets; counts refer to our selected and reformatted examples.

Source	Final samples	Training Share
Forecasting 2,640,566 samples, 54.49%		
CGTSF (Wang et al., 2025a)	53,926	2.04%
FinMultiTime S&P 500 (Xu et al., 2025b)	144,387	5.46%
MoTime (News/Wiki) (Zhou et al., 2025b)	32,589	1.23%
CAF-7M (Zheng et al., 2026)	2,314,216	42.15%
Time-MMD (Liu et al., 2024a)	20,768	0.79%
Time-MQA Forecasting (Kong et al., 2025)	74,680	2.82%
Time-Series QA & Reasoning 1,065,021 samples, 22.49%		
CaTS-Bench (Zhou et al., 2026)	15,995	0.33%
ChatTS (Xie et al., 2024)	50,377	1.05%
HiTSR (Ding et al., 2026)	157,147	3.24%
OpenSQA (Imran et al., 2024)	138,754	2.72%
OpenTSLM ECG (Langer et al., 2025)	159,313	3.28%
OpenTSLM HAR (Langer et al., 2025)	68,542	1.41%
OpenTSLM Sleep (Langer et al., 2025)	7,434	0.15%
OpenTSLM TSQA (Langer et al., 2025)	38,400	0.79%
Time-MQA Reasoning (Kong et al., 2025)	110,212	2.27%
TSAQA Reasoning (Jing et al., 2026)	125,957	2.60%
EngineMT-QA (Wang et al., 2025b)	100,296	2.07%
RATs40K (Yang et al., 2026)	32,994	0.68%
Time-MQA Imputation (Kong et al., 2025)	38,607	1.46%
TSAQA Anomaly Detection (Jing et al., 2026)	20,993	0.43%
Captioning & Description 161,883 samples, 3.34%		
OpenTSLM M4 Captioning (Langer et al., 2025)	80,000	1.65%
SensorCaps (Imran et al., 2024)	35,960	0.74%
Time-MMD Open-Ended (Liu et al., 2024a)	1,925	0.04%
TRACE Captioning (Chen et al., 2026)	43,998	0.91%
Unimodal Corpora 400,000 samples, 11.69%		
GIFT-Eval Pretrain (Aksu et al., 2024)	200,000	7.56%
Dolci-Instruct (No-Tools) (Olmo et al., 2025)	200,000	4.12%
Alignment Retention 614,113 samples, 8.00%		
Understanding Retention	116,294	1.51%
Forecasting Retention	497,819	6.48%
Final inventory	**4,881,583**	**100.00%**

B.3 Overlap with Evaluation Benchmarks

To ensure a valid evaluation, we strictly enforce no overlap with the training mixture at the split level, keeping evaluation examples held out even when related training data originate from the same benchmark family or underlying data-generating process. Table 7 summarizes this relationship for each benchmark.

Table 7: Training status of the evaluation suites. Status is determined by split membership in the mixture.

Benchmark	Status
Understanding	
TemporalBench	Out-of-domain.
TimeSeriesExam	Independently generated and rewritten samples from the same data-generating process are included in the alignment mixture.
TSAQA	Training split is included in the SFT mixture.
CaTS-Bench	Training split is included in the SFT mixture.
Forecasting	
ETT and Weather	Out-of-domain.
CiK	Out-of-domain.
Time-MMD	Training split is included in the SFT mixture.
CGTSF	Training split is included in the SFT mixture.
CAF	Part of the training split is included in the SFT mixture.
Ctrl-F	Independently generated samples from the same data-generating process are included in the alignment mixture.
GIFT-Eval	Part of the training split is used as seed data for data generation.

C Implementation Details

C.1 Loss Definition

Time-series and language modeling are jointly optimized with separate losses. Text targets use the standard next-token cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$. Time-series targets are supervised autoregressively via Multi-Patch Prediction (MPP): every patch of a target segment is predicted from its preceding context, and each predicted patch incurs a point term and a quantile term,

$$\mathcal{L}_{\mathrm{TS}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left[\left\| \hat{y}_t - y_t \right\|_2^2 + \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \max\big(q\,(y_t - \hat{y}_t^{(q)}),\ (q-1)\,(y_t - \hat{y}_t^{(q)}) \big) \right], \tag{4}$$

where $\mathcal{T}$ indexes the target patches of the example, y_t is the t-th standardized target patch, $\hat{y}_t$ its point prediction, and $\hat{y}_t^{(q)}$ its prediction at quantile $q \in \mathcal{Q} = \{0.1, 0.2, \ldots, 0.9\}$; the second term is the pinball loss. We use the point regression to describe the central trajectory and the quantile term to capture the predictive distribution and uncertainty. The total objective is $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \alpha\, \mathcal{L}_{\mathrm{TS}}$.

C.2 Model Configuration

Table 8 gives the parameter allocation across Qwen3 language towers (Yang et al., 2025), TimesFM-derived time-series towers (Das et al., 2023), and fusion stacks. Variant names round the total parameter count, including frozen weights and excluding buffers; shared weights are counted once.

Table 8: Parameter inventory across TimeBraid scales, in billions of parameters.

Variant	Language tower	Fusion	Time-series tower	Total
TimeBraid-1.2B	0.596	0.378	0.231	1.205
TimeBraid-2.5B	1.721	0.545	0.231	2.497
TimeBraid-6.7B	4.022	2.265	0.389	6.676

Geometry and layer pairing TimeBraid-1.2B and TimeBraid-2.5B interleave 20 time-series and fusion blocks across 28 language layers, while TimeBraid-6.7B pairs 36 time-series and fusion blocks with its 36 language layers, with its pretrained time-series tower expanded to 36 blocks by block duplication. Time-series segments use patches of $P = 32$ values, and each paired layer exchanges information through global residual attention.

```
# H_l: (n_l, d_l) language hidden states; H_t: (n_t, d_t) time-series
    hidden states
# tau_l, tau_t: timeline positions of each stream (global chronological
    order)
# pair_of[i]: time-series layer paired with language layer i, or None;
#   smaller variants have unpaired language layers, while 6.7B pairs all
    36 layers
# Wq_r, Wk_r, Wv_r: (d_r, d) projections to the shared width; Wo_r: (d,
    d_r), zero-init
# phi_r: hidden-state RMSNorm; psi_r: per-head QK RMSNorm (r in {l, t})

def forward(H_l, H_t, tau_l, tau_t):
    for i in range(num_lm_layers):
        H_l = lm_block[i](H_l)          # every language layer runs its
    native block (RoPE inside)
        if pair_of[i] is None:
            continue                    # unpaired layer: the language
    stream continues alone
        j = pair_of[i]
        H_t = ts_block[j](H_t)          # paired time-series layer (
    segment-local positions)
        H_l, H_t = fuse[j](H_l, H_t, tau_l, tau_t)   # global residual
    attention
    return H_l, H_t

def fuse(H_l, H_t, tau_l, tau_t):
    # project both streams into the shared attention width d (values: no
    head norm)
    q_l, k_l, v_l = psi_l(phi_l(H_l) @ Wq_l), psi_l(phi_l(H_l) @ Wk_l),
    phi_l(H_l) @ Wv_l
    q_t, k_t, v_t = psi_t(phi_t(H_t) @ Wq_t), psi_t(phi_t(H_t) @ Wk_t),
    phi_t(H_t) @ Wv_t

    # reorder into one global sequence by timeline position tau
    q, k, v, tau = merge_by_position([q_l, q_t], [k_l, k_t], [v_l, v_t],
    [tau_l, tau_t])

    # joint causal attention with RoPE on the shared global axis
    o = attention(rope(q, tau), rope(k, tau), v, mask="causal")

    # route rows back to their streams; zero-init Wo_r makes fusion start
     as identity
    o_l, o_t = split_by_stream(o)
    return H_l + o_l @ Wo_l, H_t + o_t @ Wo_t
```

Algorithm 1: Pseudocode of TimeBraid's forward pass. Every language layer runs its native block. At paired positions, the corresponding time-series block runs and both streams exchange information through global residual attention (Eqs. 1 and 2); at any unpaired position, the language stream proceeds alone.

C.3 Training Hyperparameters

Both stages update every parameter of the model (language tower, time-series tower, and the global residual attention blocks) under the settings in Table 9. The two stages share the optimizer, schedule, batch size, and objective constants; they differ in context length, step budget, and data mixture. Alignment trains for one epoch. We set the time-series loss weight to $\alpha = 0.5$ in both stages.

Table 9: Training hyperparameters for both stages. Rows spanning both columns are identical across stages.

	Stage 1 (alignment)	Stage 2 (supervised fine-tuning)
Optimizer	AdamW, 8-bit states	
Learning rate	2×10^{-5}, all parameters	
Schedule	constant with warmup	
Warmup steps	500	
Weight decay	0	
Gradient clipping	1.0	
Precision	BF16 mixed precision	
Random seed	3407	
Global batch size	128	
Sequence packing	disabled	
Patch length P	32	
Time-series loss weight α	0.5	
Point loss	squared error	
Quantile set $\mathcal{Q}$	$\{0.1, 0.2, \ldots, 0.9\}$	
Loss cap threshold c	2.0	
Target transform	asinh on causal RevIN residuals	
Context length	2,560	4,096
Training data	2.23M alignment pairs	4.88M final samples
Training steps	17k steps	30k steps

D Prompt Template Examples

Examples are wrapped in Qwen3 chat templates and a series of time-series payloads. The transcript itself contains no time-series numbers: each segment appears in the text as the empty pair `<ts></ts>`, and the N-th occurrence binds to the N-th payload. `<ts>` and `</ts>` are added to the vocabulary as ordinary tokens, `<stats>` and `</stats>` stay plain text, and only assistant content receives cross-entropy.

A context segment is preceded by `<stats>len=`L`, mean=`μ`, std=`σ`</stats>`, computed on the raw values before normalization, so the text stream carries the physical scale while the patch stream carries only normalized shape. A target segment emits a bare `<ts></ts>` and never receives a statistics block, so the model is not told the scale of the sequence it must produce. Target segments span the full window [history | future] and reuse the mean and standard deviation of their context segment, which keeps every future-aware statistic out of the input, and only the future suffix is scored. Exam-style questions are the one family whose payload keeps raw values, because their options refer to absolute magnitudes.

Each card below is one real alignment row, one per task family. Inside every `<ts></ts>` we plot the payload it binds to: **teal** for observed and context spans, **orange** for the future the assistant must generate.

Univariate description

User: Treat this cloud operations series from Borg Cluster Data 2011 as sampled every 5 minutes.
<stats>len=79, mean=0.0671359, std=0.0062208</stats> <ts></ts>
Provide a brief written summary of the series.

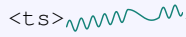
Assistant: A sharp, isolated excursion dominates the middle of the window, contrasting with the otherwise stable baseline. The series holds a steady level near 0.065 until step 39, where it drops abruptly to a local minimum of 0.052. This is immediately followed by a steep recovery to 0.076 at step 40, creating a narrow V-shape. The signal then drifts upward, reaching the window's global maximum of 0.083 at step 55, before settling into a gradual decline toward the end. The amplitude of this central event is significantly larger than the ambient noise observed in the first and last thirds of the trace.

Context-rich description

User: Describe the time series of fatalities due to a specific severity level in Porterville city. The data spans from 2003 to 2008 with a yearly frequency, covering a total population of 41,774.
<stats>len=6, mean=3.167, std=1.067</stats> <ts>  </ts>
Relevant context: Time series of fatalities (severity: Killed) in Porterville city, CA. Data range: 2003-2008. Frequency: Yearly. Total population: 41,774.

Assistant: Porterville city recorded annual fatalities (severity: Killed) between 2003 and 2008 among a population of 41,774. The series began at 3.0 in 2003 and reached its minimum of 2.0 in 2004. The data then exhibited high volatility, spiking to 4.0 in 2005 before dropping back to 2.0 in 2006. Following this trough, the series trended upward, reaching a net increase of 2.0 over the period and closing at its maximum value of 5.0 in 2008.

Time-series question answering

User: You are given a pair of synthetic time series generated by procedural time-series templates.
Time series 1: <stats>len=296, mean=3.11567, std=2.01952</stats>
<ts></ts>
Time series 2: <stats>len=236, mean=2.34241, std=1.70843</stats>
<ts></ts>
Instruction: Identify which time series contains the anomaly and describe the anomaly type based on the visible statistics and series patterns. Respond with one concise declarative sentence.

Assistant: Time series 1 has a speed up/down anomaly, meaning the period of its cyclic components differs from the rest of the series.

Multivariate relational understanding

User: In a Web Servers system, there are 13 metrics:
Request Rates is of length 256: <stats>len=256, mean=-0.4419,
std=0.420386</stats> <ts></ts>;
Backend Server Latency is of length 256: <stats>len=256, mean=5.0224,
std=1.20905</stats> <ts></ts>;
[6 metrics elided, same form]
Security Event Logs is of length 256: <stats>len=256, mean=-119.453,
std=155.282</stats> <ts></ts>;
[4 metrics elided]. Analyze the trend of Security Event Logs in detail.

Assistant: From point 11 to point 55, there is an increasing trend. From point 55 to point 126, there is a decreasing trend. From point 126 to point 188, there is an increasing trend. From point 188 to point 245, there is a decreasing trend.

Contextual forecasting

User: We have been tracking our annual volume for the last five years: `<stats>len=5, mean=4165.4, std=175.205</stats>` `<ts>`  `</ts>`. Historically, we've seen similar growth patterns during previous expansion phases where a slow start was followed by a rapid surge in demand. Based on current market projections and our scaling strategy, we expect a similar acceleration over the next five years, starting with a modest increase to around 4,701 in year 6 and climbing steeply to reach approximately 5,622 by year 10. Please provide the full 10-year sequence, including the observed history and the forecasted values.

Assistant: `<ts>`  `</ts>`

Controllable forecasting, three sibling rows

User: Local tech monitors are tracking weekly visitor traffic for the site, with current data showing `<stats>len=6, mean=525.667, std=302.232</stats>` `<ts>`  `</ts>`. With a major marketing push and site relaunch peaking in about four weeks, officials expect a significant surge in traffic, potentially reaching around 869 visitors. To assist with server capacity planning, please provide the full sequence of weekly traffic, including the observed data and the forecast for the next 5 weeks.

Assistant: `<ts>`  `</ts>`

User: Team,
I'm reviewing our weekly page views for the current seasonal cycle to coordinate our upcoming resource allocation. We've seen some volatility lately, but typically this period involves a dip followed by a recovery as our annual event nears.
Based on the last six weeks `<stats>len=6, mean=525.667, std=302.232</stats>` `<ts>`  `</ts>`, we expect a similar pattern. We anticipate traffic hitting a low of around 348 next week, climbing to a peak of about 661 by week four, and then dropping back to roughly 339 in week five.
Please provide the full sequence of 11 weekly values, including the observed data and the five-week forecast, so we can finalize the schedule.

Assistant: `<ts>`  `</ts>`

User: Weekly Page View Log – Landing Page
Current Status: Tracking traffic trends for the landing page prior to decommissioning.
Recent Activity: `<stats>len=6, mean=525.667, std=302.232</stats>` `<ts>`  `</ts>`
Note: This page is scheduled for decommissioning starting next week. Traffic is expected to drop immediately to nearly zero and remain flat once the URL redirect is active.
Please provide the completed 11-week sequence, including the observed data and the projected values for the final 5 weeks.

Assistant: `<ts>`  `</ts>`

E EVALUATION PROTOCOLS

For every public benchmark, we use its official split or released evaluation set and follow its official evaluation protocol and input-length setting unless stated otherwise. Ctrl-F uses our frozen evaluation set. Each paragraph below states whether baseline results are taken from an existing study or evaluated by us.

TemporalBench. TemporalBench is a multi-domain benchmark over real numerical series that separates historical-structure interpretation (T1), context-free prediction (T2), context-grounded reasoning (T3), and event-conditioned forecasting (T4), testing whether agents can align temporal patterns with external context and adapt when conditions change. Multiple-choice tasks use accuracy; forecasting uses domain-specific MAE or sMAPE variants, where sMAPE normalizes absolute error by the magnitudes of the target and prediction and MIMIC uses the benchmark's observation-

weighted aggregation. We evaluate every model ourselves on the official TemporalBench evaluation set (Weng et al., 2026) (Table 14).

TimeSeriesExam. TimeSeriesExam is a configurable, procedurally generated multiple-choice exam in which one or two controlled synthetic series, represented as plots or serialized values, are paired with answer options and an in-context example to test pattern and noise understanding, anomaly detection, comparative reasoning, and Granger-causality reasoning. We use the official evaluation set (Cai et al., 2024) and take the shared baseline results from the TS-Reasoner paper (Yu et al., 2025). GPT-5.4, Gemini-2.5-Flash, Qwen3-8B, TimeOmni-1-7B, TimeOmni-VL, and TimeBraid are additional evaluations performed by us on the same released set with its scorer (Table 10). Qwen3-8B receives the raw series as numerical text, without plots or in-context examples, and is scored with the official flexible rule; TimeBraid receives native time-series input.

TSAQA. TSAQA formulates each instance as a numerical series, natural-language context, and a true-or-false, multiple-choice, or puzzling question, spanning anomaly detection and classification together with characterization, comparison, data transformation, and temporal-relation reasoning. Puzzling questions ask the model to reorder four shuffled successor patches and are scored by the fraction placed in the correct position. We use the official test set (Jing et al., 2026); except for GPT-5.4, the LLM and finetuned baseline results are taken from the TSAQA paper, while GPT-5.4, the time-series language models, and the unified models are evaluated by us on the same 42,000 cases with the paper-defined scoring rule (Table 11).

CaTS-Bench. In its primary captioning task, CaTS-Bench pairs raw time-indexed values with contextual metadata, a line plot, and a standardized instruction, asking models to synthesize numerical, visual, and semantic evidence into a grounded description evaluated against human-rewritten references. Embedding and lexical metrics measure semantic and wording consistency with the reference; Numeric Fidelity extracts numbers from the reference and output and combines matched-number accuracy with reference-number recall. We use the official human-rewritten evaluation set (Zhou et al., 2026), take the available baseline results from the CaTS-Bench paper, and evaluate GPT-5.4, the time-series language models, and the unified models ourselves on the same set with the six released-formula local metrics (Table 12). We evaluate Gemini-2.5-Flash (`google/gemini-2.5-flash`) and GPT-4.1 (`openai/gpt-4.1`) on all 1,746 cases using the released image prompt branch with the JPEG line plot, numerical series, and contextual metadata. Qwen3-8B (`qwen/qwen3-8b`, thinking disabled) and LLaMA3.1-8B-Instruct (`meta-llama/llama-3.1-8b-instruct`) instead use the released no-image prompt branch: both receive the numerical series and contextual metadata as text, without a plot. These four new rows use the same ten-domain macro aggregation of the six local metrics.

CiK. CiK comprises 71 probabilistic forecasting tasks across seven domains, each paired with historical, future, covariate, causal, or intemporal natural-language context that is essential beyond the observed numerical history, and tests whether models integrate both modalities. RCRPS emphasizes context-relevant future regions, penalizes violations of textual constraints, and normalizes scores by task scale; we aggregate it using the official task weights. We evaluate every model ourselves on the official 355-instance CiK evaluation set (Williams et al., 2024) (Table 13).

Time-MMD. Time-MMD evaluates multimodal forecasting across nine real-world domains by pairing numerical histories with temporally aligned historical text at four frequency-dependent horizons, testing whether extra-numerical domain knowledge improves prediction. We report each domain's MSE and MAE macro-averaged over its four horizons. We use the official evaluation split (Liu et al., 2024a). Every row of Table 15 is evaluated by us, with each multimodal forecasting model receiving the window's paired text: GPT4MTS, TaTS, Time-VLM (Zhong et al., 2025), the text-enhanced PatchTST* (VoT style (Wang et al., 2026)), iTransformer*, and RaFT*, and the additional DLinear, Reformer (MMTSFlib style (Liu et al., 2024a)), and Time-LLM runs. TimeBraid, all time-series foundation models, and MIGAS-1.5 use a fixed 128-step lookback; the text-conditioned baselines, ChatTime, and Aurora use the official lookbacks of 8, 36, and 96. We select a single text-mixing weight of $\lambda = 0.3$ on the validation split and then hold it fixed across all nine domains and history lengths, rather than tuning it separately for each dataset; this uniform setting simplifies evaluation and tests generality across domains.

CGTSF. CGTSF uses the chronological 6:2:2 splits of MSPG, LEU, and PTF, pairing histories with leakage-controlled background, weather-forecast, and calendar information to test the auxiliary role of text in forecasting at four history lengths per dataset. We standardize errors using training-split statistics and report Macro MSE and Macro MAE as the equal averages across the twelve dataset×history-length settings. We evaluate every model ourselves on the released CGTSF datasets using the paper-defined splits (Wang et al., 2025a) (Table 16). We adopt the same implementation of the dataset-specific baselines as in the Time-MMD evaluation.

CAF. CAF-7M evaluates context-aided probabilistic forecasting on verified history–scenario–future windows from datasets reserved for testing, with contexts designed to provide information complementary to the numerical history. Normalized CRPS evaluates the full predictive distribution after adjusting for target scale. We use the official evaluation split (Zheng et al., 2026); DoubleCast and TimeLLM are taken from the CAF-7M paper, and all remaining rows are evaluated by us under the same official protocol (Table 17).

Ctrl-F. Ctrl-F is our controlled-future test set, built on numerical histories replayed from held-out GIFT-Eval test anchors. Each history is paired with three deliberately separated synthetic future siblings and a condition identifying the intended continuation, testing whether forecasts follow the condition rather than the shared history alone. The sibling futures are constructed counterfactual targets rather than official GIFT targets, generated with the same procedure used for the control rows of the alignment corpus. Top-1 is correct when a prediction has lower MAE to its intended sibling than to either alternative, with a chance level of 33.33%. We evaluate every model ourselves on the fixed Ctrl-F evaluation set using the same evaluation protocol (Table 18).

Ctrl-F instance: one 22-step hospital history, three sibling conditions at horizon 12

User: `(history=22, horizon=12)`
Use this continuation condition: The series drops sharply to a local minimum at future step 3, followed by a volatile recovery that peaks at future steps 4 and 9 before ending with a slight upward trend.
History: `<stats>len=22, mean=53.1818, std=8.89656</stats>`
`<ts>`  `</ts>`

Assistant: `<ts>`  `</ts>`

User: *[identical, except]* Use this continuation condition: Maintain a strong downward drift across future steps 1 to 12.

Assistant: `<ts>`  `</ts>`

User: *[identical, except]* Use this continuation condition: Create a local downward pulse in future steps 3 to 8. Cap upper peaks in future steps 1 to 12.

Assistant: `<ts>`  `</ts>`

GIFT-Eval. GIFT-Eval evaluates general zero-shot point and probabilistic forecasting across 97 dataset–frequency–horizon configurations spanning univariate and multivariate series, seven domains, ten sampling frequencies, and short-to-long horizons, targeting generalization across heterogeneous forecasting settings. The official leaderboard's MASE scales absolute point error by seasonal-naive error, while CRPS evaluates the predictive distribution; both are normalized against Seasonal Naive per configuration and aggregated by geometric mean, so lower is better and 1.0 matches Seasonal Naive. We take baseline results from the official leaderboard (Aksu et al., 2024) and evaluate TimeBraid using the last 2,880 history points (Table 4).

Long-Term Forecasting. Long-term forecasting follows the standard multivariate setting on ETTm1, ETTm2, ETTh1, ETTh2, and the 21-variable Jena Weather dataset at horizons

$\{96, 192, 336, 720\}$, testing numerical-history-only extrapolation of trends, periodic structure, and long-range dependencies (Zhou et al., 2021; Wu et al., 2021). TimeBraid-2.5B and the time-series foundation models are evaluated zero-shot, while the task-specific supervised baselines are trained on each dataset. Most baseline results are taken from the Sundial and Time-MoE papers (Liu et al., 2025b; Shi et al., 2025). We additionally evaluate TimesFM-2.5 and Chronos-2 ourselves, using the same 2,880-step lookback windows as TimeBraid (Table 20).

F Detailed Experimental Results

This appendix reports the benchmark-level results summarized in the main text. Unless stated otherwise, higher is better for accuracy and similarity metrics, whereas lower is better for forecasting errors and ranks. Across tables, bold, underline, and a dagger mark the best, second-best, and third-best distinct values among comparable models shown in each table, excluding Closed-source Reference Models; ties share a rank. Dashes mark unavailable results.

F.1 Time-Series Understanding

Table 10: TimeSeriesExam accuracy (%) by category: pattern recognition (PR), noise understanding (NU), anomaly detection (AD), similarity analysis (SA), causality analysis (CA), and overall (OA).

Model	PR	NU	AD	SA	CA	OA
Closed-source Reference Models						
GPT-5.4	69.34	64.29	70.37	74.17	50.00	67.83
GPT-4o	59.03	55.17	53.49	62.83	31.75	55.96
Gemini-2.5-Flash	54.97	54.76	47.22	63.33	41.67	53.89
GPT-4o (vision)	67.12	62.07	62.79	64.60	26.98	62.12
GPT-4.1 (vision)	69.81	68.97	68.22	75.22	41.27	67.89
Open-source Large Language Models						
DeepSeek-Chat	65.23†	55.17	**52.71**	63.71†	42.86	59.89†
Llama-3.1-8B-Instruct	37.73	37.93	30.23	36.28	28.57	35.52
Qwen2.5-7B-Instruct	47.17	47.13	41.86	53.10	41.27	46.66
Qwen3-8B	49.45	54.76	33.33	51.67	43.06†	47.45
Open-source Vision-Language Models						
Qwen2.5-VL-7B-Instruct	25.34	32.18	19.38	42.48	12.70	26.61
InternVL3-8B	50.13	52.87	43.41	54.87	38.09	49.01
Time-Series Language Models						
ChatTS	50.13	50.57	<u>50.38</u>	61.95	34.92	50.72
TS-Reasoner-7B	52.29	<u>60.92</u>	**52.71**	<u>64.60</u>	41.27	54.26
TimeOmni-1-7B	42.82	48.81	22.22	24.17	19.44	35.25
Unified Models						
ChatTime-7B	42.85	49.42	35.65	44.24	34.92	41.94
TimeOmni-VL	50.83	57.14	36.11	54.17	41.67	49.06
TimeBraid-1.2B (Ours)	55.25	48.81	37.04	54.17	38.89	50.13
TimeBraid-2.5B (Ours)	<u>67.13</u>	59.52†	46.30†	58.33	**48.61**	<u>60.05</u>
TimeBraid-6.7B (Ours)	**69.89**	**65.48**	42.59	**69.17**	<u>47.22</u>	**63.14**

Table 11: TSAQA test-set accuracy (%). A.D.: anomaly detection; CLS: classification; TF/MC/PZ: true-or-false, multiple-choice, and puzzling/ordering formats. TSAQA-specific SFT models are LoRA fine-tuned on the TSAQA training split (Jing et al., 2026). Ranks include these models and exclude Closed-source Reference Models; ties share a rank.

Model	A.D.	CLS	Characterization		Comparison		Data Transform		Temporal Relation			Overall
	TF	MC	TF	MC	TF	MC	TF	MC	TF	MC	PZ	
Closed-source Reference Models												
GPT-5.4	53.32	49.57	82.08	81.98	77.59	74.47	65.79	54.94	74.12	82.96	51.02	63.10
GPT-4.1	55.85	50.38	92.97	89.36	83.57	76.99	54.36	51.13	65.90	79.09	45.77	62.82
GPT-4o	54.32	47.20	88.15	84.15	78.61	69.07	60.66	53.24	62.25	75.58	45.61	60.73
Claude-3.5-Sonnet	51.27	41.23	74.39	78.45	66.59	74.14	65.79	57.07	82.05	82.15	54.56	61.19
Gemini-2.5-Flash	52.08	49.07	85.48	81.08	77.79	72.21	63.62	60.17	75.05	84.49	60.84	65.08
Open-source Large Language Models												
Qwen3-8B	50.60	50.52	77.35	66.87	71.04	63.21	52.43	34.46	65.22	67.14	21.93	51.04
LLaMA3.1-8B	54.92	50.20	68.10	62.26	67.84	49.98	51.90	36.56	54.82	40.95	6.80	44.93
Ministral-8B	53.35	34.08	71.06	63.93	47.54	52.90	50.70	25.28	50.58	33.88	30.77	44.65
Qwen3-0.6B	50.40	35.83	62.00	48.78	58.03	37.51	49.03	23.62	51.99	37.33	13.38	39.06
LLaMA3.2-1B	49.47	39.48	63.74	52.55	61.02	36.82	48.87	4.20	48.97	5.44	6.76	35.70
Gemma3-1B	49.15	49.83	63.74	47.71	61.19	43.37	49.37	24.88	49.42	25.84	23.97	43.03
Large Language Models (TSAQA-Specific SFT)												
LLaMA3.1-8B	**91.02**	**91.27**	**92.44**	83.68	**86.72**	**79.31**	**90.17**	**86.62**	**96.94**	97.41	**67.68**	**85.26**
Qwen3-8B	87.70	90.05	92.37	**85.42**	86.55	79.08	89.84	84.99†	96.84	**97.56**	66.21	84.29
Ministral-8B	71.56	74.28	91.31	80.78	84.14	74.63	75.15	71.61	94.07†	94.15†	56.82	74.74
Qwen3-0.6B	83.68†	85.78†	89.38	74.87	80.65	64.84	80.51	73.28	93.92	93.79	63.34†	78.32
LLaMA3.2-1B	83.08	83.83	87.71	74.37	78.61	60.88	68.09	51.67	91.39	88.81	57.53	73.48
Gemma3-1B	83.10	84.05	87.88	72.54	78.61	59.31	64.06	45.23	91.00	88.05	42.92	69.70
Time-Series Language Models												
ChatTS	49.42	52.08	69.20	60.43	67.13	58.27	49.97	30.29	54.04	50.61	33.59	49.83
TimeOmni-1-7B	50.98	56.47	39.93	59.06	40.34	44.55	50.70	28.39	50.58	40.79	30.93	44.40
Unified Models												
ChatTime-7B	50.20	43.47	48.92	27.29	50.49	25.14	49.53	25.15	49.81	23.80	27.87	38.38
TimeOmni-VL	50.82	51.15	41.92	44.18	40.71	28.87	53.23	23.55	58.33	23.63	15.52	39.27
TimeBraid-1.2B (Ours)	82.58	80.78	89.71	77.68	81.80	69.20	80.28	77.92	89.62	89.43	45.04	76.61
TimeBraid-2.5B (Ours)	83.23	79.25	90.64	80.05	83.23	72.86	83.41	82.92	89.71	91.71	48.36	78.31
TimeBraid-6.7B (Ours)	83.43	85.78†	91.41†	81.78†	85.94†	75.12†	86.98†	86.39	92.05	92.04	49.39	80.65†

Table 12: Full results on the CaTS-Bench human-rewritten split: five metrics of alignment with reference captions and the released Numeric Fidelity score, all in $[0, 1]$; higher is better. Rows marked finetuned are trained on the CaTS-Bench training split.

Model	Lexical and Semantic Metrics					Numeric
	DeBERTa F1	SimCSE	BLEU	ROUGE-L	METEOR	
Closed-source Reference Models						
GPT-5.4	0.660	0.843	0.064	0.239	0.264	0.778
Gemini 2.5 Flash	0.694	0.887	0.100	0.276	0.311	0.802
GPT-4.1	0.672	0.886	0.071	0.238	0.306	0.757
Gemini 2.0 Flash	0.694	0.884	0.113	0.283	0.304	0.757
GPT-4o	0.685	0.886	0.090	0.259	0.314	0.739
Open-source Text-Only Language Models (no plot)						
Qwen3-8B	0.670	0.865	0.082	0.243	0.283	0.678
LLaMA3.1-8B-Instruct	0.667	0.859	0.081	0.249	0.284	0.697
Open-source Vision-Language Models						
InternVL 2.5 38B	0.682	0.867	0.085	0.262	0.294	0.740
Gemma 3 27B	0.680	0.883†	0.089	0.255	**0.307**	0.745†
LLaVA v1.6 (finetuned)	0.644	0.802	0.064	0.234	0.250	0.589
Idefics 2 (finetuned)	**0.713**	0.885	**0.131**	**0.325**	0.303	0.748
QwenVL (finetuned)	0.678	0.863	0.090	0.257	0.272	0.669
Llama 3.2 Vision (finetuned)	0.672	0.851	0.087	0.266	0.295†	0.740
Phi-4 Multimodal Instruct (finetuned)	0.664	0.862	0.078	0.243	0.290	0.703
Time-Series Language Models						
ChatTS	0.614	0.665	0.040	0.191	0.198	0.648
TimeOmni-1-7B	0.634	0.796	0.046	0.205	0.231	**0.769**
Unified Models						
ChatTime-7B	0.371	0.234	0.004	0.032	0.030	0.077
TimeOmni-VL	0.598	0.568	0.027	0.179	0.165	0.370
TimeBraid-1.2B (Ours)	0.706	0.880	0.120	0.305	0.288	0.666
TimeBraid-2.5B (Ours)	0.708†	0.885	0.121†	0.307†	0.289	0.670
TimeBraid-6.7B (Ours)	0.712	**0.886**	0.123	0.313	0.292	0.684

F.2 TIME-SERIES FORECASTING

F.2.1 CONTEXTUAL REASONING FORECASTING

Table 13: Results on CiK: RCRPS weighted by the official task weights, reported with standard errors and stratified by context type. Asterisks mark models that do not use natural-language context.

MODEL	AVERAGE RCRPS	INTEMPORAL INFORMATION	HISTORICAL INFORMATION	FUTURE INFORMATION	COVARIATE INFORMATION	CAUSAL INFORMATION
Closed-source Reference Models						
Gemini-2.5-Flash	0.110 ± 0.002	0.134 ± 0.003	0.142 ± 0.001	0.036 ± 0.001	0.116 ± 0.003	0.252 ± 0.012
GPT-5.4	0.145 ± 0.000	0.182 ± 0.001	0.116 ± 0.001	0.049 ± 0.000	0.160 ± 0.001	0.414 ± 0.001
GPT-4o	0.257 ± 0.001	0.305 ± 0.002	0.135 ± 0.001	0.159 ± 0.000	0.238 ± 0.001	0.613 ± 0.006
GPT-5.4-mini	0.277 ± 0.000	0.302 ± 0.001	0.173 ± 0.001	0.212 ± 0.000	0.261 ± 0.001	0.553 ± 0.000
Open-source Large Language Models						
DeepSeek-Chat	**0.225 ± 0.004**	0.286 ± 0.006	0.141 ± 0.001	**0.089 ± 0.008**	**0.166 ± 0.001**	**0.431 ± 0.005**
Qwen-2.5-7B-Inst	0.344 ± 0.001	0.368 ± 0.002	0.195 ± 0.001	0.283 ± 0.002†	0.333 ± 0.001	0.562 ± 0.001
Llama-3-8B-Inst	0.476 ± 0.003	0.573 ± 0.005	0.245 ± 0.001	0.301 ± 0.000	0.486 ± 0.004	0.526 ± 0.001
Time-Series Foundation Models						
Moirai-2*	0.288 ± 0.002	**0.252 ± 0.003**	0.119 ± 0.002	0.366 ± 0.004	0.244 ± 0.003	0.435 ± 0.008†
TimesFM*	0.295 ± 0.002	0.265 ± 0.003	**0.114 ± 0.001**	0.373 ± 0.002	0.251 ± 0.002	0.474 ± 0.006
Sundial*	0.314 ± 0.001	0.285 ± 0.002	0.119 ± 0.001	0.381 ± 0.001	0.263 ± 0.001	0.486 ± 0.004
TimeMoE*	0.327 ± 0.001	0.303 ± 0.001	0.150 ± 0.000	0.387 ± 0.002	0.271 ± 0.001	0.488 ± 0.000
Chronos-2*	0.295 ± 0.003	0.271 ± 0.005	0.183 ± 0.007	0.354 ± 0.004	0.243 ± 0.003†	0.445 ± 0.012
Traditional Time-Series Models						
DLinear*	0.337 ± 0.001	0.337 ± 0.001	0.230 ± 0.000	0.345 ± 0.003	0.284 ± 0.001	0.548 ± 0.000
iTransformer*	0.359 ± 0.001	0.347 ± 0.001	0.230 ± 0.000	0.394 ± 0.003	0.305 ± 0.001	0.565 ± 0.000
ARIMA*	0.501 ± 0.003	0.620 ± 0.003	0.189 ± 0.004	0.319 ± 0.006	0.275 ± 0.002	0.455 ± 0.005
Multimodal Forecasting Models						
UniTime	0.368 ± 0.002	0.453 ± 0.002	0.156 ± 0.000	0.197 ± 0.004	0.392 ± 0.002	0.433 ± 0.001
Unified Models						
ChatTime	0.344 ± 0.000	0.332 ± 0.000	0.191 ± 0.000	0.334 ± 0.000	0.315 ± 0.000	0.436 ± 0.000
TimeOmni-VL	0.655 ± 0.028	0.678 ± 0.043	0.260 ± 0.008	0.648 ± 0.023	0.586 ± 0.035	0.818 ± 0.132
TimeBraid-1.2B (Ours)	0.301 ± 0.002	0.278 ± 0.003	0.138 ± 0.002	0.359 ± 0.004	0.250 ± 0.002	0.472 ± 0.006
TimeBraid-2.5B (Ours)	0.289 ± 0.002†	0.270 ± 0.002	0.142 ± 0.002	0.337 ± 0.003	0.239 ± 0.002	0.468 ± 0.005
TimeBraid-6.7B (Ours)	0.294 ± 0.002	0.267 ± 0.003†	0.134 ± 0.003†	0.359 ± 0.004	0.243 ± 0.003†	0.437 ± 0.009

F.2.2 TEMPORALBENCH

Table 14: TemporalBench results under the official evaluation setup: (a) accuracy on the 16 multiple-choice tasks, with Average the unweighted macro-average, and (b) forecasting error, not averaged across datasets because the metrics differ.

Model	FreshRetailNet				PSML				Causal Chambers				MIMIC				Average
	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4	
Closed-source Reference Models																	
GPT-5.4	45.45%	28.03%	41.48%	42.42%	50.50%	31.33%	33.20%	57.33%	18.67%	53.33%	37.20%	45.33%	35.11%	29.08%	35.56%	31.91%	38.50%
GPT-4.1	38.64%	31.82%	45.45%	38.64%	48.50%	27.33%	40.00%	53.33%	12.00%	46.00%	48.80%	41.33%	18.62%	29.08%	42.68%	28.37%	36.91%
GPT-4o	63.07%	16.67%	28.98%	39.39%	69.00%	23.33%	35.20%	36.67%	10.00%	22.67%	34.00%	42.00%	46.81%	19.86%	0.00%	29.08%	32.30%
Gemini-2.5-Flash	59.66%	22.73%	29.55%	38.64%	72.50%	23.33%	22.00%	46.67%	10.67%	45.33%	37.20%	44.00%	42.55%	29.08%	0.00%	29.79%	34.61%
Claude-Sonnet-4	55.11%	29.55%	34.66%	43.94%	50.00%	28.67%	33.20%	40.00%	15.33%	60.67%	37.60%	58.67%	35.11%	21.99%	0.00%	32.62%	36.07%
Open-source Large Language Models																	
DeepSeek-Chat	**63.64%**	15.91%	28.98%	34.09%	**57.00%**	25.33%	32.00%	56.67%	14.67%	28.00%	33.60%	32.00%	28.19%	24.11%	0.00%	23.40%	31.10%
Qwen3-8B	36.36%	30.30%†	51.70%	40.15%†	38.50%	29.33%	**48.80%**	54.67%†	17.33%	15.33%	42.80%	15.33%	16.49%	21.28%	38.49%	21.99%	32.43%
Llama-3.1-8B	29.55%	25.76%	46.59%†	22.73%	30.00%	23.33%	38.00%†	50.67%	15.33%	12.67%	27.60%	13.33%	13.30%	17.02%	30.54%	17.73%	25.88%
Time-Series Language Models																	
ChatTS	15.91%	18.18%	31.82%	34.09%	46.00%†	19.33%	38.40%	54.00%	22.67%†	7.33%	35.20%	11.33%	30.85%	11.35%	0.00%	21.99%	24.90%
TimeOmni-1-7B	26.70%	21.97%	29.55%	26.52%	33.50%	24.00%	26.00%	40.00%	21.33%	23.33%	30.40%	19.33%	35.64%	27.66%	0.00%	24.82%	25.67%
Unified Models																	
ChatTime-7B	31.25%	27.27%	31.82%	23.48%	37.50%	27.33%	30.00%	19.33%	23.33%	20.67%	32.80%	26.00%	**40.43%**	26.24%	27.20%	**30.50%**	28.45%
TimeOmni-VL	19.89%	39.39%	43.75%	43.18%	52.00%	29.33%	11.20%	44.67%	18.00%	28.67%	45.20%†	31.33%	18.09%	**33.33%**	33.47%	29.79%	32.32%
TimeBraid-1.2B (Ours)	17.61%	**42.42%**	41.48%	**50.00%**	14.50%	32.67%†	19.20%	18.67%	21.33%	**61.33%**	36.00%	**61.33%**	34.57%	29.08%†	35.98%†	**30.50%**	34.17%†
TimeBraid-2.5B (Ours)	34.66%†	25.76%	**52.27%**	39.39%	14.50%	36.00%	16.80%	52.00%	16.00%	50.67%	**47.20%**	54.00%†	35.11%†	32.62%	35.15%	29.79%	35.74%
TimeBraid-6.7B (Ours)	30.68%	27.27%	44.32%	37.88%	18.00%	**41.33%**	23.60%	**58.00%**	**24.00%**	49.33%†	46.40%	59.33%	33.51%	17.73%	**46.86%**	26.95%†	**36.58%**

(a) Multiple-choice accuracy.

Model	FreshRetailNet		PSML		Causal Chambers		MIMIC	
	T2	T4	T2	T4	T2	T4	T2	T4
Closed-source Reference Models								
GPT-5.4	0.132	0.129	0.236	0.243	3.287	4.279	9.853	10.005
GPT-4o	0.127	0.234	0.333	0.435	1.988	2.755	15.913	16.860
Gemini-2.5-Flash	0.106	0.118	0.304	0.332	2.244	2.370	9.902	12.557
Claude-Sonnet-4	0.116	0.182	0.253	0.318	2.561	2.725	9.098	13.545
Open-source Large Language Models								
DeepSeek-Chat	**0.113**	0.135	0.333	0.374	**2.532**	**2.126**	14.981	14.789
Time-Series Foundation Models								
TimesFM-2.5	**0.113**	**0.115**	0.229	0.231	4.850	4.846	9.215	9.223
Chronos-2	0.116†	0.119	0.207	0.209	4.812	4.811	9.643	9.647
Moirai-2	0.115	0.116	0.213	0.210	4.766†	4.767†	9.051	9.045
Sundial	0.122	0.121	0.205	0.203†	5.309	5.332	9.973	9.982
TimeMoE	0.119	0.121	**0.198**	0.198	5.331	5.347	9.600	9.632
Multimodal Forecasting Models								
MIGAS-1.5	0.115	**0.115**	0.222	0.221	4.717	4.753	9.553	9.602
Aurora	0.126	0.125	0.204†	0.204	6.918	6.917	10.469	10.475
Unified Models								
ChatTime-7B	0.118	0.143	0.225	0.223	5.397	5.220	10.256	10.379
TimeOmni-VL	1.537	1.587	0.726	0.787	41.261	54.678	77.746	85.100
TimeBraid-1.2B (Ours)	0.117	0.118†	0.205	0.204	5.317	5.316	8.946†	8.996
TimeBraid-2.5B (Ours)	0.120	0.119	0.201	0.203†	5.288	5.307	8.945	9.002†
TimeBraid-6.7B (Ours)	0.116†	0.116	**0.198**	**0.197**	5.304	5.316	**8.931**	**8.972**

(b) Forecasting error on 852 current-leaderboard-dev cases (FreshRetailNet and Causal Chambers: MAE; PSML: sMAPE; MIMIC: OW-sMAPE).

F.2.3 TIME-MMD

Table 15: Time-MMD forecasting with paired text, reporting per-domain MSE and MAE macro-averaged over four horizons. Every multimodal forecasting model receives each window's paired text and runs at the official 8/36/96 lookbacks, as do ChatTime and Aurora; TimeBraid, the time-series foundation models, and MIGAS-1.5 use a fixed 128-step history. Avg Rank averages each model's rank across the nine domains; lower is better.

Unified Models									*Multimodal Forecasting Models*																		*Time-Series Foundation Models*							
Models	TimeBraid-1.2B		TimeBraid-2.5B		TimeBraid-6.7B		ChatTime-7B		Time-LLM		GPT4MTS		TeTS		Time-VLM		PatchTST*		Transformer*		RaFT*		DLinear		Reformer		MIGAS-1.5		Aurora		TimesFM-2.5		Chronos-2		$\text{Moirai}_{\text{Base-2.0}}$		$\text{Sundial}_{\text{Base}}$		Time-MoE-200M	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Agriculture	0.116	0.209	0.137	0.232	0.144	0.235	0.130	0.240	0.098	0.204	0.093	0.189	0.091†	0.196	0.093	0.193†	0.093	0.197	0.164	0.287	0.190	0.322	0.249	0.374	0.497	0.556	**0.086**	**0.185**	0.109	0.208	0.127	0.223	**0.085**	**0.185**	0.102	0.197	0.102	0.195	0.107	0.200
Climate	0.899	0.746†	0.871†	0.729	0.938	0.760	2.279	1.222	1.320	0.938	1.262	0.919	1.307	0.938	1.282	0.930	1.267	0.923	1.247	0.910	1.848	1.092	1.244	0.907	1.131	0.850	0.983	0.791	1.428	0.972	**0.859**	0.729	0.988	0.794	0.949	0.770	0.865	**0.714**	0.935	0.748
Economy	0.025	0.125	0.025	0.125	0.018	0.104	0.071	0.219	0.031	0.143	0.027	0.134	0.016	0.100†	0.020	0.113	0.019	0.109	0.108	0.297	0.177	0.351	0.192	0.382	1.118	0.977	**0.015**	0.098	0.040	0.166	0.017†	0.104	**0.015**	**0.096**	0.020	0.109	0.022	0.119	0.021	0.117
Energy	0.256	0.348	0.235	0.335	**0.231**	**0.332**	0.365	0.460	0.279	0.384	0.266	0.385	0.277	0.387	0.253	0.367	0.269	0.381	0.294	0.407	0.245	0.350	0.341	0.434	0.555	0.571	0.242	0.347	0.367	0.448	0.239†	0.338†	0.262	0.357	0.277	0.368	0.266	0.353	0.277	0.357
Environment	0.516	0.509†	0.507	0.498	0.507	0.513	1.099	0.757	0.519	0.516	0.518	0.517	**0.431**	**0.489**	0.494†	0.521	0.492	0.510	0.519	0.521	0.548	0.553	0.579	0.625	0.523	0.575	0.671	0.591	0.597	0.575	0.569	0.521	0.566	0.527	0.581	0.543	0.561	0.550	0.579	0.562
Health(US)	1.002	0.640	0.956†	0.619†	0.925	0.608	2.516	1.037	1.488	0.827	1.519	0.826	1.665	0.810	1.482	0.820	1.556	0.799	1.726	0.808	1.534	0.873	1.714	0.870	1.567	0.824	1.302	0.728	2.000	1.022	**0.665**	**0.484**	1.217	0.698	1.280	0.763	1.240	0.749	1.229	0.731
Security	108.957†	4.695	108.864	**4.691**	**108.552**	4.715†	144.336	5.700	114.884	5.486	111.210	5.288	120.945	5.656	112.981	5.320	111.667	5.265	119.281	5.588	160.934	6.405	109.860	5.264	119.267	5.682	109.911	4.784	120.737	6.028	114.802	5.150	110.309	4.819	109.829	4.811	111.686	5.010	110.392	4.927
SocialGood	0.808†	**0.351**	0.839	0.368	**0.799**	0.352	1.227	0.576	1.023	0.468	1.014	0.451	1.148	0.464	1.068	0.498	1.110	0.484	1.187	0.468	0.994	0.477	0.979	0.500	1.008	0.638	1.110	0.396	1.155	0.576	0.845	0.365	1.110	0.396	0.859	0.364	0.856	0.361†	0.801	0.364
Traffic	0.148	0.213	**0.145**	0.206†	0.153	0.211	0.505	0.555	0.223	0.299	0.249	0.321	0.227	0.278	0.229	0.305	0.216	0.267	0.216	0.286	0.460	0.525	0.316	0.451	0.319	0.471	0.200	0.216	0.322	0.426	0.152†	**0.179**	0.201	0.217	0.163	0.191	0.160	0.211	0.167	0.225
Avg Rank	5.83†	5.39†	5.11	5.11	**4.50**	**4.78**	18.56	18.61	12.28	12.78	10.50	11.33	11.11	10.17	9.50	11.33	9.72	10.00	14.89	13.50	14.67	16.22	14.17	16.22	15.22	17.28	8.06	6.56	17.00	16.78	6.33	5.44	7.61	6.50	8.67	7.17	7.89	6.61	8.39	8.22

F.2.4 CGTSF

CGTSF is the context-guided forecasting benchmark of ChatTime (Wang et al., 2025a), with three real-world datasets paired with aligned text: MSPG (Melbourne solar power generation), LEU (London electricity usage), and PTF (Paris traffic flow).

Table 16: CGTSF context-guided forecasting (Wang et al., 2025a) on MSPG (Melbourne solar power generation), LEU (London electricity usage), and PTF (Paris traffic flow). Errors are standardized using training-split statistics; per-dataset entries average four history-length settings, and Macro MSE and Macro MAE weight the twelve settings equally. We replicate the official evaluation setting and evaluate every baseline ourselves.

Model	Per-dataset (MSE / MAE ↓)			Macro	
	MSPG	LEU	PTF	MSE ↓	MAE ↓
Time-Series Foundation Models					
Chronos-2	0.649 / 0.388	0.715 / **0.414**	0.230 / 0.285	0.531	0.363†
TimesFM-2.5	0.585 / 0.384	0.777 / 0.442	0.243 / 0.306	0.535	0.377
Sundial	0.763 / 0.507	0.683 / 0.453	0.242 / 0.306	0.563	0.422
Moirai-2	1.000 / 0.590	0.681 / 0.422	0.284 / 0.332	0.655	0.448
Time-MoE-200M	1.138 / 0.713	0.698 / 0.450	0.239 / 0.302	0.692	0.488
Multimodal Forecasting Models					
MIGAS-1.5	0.757 / 0.476	0.887 / 0.550	0.386 / 0.444	0.677	0.490
Aurora	0.789 / 0.562	0.757 / 0.503	0.336 / 0.394	0.627	0.486
Time-LLM	0.583 / 0.476	0.699 / 0.480	0.163 / 0.273†	0.482	0.409
GPT4MTS	0.567 / 0.438	0.694 / 0.481	0.180 / 0.273†	0.481	0.397
Time-VLM	0.818 / 0.644	0.725 / 0.498	0.175 / 0.278	0.573	0.473
PatchTST*	0.602 / 0.479	0.706 / 0.489	0.179 / 0.277	0.496	0.415
iTransformer*	0.737 / 0.507	0.698 / 0.490	0.193 / 0.285	0.543	0.428
RaFT*	1.398 / 0.608	0.701 / 0.503	0.238 / 0.322	0.779	0.478
DLinear	0.619 / 0.470	0.687 / 0.487	0.197 / 0.300	0.501	0.419
Reformer	0.859 / 0.584	0.736 / 0.514	0.269 / 0.355	0.621	0.484
Unified Models					
ChatTime	2.727 / 0.945	1.015 / 0.507	0.359 / 0.373	1.367	0.609
TimeBraid-1.2B (Ours)	0.472† / 0.371†	0.642 / 0.425	0.136 / 0.229	0.417†	0.342
TimeBraid-2.5B (Ours)	0.467 / 0.368	0.644† / 0.423†	**0.133** / **0.225**	0.415	**0.338**
TimeBraid-6.7B (Ours)	**0.459** / **0.362**	**0.641** / 0.424	0.138† / 0.229	**0.413**	**0.338**

We report history-z-normalized errors: per-dataset MSE/MAE average each dataset's four history-length settings, and Macro MSE and Macro MAE equally average the twelve dataset×history-length settings.

F.2.5 CAF (CONTEXT-AIDED FORECASTING)

CAF evaluates whether models use scenario information that complements the observed history while forecasting a predictive distribution. We report normalized CRPS with standard errors on all 904 correct-context test cases and their HARD and EASY strata; the evaluation set is held out from our mixture as summarized in Appendix B.3.

F.2.6 CTRL-F

Ctrl-F tests whether a forecast follows a natural-language future condition when three distinct continuations share the same observed history. The set contains 50 domain-balanced groups across commerce, compute, epidemic, health, and traffic, whose histories are replayed from held-out GIFT-Eval test anchors. Each group contributes three original-condition windows, three paraphrased-condition windows, and one shared blank-text control, yielding 350 instances in total. We report history-normalized forecasting error on the 150 original-condition windows and Top-1 sibling retrieval against the three candidate futures (chance $= 33.33\%$).

Table 17: CAF normalized CRPS on all 904 correct-context examples and the hard/easy strata (lower is better; mean $\pm$ s.e.).

Model	Normalized CRPS ↓		
	ALL	HARD	EASY
Closed-source Reference Models			
GPT-5.4	0.233 ±0.011	0.225 ±0.012	0.243 ±0.019
GPT-4o	0.234 ±0.011	0.228 ±0.012	0.241 ±0.018
Gemini-2.5-Flash	0.247 ±0.012	0.247 ±0.016	0.246 ±0.018
Open-source Large Language Models			
DeepSeek-Chat	0.267 ±0.013	0.254 ±0.015	0.283 ±0.021
Time-Series Foundation Models			
Moirai-2	0.275 ±0.014	0.299 ±0.019	0.248 ±0.020
Chronos-2	0.277 ±0.015	0.298 ±0.019	0.252 ±0.022
TimesFM-2.5	0.281 ±0.016	0.293 ±0.020	0.267 ±0.025
Sundial	0.331 ±0.017	0.340 ±0.021	0.321 ±0.028
Time-MoE-200M	0.414 ±0.023	0.388 ±0.023	0.445 ±0.042
Multimodal Forecasting Models			
MIGAS-1.5	0.346 ±0.017	0.379 ±0.023	0.307 ±0.025
Aurora	0.460 ±0.021	0.428 ±0.021	0.497 ±0.039
DoubleCast	0.231 ±0.001	0.251 ±0.001†	**0.204 ±0.002**
TimeLLM	0.501 ±0.001	0.428 ±0.001	0.587 ±0.001
Unified Models			
ChatTime-7B	0.345 ±0.015	0.375 ±0.020	0.310 ±0.022
TimeOmni-VL	0.489 ±0.025	0.425 ±0.027	0.565 ±0.043
TimeBraid-1.2B (Ours)	0.235 ±0.012†	0.243 ±0.016	0.225 ±0.018†
TimeBraid-2.5B (Ours)	0.235 ±0.012†	0.243 ±0.014	0.226 ±0.019
TimeBraid-6.7B (Ours)	**0.225 ±0.012**	**0.231 ±0.015**	0.219 ±0.018

Table 18: Ctrl-F on 150 correct-arm windows, reporting history-normalized MSE and MAE with correct-sibling Top-1 accuracy (chance $= 33.33\%$).

Model	MSE↓	MAE↓	Top-1 (%)↑
Closed-source Reference Models			
GPT-5.4	5.188	1.563	51.33
GPT-4o	5.447	1.591	60.00
Gemini-2.5-Flash	6.624	1.771	56.00
Open-source Large Language Models			
DeepSeek-Chat	6.405	1.692	**59.33**
Time-Series Foundation Models			
Moirai-2	8.425	1.987	33.33
Chronos-2	8.597	2.004	33.33
TimesFM-2.5	8.784	2.025	33.33
Sundial	8.829	2.074	33.33
Time-MoE-200M	9.077	2.114	33.33
Multimodal Forecasting Models			
MIGAS-1.5	8.604	2.009	33.33
Aurora	9.091	2.141	33.33
Unified Models			
ChatTime	9.165	2.140	33.33
TimeOmni-VL	10.387	2.367	32.00
TimeBraid-1.2B (Ours)	6.826†	1.768†	40.67†
TimeBraid-2.5B (Ours)	6.972	1.781	40.67†
TimeBraid-6.7B (Ours)	**6.232**	**1.689**	43.33

F.2.7 UNIMODAL FORECASTING

Table 19: Expanded GIFT-Eval comparison for the methods in Table 4 and statistical comparators. Normalized MASE and CRPS are geometric means after division by Seasonal Naive within each of the 97 official configurations; lower is better. TimeBraid-2.5B uses a 2,880-step history.

Model	MASE↓	CRPS↓
Statistical Methods		
Naive	1.270	1.591
Seasonal Naive	1.000	1.000
Auto ARIMA	1.074	0.912
Task-Specific (Supervised)		
DeepAR	1.343	0.853
TiDE	1.091	0.772
N-BEATS	0.938	0.816
PatchTST	0.849	0.587
iTransformer	0.893	0.620
Pretrained Forecasting Methods		
TimesFM-2.5	0.705	0.490
TabPFN-TS	0.771	0.544
Chronos-2	0.698	0.485†
Moirai-2	0.728	0.516
Sundial Base	0.750	0.559
TiRex	0.716	0.488
Toto-2.0 FnF	**0.676**	**0.463**
Chronicle	1.053	0.754
Timer-S1	0.693†	0.485†
TTM-R3-PT	0.724	0.520
Kairos-50M	0.742	0.548
TiRex-2-Pretrained	<u>0.678</u>	<u>0.467</u>
Moirai-1.1-R-large	0.875	0.599
Unified Model		
TimeBraid-2.5B (Ours)	0.763	0.546

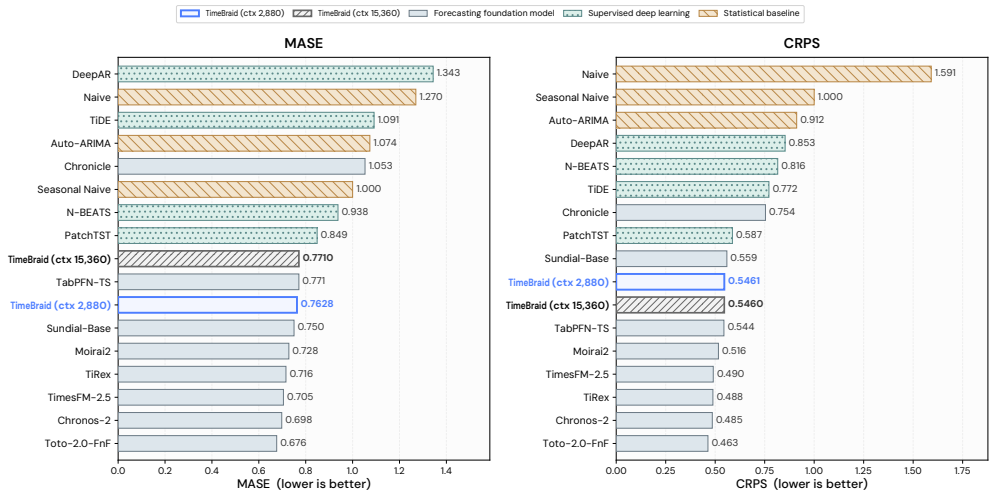


Figure 7: Aggregate GIFT-Eval forecasting performance across models and context lengths. Lower is better.

Table 20: Per-horizon long-term forecasting on ETT and Weather, reporting MSE and MAE at horizons $\{96, 192, 336, 720\}$ and their average; lower is better. TimeBraid-2.5B and the time-series foundation models are evaluated zero-shot; the task-specific baselines use full-shot training on each dataset. TimeBraid-2.5B uses a 2,880-step lookback; the baselines retain their official input lengths. Each dataset's Avg row averages errors over the four horizons; Avg Rank averages each model's rank across the twenty dataset–horizon settings.

Models		*TimeBraid* TimeBraid-2.5B (Ours)		*Time-Series Foundation Models* $\text{Sundial}_{\text{Large}}$ (Zero-shot)		Time-$\text{MoE}_{\text{Ultra}}$ (Zero-shot)		$\text{Moirai}_{\text{Large}}$ (Zero-shot)		TimesFM-2.5 (Zero-shot)		Chronos-2 (Zero-shot)		*Traditional Time-Series Models* iTransformer (Full-shot)		TimeMixer (Full-shot)		PatchTST (Full-shot)		DLinear (Full-shot)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.310	0.336	**0.273**	0.329†	0.281	0.341	0.380	0.361	0.307	0.326	0.301†	**0.312**	0.334	0.368	0.320	0.357	0.329	0.367	0.345	0.372
	192	0.356	0.364	0.312	0.357	**0.305**	0.358†	0.412	0.383	0.358	0.358†	0.352†	**0.343**	0.377	0.391	0.361	0.381	0.367	0.385	0.380	0.389
	336	0.388†	0.385	**0.343**	0.378	0.369	0.395	0.436	0.400	0.389	0.381†	0.388†	**0.367**	0.426	0.420	0.390	0.404	0.399	0.410	0.413	0.413
	720	0.439	0.415	**0.397**	0.413	0.469	0.472	0.462	0.420	0.444†	0.414†	0.451	**0.403**	0.491	0.459	0.454	0.441	0.454	0.439	0.474	0.453
	Avg	0.373†	0.375	**0.331**	0.369	0.356	0.391	0.422	0.391	0.375	0.370†	0.373†	**0.356**	0.407	0.409	0.381	0.395	0.387	0.400	0.403	0.406
ETTm2	96	0.171†	0.250†	0.172	0.255	0.198	0.288	0.211	0.274	0.170	0.239	**0.161**	**0.225**	0.180	0.264	0.175	0.258	0.175	0.259	0.193	0.292
	192	**0.225**	0.290†	0.227†	0.296	0.235	0.312	0.281	0.318	0.234	0.283	0.226	**0.271**	0.250	0.309	0.237	0.299	0.241	0.302	0.284	0.362
	336	**0.273**	0.324†	0.275	0.331	0.293	0.348	0.341	0.355	0.293	0.321	0.278†	**0.307**	0.311	0.348	0.298	0.340	0.305	0.343	0.369	0.427
	720	0.356†	0.378	**0.343**	0.378	0.427	0.428	0.485	0.428	0.389	0.379†	0.352	**0.359**	0.412	0.407	0.391	0.396	0.402	0.400	0.554	0.522
	Avg	0.256	0.311†	**0.254**	0.315	0.288	0.344	0.329	0.343	0.271†	0.305	**0.254**	**0.291**	0.288	0.332	0.275	0.323	0.280	0.326	0.350	0.400
ETTh1	96	0.364†	0.384	**0.346**	0.383	0.349	0.379	0.381	0.388	0.366	0.381†	0.369	**0.372**	0.386	0.405	0.375	0.400	0.414	0.419	0.386	0.400
	192	0.396†	0.404†	**0.386**	0.410	0.395	0.413	0.434	0.415	0.401	0.402	0.417	**0.400**	0.441	0.436	0.436	0.429	0.460	0.445	0.437	0.432
	336	**0.407**	0.413	0.410	0.426	0.447	0.453	0.485	0.445	0.415†	**0.412**	0.447	0.417†	0.487	0.458	0.484	0.458	0.501	0.466	0.481	0.459
	720	**0.404**	0.430	0.438	0.459	0.457	0.462	0.611	0.510	**0.404**	**0.423**	0.445†	0.431†	0.503	0.491	0.498	0.482	0.500	0.488	0.519	0.516
	Avg	**0.393**	0.407	0.395	0.420†	0.412	0.426	0.480	0.439	0.396†	**0.405**	0.420	**0.405**	0.454	0.447	0.448	0.442	0.468	0.454	0.455	0.451
ETTh2	96	0.284	0.334	**0.269**	0.330†	0.292	0.352	0.296	0.330†	0.275	0.318	0.280†	**0.316**	0.297	0.349	0.289	0.341	0.302	0.348	0.333	0.387
	192	0.341	0.373	**0.325**	0.373	0.347	0.379	0.361	0.371†	0.344†	0.365	0.346	**0.361**	0.380	0.400	0.372	0.392	0.388	0.400	0.477	0.476
	336	0.360	0.391†	**0.354**	0.400	0.406	0.419	0.390	0.390	0.374	0.390	0.370†	**0.386**	0.428	0.432	0.386	0.414	0.426	0.433	0.594	0.541
	720	**0.368**	0.410†	0.389	0.443	0.439	0.447	0.423	0.418	0.377†	0.402	0.370	**0.400**	0.427	0.445	0.412	0.434	0.431	0.446	0.831	0.657
	Avg	0.338	0.377†	**0.334**	0.387	0.371	0.399	0.367	0.377†	0.343	0.369	0.342†	**0.366**	0.383	0.406	0.364	0.395	0.386	0.406	0.558	0.515
Weather	96	0.145	0.189†	0.157†	0.208	0.157†	0.211	0.199	0.211	0.145	0.179	**0.139**	**0.172**	0.174	0.214	0.163	0.209	0.177	0.218	0.196	0.255
	192	0.193	0.236†	0.207	0.256	0.208	0.256	0.246	0.251	0.195†	0.228	**0.186**	**0.219**	0.221	0.254	0.208	0.250	0.225	0.259	0.237	0.296
	336	0.245	0.278†	0.259	0.295	0.255	0.290	0.274	0.291	0.260	0.275	**0.240**	**0.260**	0.278	0.296	0.251†	0.287	0.278	0.297	0.283	0.335
	720	0.329†	0.337†	0.327	0.342	0.405	0.397	0.337	0.340	0.359	0.336	**0.311**	**0.311**	0.358	0.349	0.339	0.341	0.354	0.348	0.345	0.381
	Avg	0.228	0.260†	0.238†	0.275	0.256	0.288	0.264	0.273	0.240	0.255	**0.219**	**0.241**	0.257	0.278	0.240	0.271	0.258	0.280	0.265	0.316
Avg Rank		2.48	3.45†	**2.23**	4.15	5.20	6.72	8.05	6.00	3.83	2.25	2.80†	**1.20**	8.10	8.07	5.67	5.80	7.72	7.88	8.93	9.47

G Additional Ablations

This section completes the ablation study of Section 4.3. Appendix G.1 gives the full dual- versus tri-tower comparison summarized in RQ2. Two further questions concern the trained model rather than the alignment design. RQ6 asks whether joint training preserves the language ability of the pretrained language tower, and RQ7 asks how strongly forecasts should draw on text across domains and history lengths.

G.1 RQ2 (Extended): Expert Separation

Unified vision models find that understanding and generation favor different representations and decouple them accordingly (Deng et al., 2025; Chen et al., 2025), which suggests giving perception and forecasting separate time-series towers. Figure 3E tests this hypothesis: the tri-tower variant initializes an independent perception expert and forecasting expert from the pretrained time-series models, while the dual-tower model uses the same time-series model to process both perception and forecasting streams. The two trajectories are nearly identical on both losses throughout training. Unlike vision understanding and generation, a shared temporal representation space is capable of processing information for perception and forecasting. We leave the study of using different time-series foundation models for different experts for future work.

G.2 RQ6: Language Ability Retention

Table 21: Five-shot MMLU accuracy (%) of TimeBraid-2.5B across training stages (Hendrycks et al., 2020). *Base* is the pretrained Qwen3-1.7B language tower evaluated standalone on text-only prompts; higher is better.

Stage	MMLU (%)
Base (Qwen3-1.7B language tower)	60.30
Align	40.88
Align, w/ unimodal replay	53.38
SFT	50.68
SFT, w/ unimodal replay during alignment	50.09

To examine whether joint training preserves the language ability of the pretrained language tower, we evaluate TimeBraid on MMLU at the end of each training stage. We also train a variant that mixes unimodal data into the alignment stage, replaying the Stage-2 unimodal corpora (Table 6) alongside the paired examples. In Table 21, we report the MMLU accuracy of these checkpoints.

From the table, the alignment stage costs the language tower 19.4 points, and mixing unimodal data into this stage recovers MMLU to 53.4. After supervised fine-tuning, the two variants end within 0.6 points of each other, since the fine-tuning mixture already contains an 11.69% unimodal share. We therefore keep the alignment stage purely paired. Overall, mixing unimodal data preserves the language ability of the unified model, although the shares we use do not preserve it fully and leave the released model 9.6 points below its backbone. We leave the search for a unimodal ratio that removes this gap to future work.

G.3 RQ7: TEXT-GUIDANCE SENSITIVITY

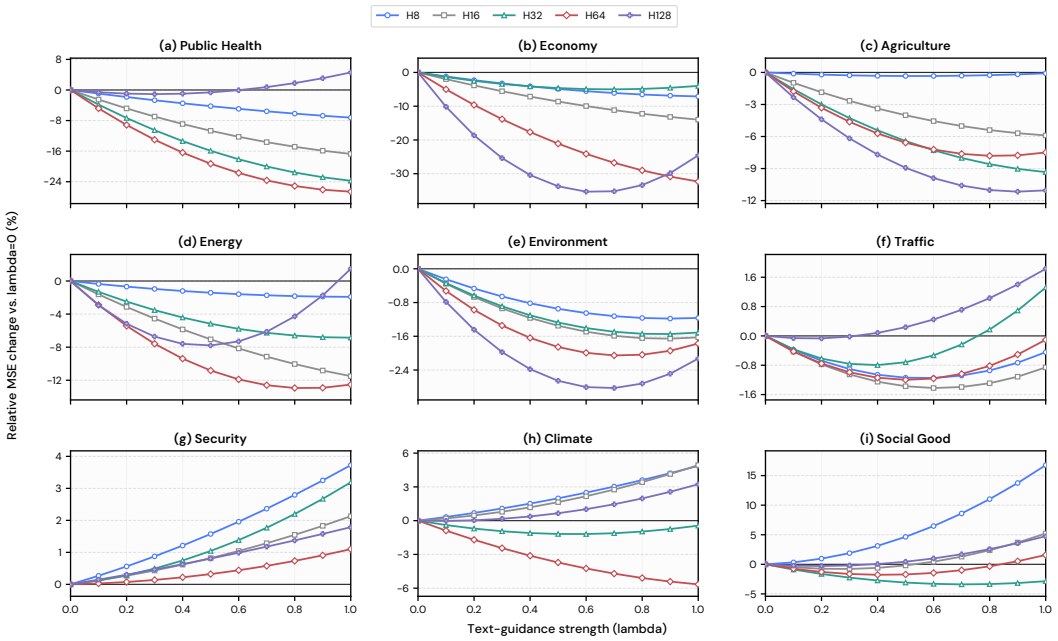


Figure 8: Effect of text guidance across domains and history lengths. Each panel sweeps the text-mixing weight λ on one Time-MMD domain, reporting the relative MSE change against the text-free forecast ($\lambda = 0$) at input histories 8–128; lower is better. At least one nonzero weight helps at every history length in Public Health, Economy, Agriculture, Energy, Environment, and Traffic; Security prefers $\lambda = 0$; Climate and Social Good change sign with history. We report this sweep as a sensitivity analysis because the released text is not point-in-time certified; it is separate from model selection, for which a single $\lambda = 0.3$ is selected on validation data and fixed across all domains.

Figure 8 shows that no single text weight works across all domains. Larger λ usually helps Economy, Public Health, Agriculture, and Energy; Traffic gains little at smaller weights, Security does not improve, and Climate and Social Good change with history length. The first group receives reports on trade, petroleum, broiler markets, or influenza that address the forecast target or a direct driver. The weaker pairs are less direct. Traffic pairs national monthly vehicle miles traveled with local road counts, air travel, and long-term plans. Security pairs monthly FEMA grant amounts, which contain rare event-driven spikes, with past disasters and grant rules that do not specify the next spike's timing or size. Environment pairs volatile daily New York AQI with broad or out-of-state reports; Climate and Social Good sometimes use reports from another period or region (Liu et al., 2024a). Text can therefore add little because it does not match the target, or because it does not specify an abrupt future change. In domains with weak or noisy context, a longer history can be more reliable than text-heavy mixing: Security is best without text, while Traffic and Public Health retain only small text weights at $H = 128$. In contrast, Economy, Agriculture, and Environment continue to benefit from larger text weights.

H Related Work

Unified Multimodal Models. Unified multimodal models differ chiefly in how a discrete language backbone hosts a continuous modality (An et al., 2026; Tong et al., 2026). One line quantizes every input into a shared token vocabulary and trains a single next-token predictor (Team, 2024; Zhan et al., 2024; Wang et al., 2024); a second mixes autoregressive text prediction with diffusion-style generation inside one transformer, increasingly over continuous latents (Zhou et al., 2025a; Xie et al., 2025; Deng et al., 2025; Xie et al., 2026); a third decouples the representations serving understanding from those serving generation (Chen et al., 2025). Omni-modal systems extend these recipes to speech, audio, and video (Xu et al., 2025a; Cui et al., 2026; Kim et al., 2026). Two lessons recur across this progression: continuous signals lose fidelity when forced through a discrete vocabulary (Fan et al., 2025), and understanding and generation favor different treatments of the same modality. Yet time series, for all their ubiquity, remain largely absent from these systems. TimeBraid carries both lessons to this missing modality: it aligns time series with language in continuous space, preserving numeric precision at both input and output, and gives perception and forecasting different input processing while sharing one time-series backbone.

Time-Series Foundation Models. Pretraining on large signal corpora yields foundation models that forecast zero-shot across domains (Garza et al., 2023; Rasul et al., 2023; Das et al., 2023; Ansari et al., 2024; Woo et al., 2024). The family has since diversified along axes familiar from language modeling: sparse experts (Shi et al., 2025), long-context and serial scaling (Liu et al., 2025a; 2026), generative decoding (Liu et al., 2025b), compact multi-task backbones (Goswami et al., 2024; Ekambaram et al., 2024; Gao et al., 2024; Xiao et al., 2025), and domain specialization (Cohen et al., 2026). Yet the interface stays numeric: domain knowledge and instructions have no way in and explanations no way out, even though textual context can materially improve forecasts (Williams et al., 2024; Liu et al., 2024a). TimeBraid builds on this line, initializing its perception and forecasting experts from pretrained time-series blocks to inherit their temporal competence, while adding the language interface and world knowledge these models lack.

Coupling Time Series with Language. Work coupling time series with language typically picks one pretrained host and moves the other modality into it. Understanding models host series in an LLM through an attached encoder or projector (Xie et al., 2024; Guan et al., 2026a; Wang et al., 2025b; Langer et al., 2025), even when that encoder is itself a pretrained time-series foundation model (Yu et al., 2025). Forecasting models host series in a language or vision–language model via digit serialization (Gruver et al., 2023), input reprogramming or adaptation (Jin et al., 2024; Chang et al., 2025; Liu et al., 2024b), or rendered images (Zhong et al., 2025); conversely, context-aided forecasters host text in a numeric model through an added text branch (Liu et al., 2024a). All of these cover only one direction of the interface, and the move itself is lossy: alignment between time-series, vision, and language representations has clear limits (Yashwante & Yu, 2026), and on purely numeric benchmarks the language host adds little (Tan et al., 2024). The few models covering both directions pay a different price: ChatTime hosts values in an LLM vocabulary as discrete tokens (Wang et al., 2025a), TimeOmni-VL hosts them in pixel space as rendered plots (Guan et al., 2026b), and the concurrent Chronicle learns a joint space from scratch with 324M parameters, forgoing pretrained competence on both sides (Quinlan et al., 2026). TimeBraid retains native processing in both pretrained language and time-series models and connects them through global residual attention in a shared space for bidirectional understanding and generation.

I LIMITATIONS

TimeBraid is our early attempt toward unified time-series and language models. It is trained without a continued pretraining stage. Public interleaved series–text corpora at pretraining scale are not yet mature, so our recipe moves directly from alignment to supervised fine-tuning. Arguably, large-scale continued pretraining would instill broader and richer world knowledge and improve the model across the board.

Perception carries bounded numeric precision. The model sometimes hallucinates the value at a specific point and misreads complex waveforms (long, high-frequency, or low signal-to-noise). Training coverage contributes, but the main reason is the patch tokenization inherited from the time-series backbone, which compresses many raw values into each embedding and does not fully preserve grounded local detail.

Forecast control weakens when text contradicts history. The forecasting expert inherits the transition dynamics of its pretrained backbone, which bias it toward historically consistent continuations. Direct control signals that demand behavior at odds with the history (an abrupt level shift, a sudden regime break) are followed loosely. Furthermore, large-scale high-quality real-world text-conditioned forecasting data and multi-modal multi-variate data are not readily available in public access. When the input text guidance and content are highly complex and noisy, TimeBraid cannot completely translate it to effective forecasting signals. However, we see that using a frontier language model for reasoning and instruction conversion helps mitigate these issues, and it implies the need for combination with agents for a better forecasting system.

J SUCCESS AND FAILURE CASES

The following selected qualitative examples illustrate forecasting and understanding capabilities. For context-conditioned forecasts, we show the supplied contextual text or a verbatim excerpt. Matched comparisons keep the checkpoint, numerical history, prediction horizon, and inference settings fixed, removing only the contextual information; the forecast instruction remains. Reported error reductions refer to the individual examples shown.



Figure 9: Context-conditioned forecasting with and without scenario text.

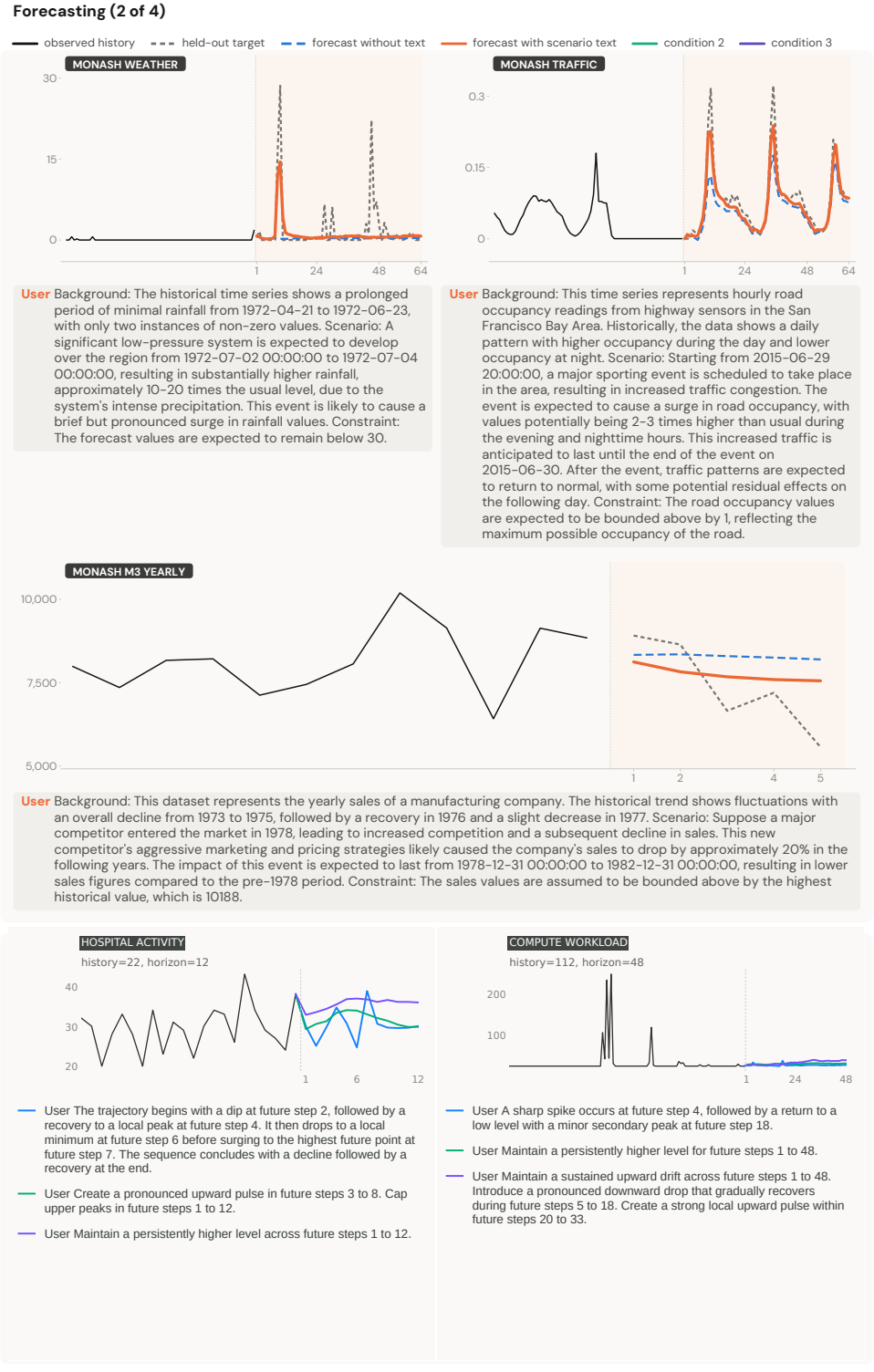


Figure 10: Context-conditioned and text-controlled forecasting on test examples. In the lower panels, solid blue, green, and purple denote conditions 1–3 over the same held-out history; dashed blue in the upper panels denotes forecasting without text. Only the differing condition text is displayed.

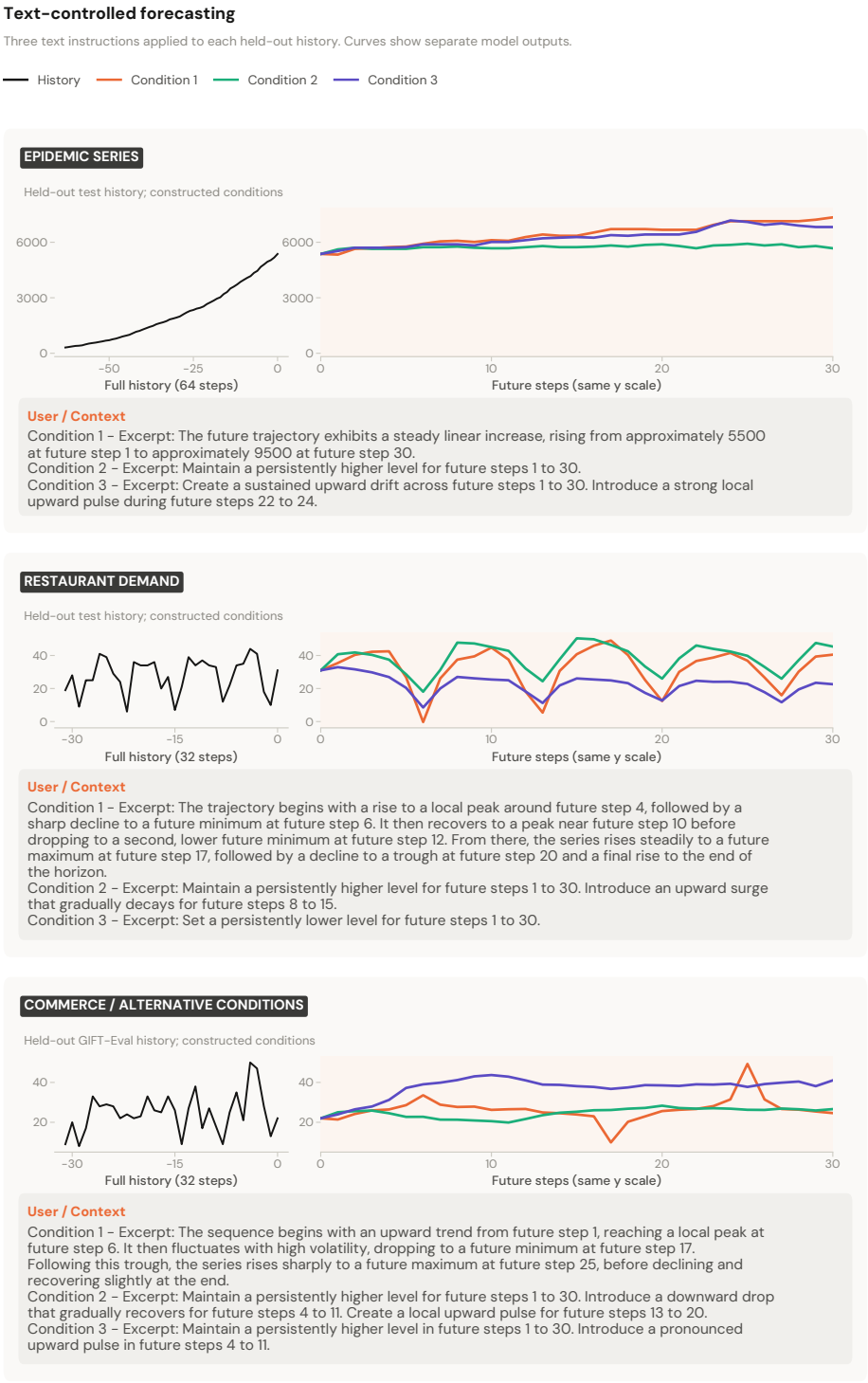


Figure 11: Text-controlled forecasting on held-out test histories. Each group uses three textual conditions with the same numerical input. Colored curves are recorded model outputs; the text below each panel quotes the differing condition. These are constructed scenarios, not observed alternative futures. Full histories and complete forecast horizons are shown on separate horizontal axes with the same vertical scale.

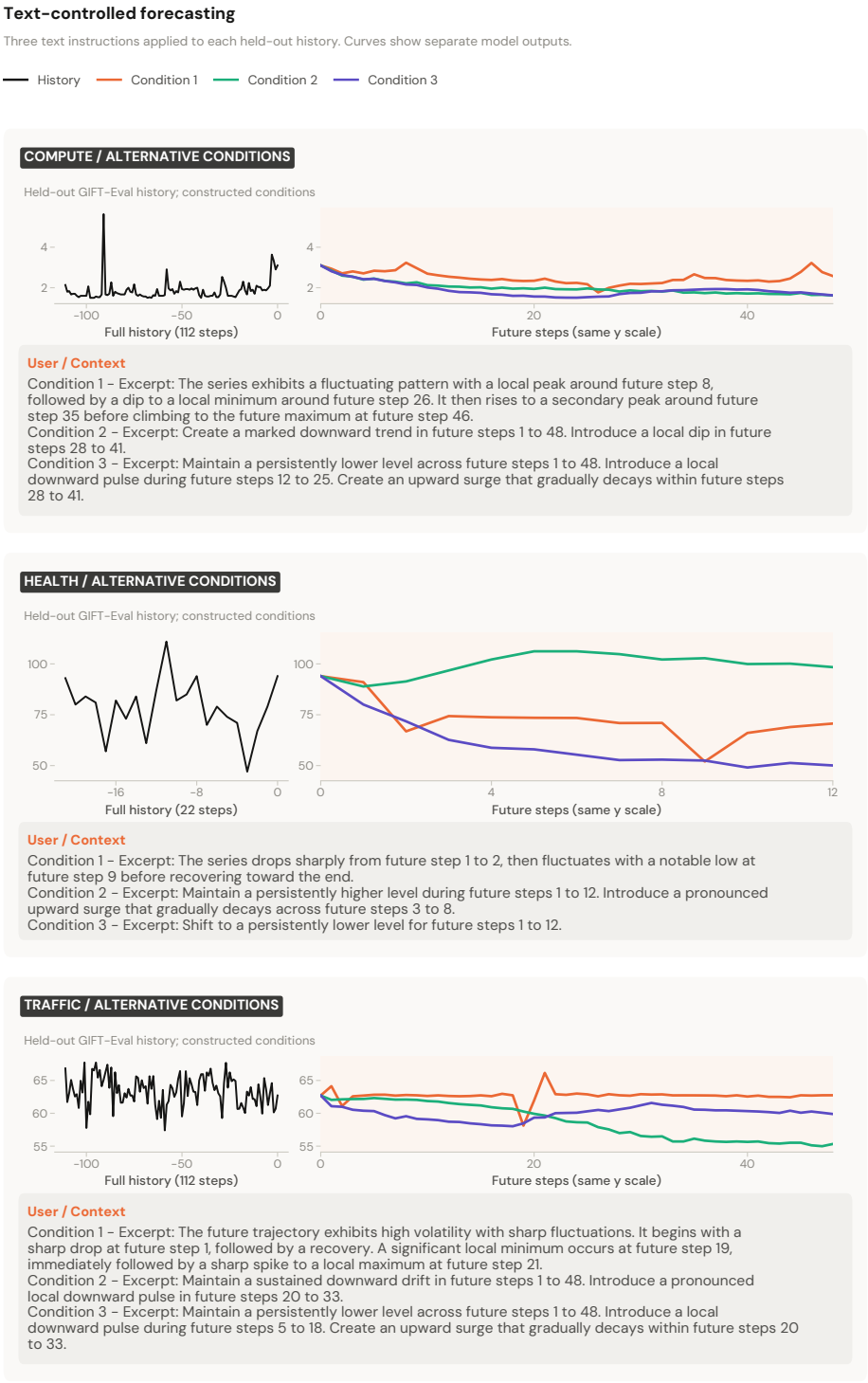


Figure 12: Additional text-controlled forecasts on held-out histories. All three predictions use the displayed textual conditions, quoted from the full inputs. The constructed scenarios vary trends, levels, and local changes; they illustrate the model's responses to distinct instructions. Both axes use the same vertical scale and display the complete input and output sequences.

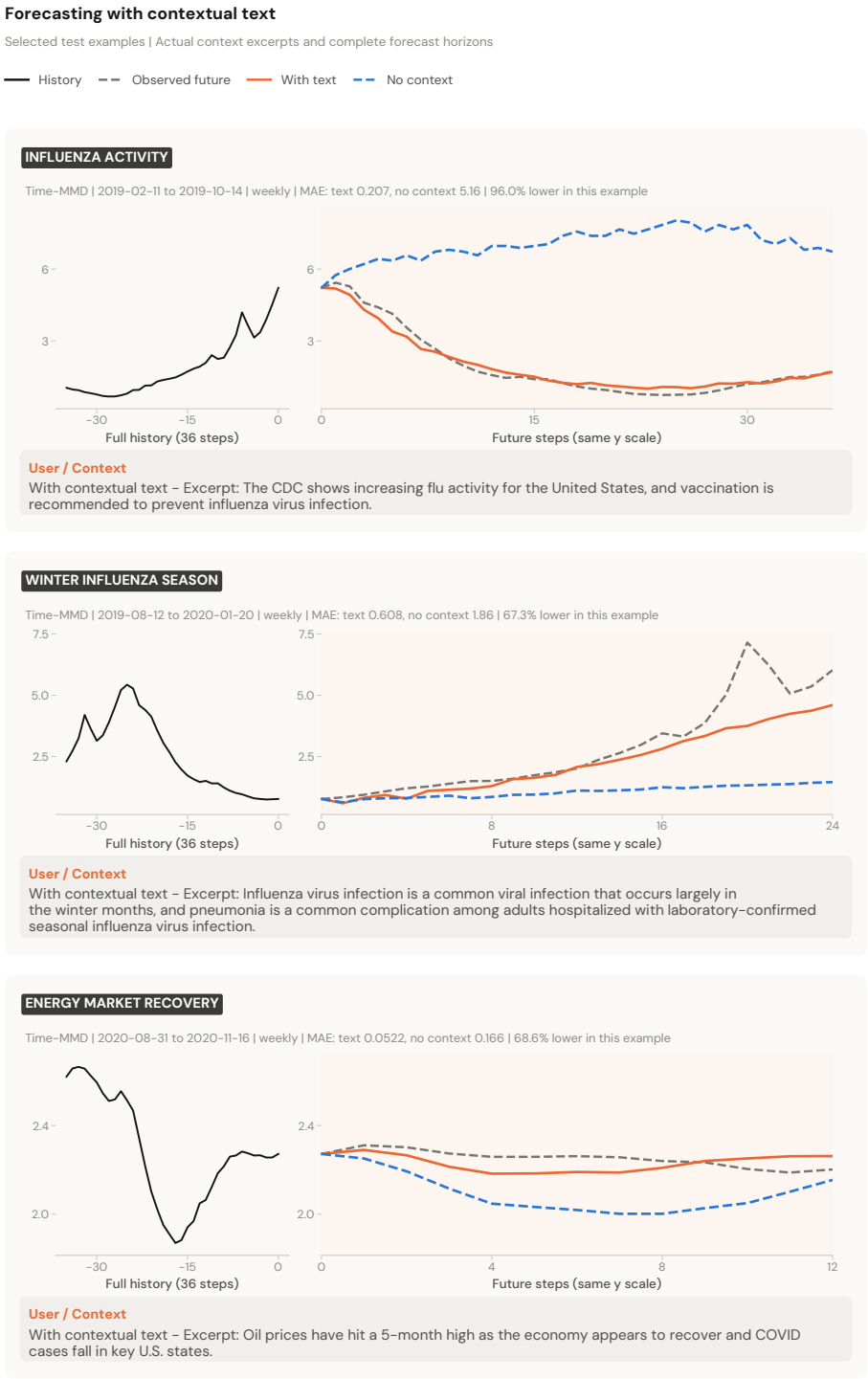


Figure 13: Selected Time-MMD test forecasts with matched contextual-text ablations. Orange uses the full dataset-provided textual context; dashed blue removes that context while retaining the same forecast instruction, numerical history, and horizon. Dashed gray is the observed future. Text below each panel is a verbatim excerpt of the full context used for prediction. MAE is measured over the complete future horizon in source units; reductions describe these selected examples.

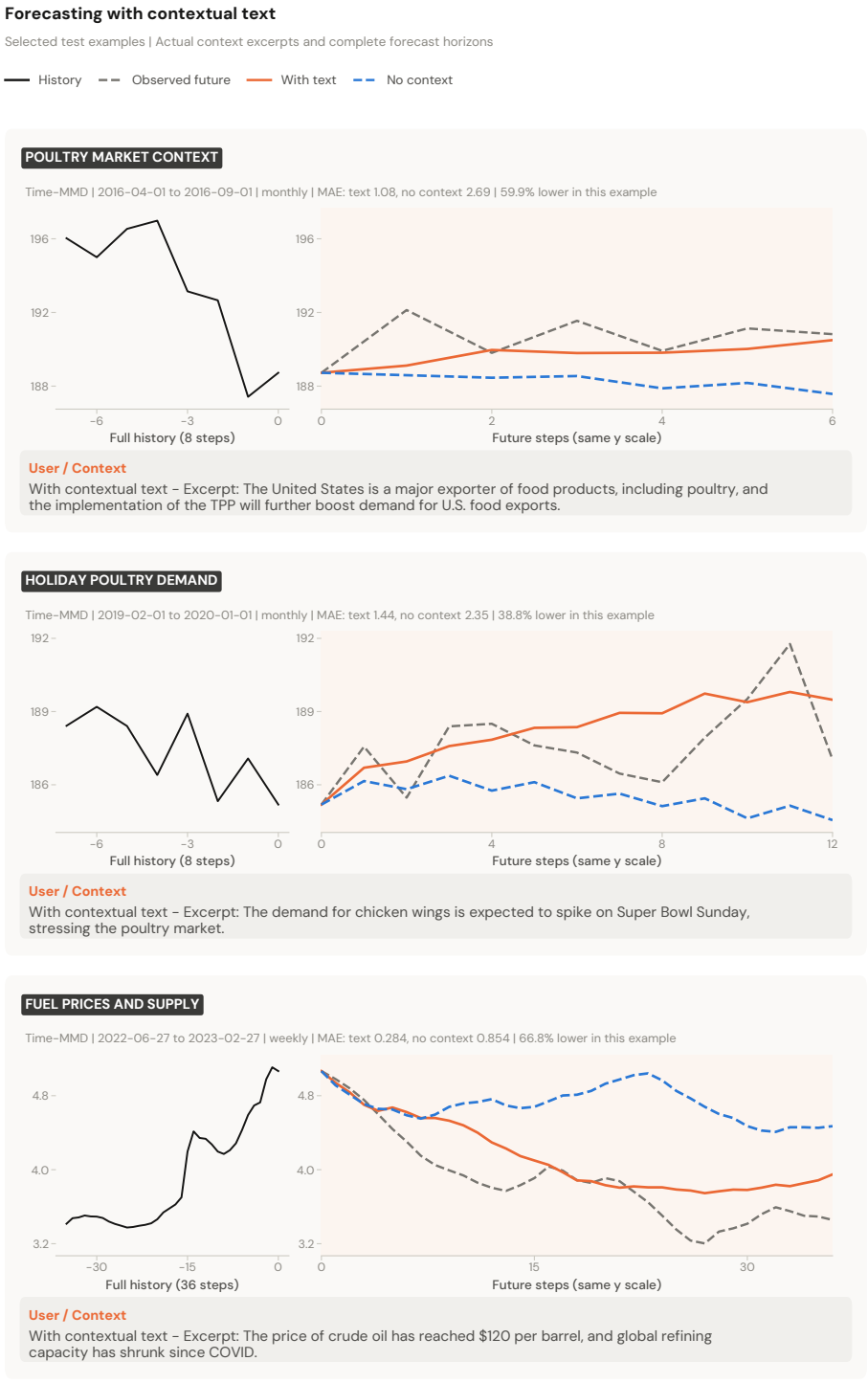


Figure 14: Additional matched Time-MMD test forecasts with market context. Orange forecasts use textual context; dashed blue forecasts remove only that context. Displayed excerpts are taken from the full inputs. The same checkpoint and inference settings are used for both arms. Complete histories and future horizons are shown, and the two axes in each example share their vertical scale.

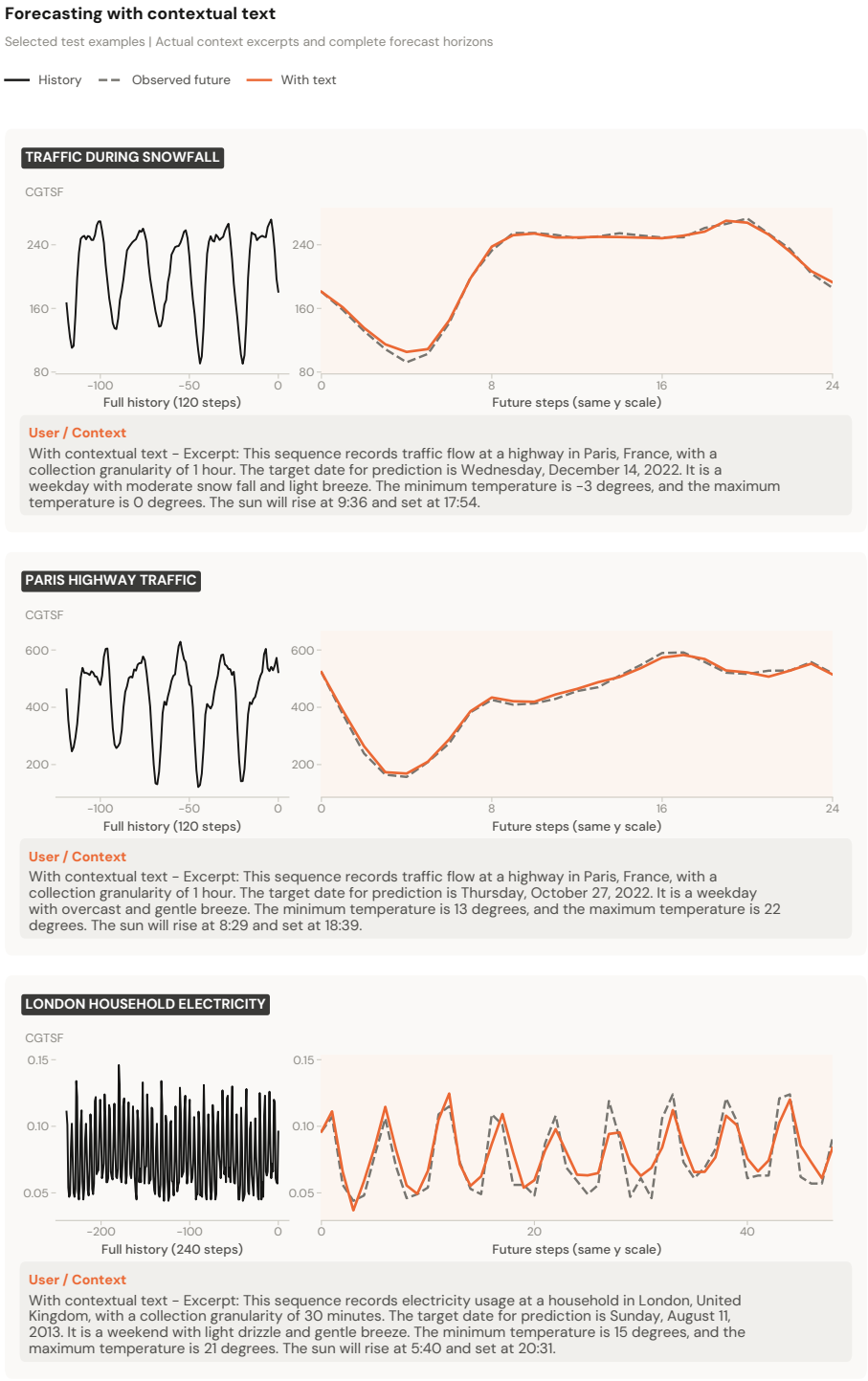


Figure 15: Selected CGTSF test forecasts using weather, calendar, and location text together with the numerical history. The contextual passages supplied to the model are displayed below the curves. Orange is the recorded text-conditioned prediction and dashed gray is the observed future. These examples show conditional forecast quality; a matched context-removal comparison is not included for this page.

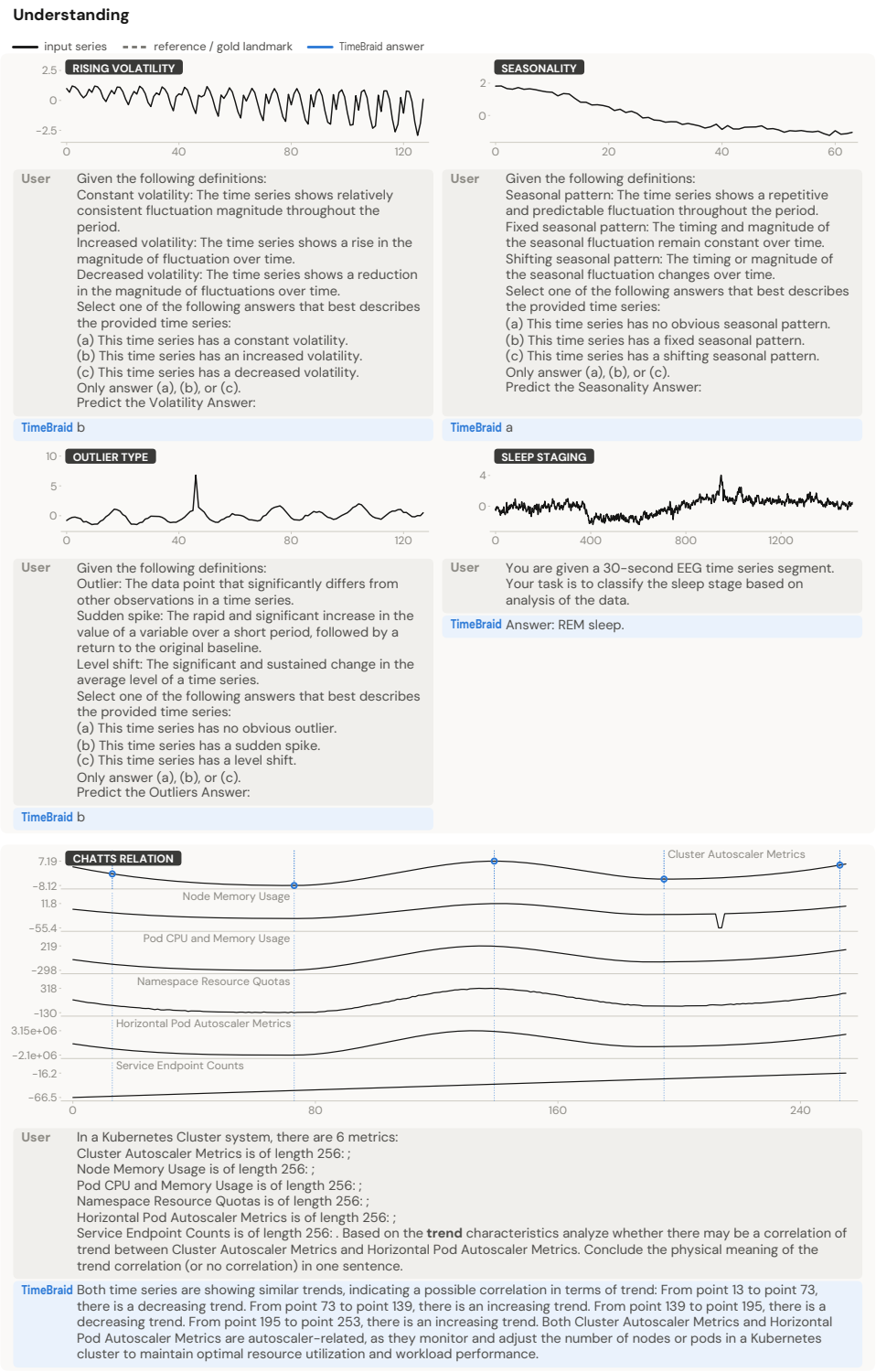


Figure 16: Series-property QA, sleep-stage classification, and multivariate relation analysis.

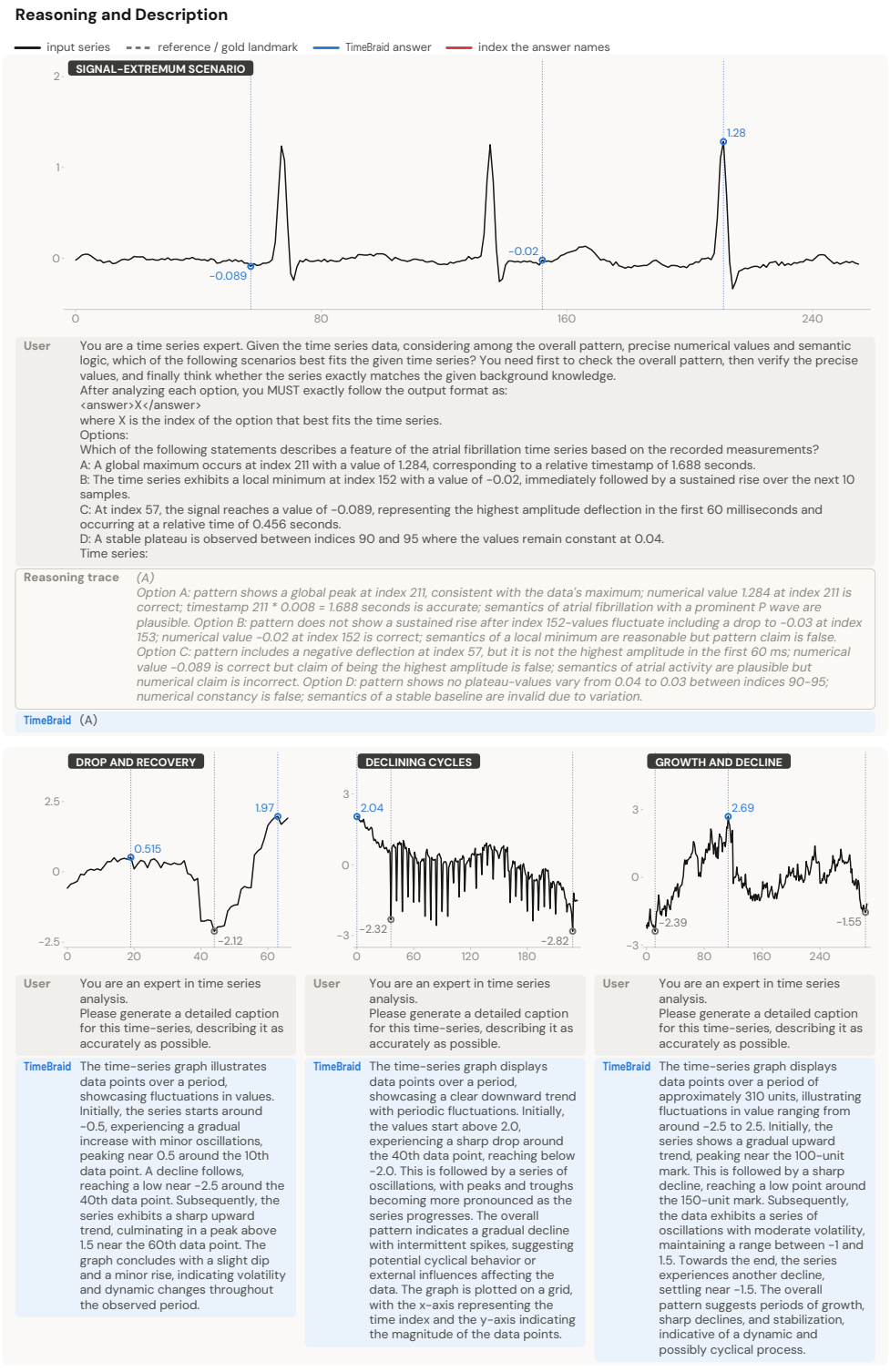


Figure 17: Scenario-based reasoning and time-series captioning.

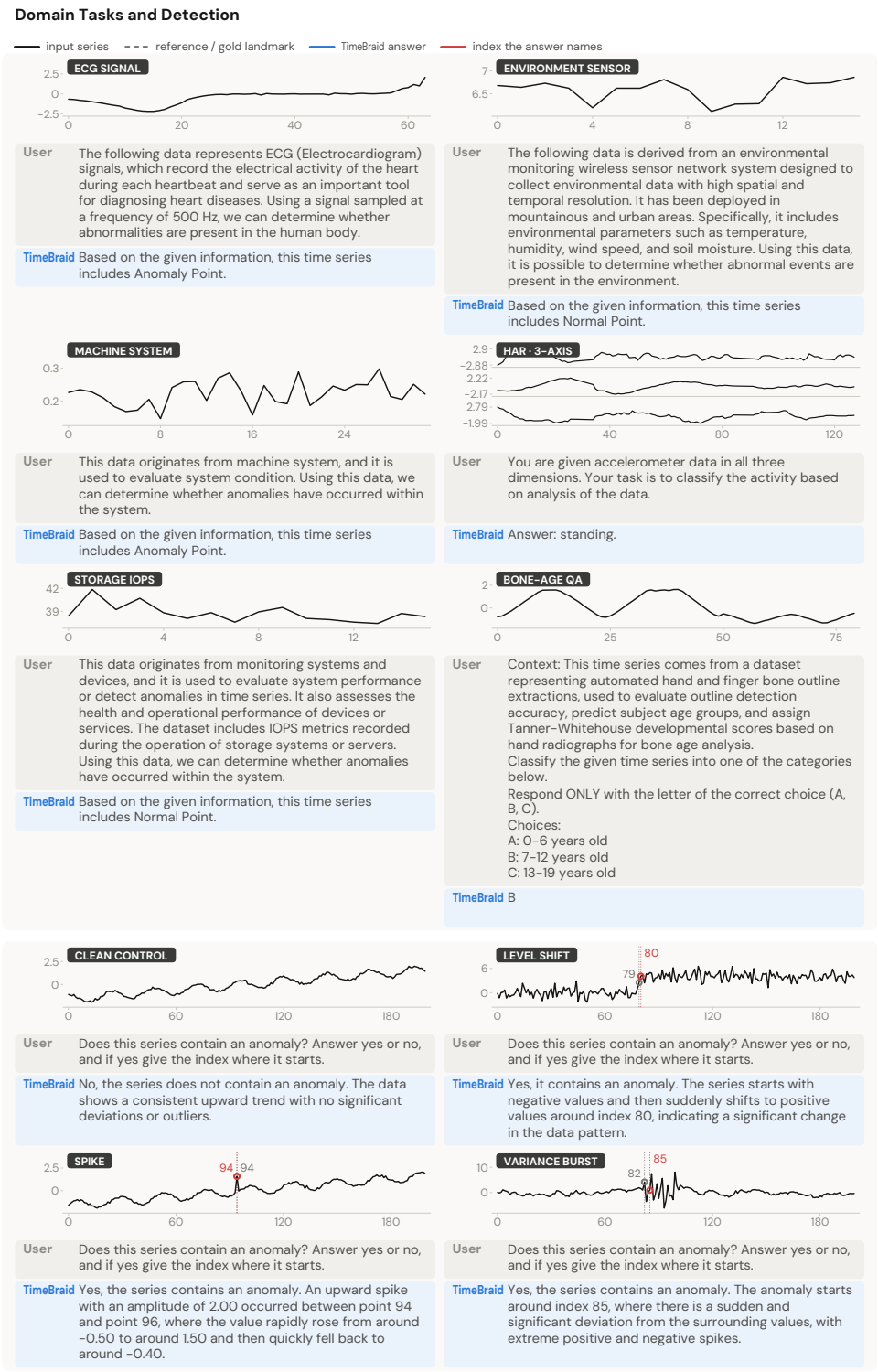


Figure 18: Sensor anomalies, human activity, bone age, and anomaly localization.

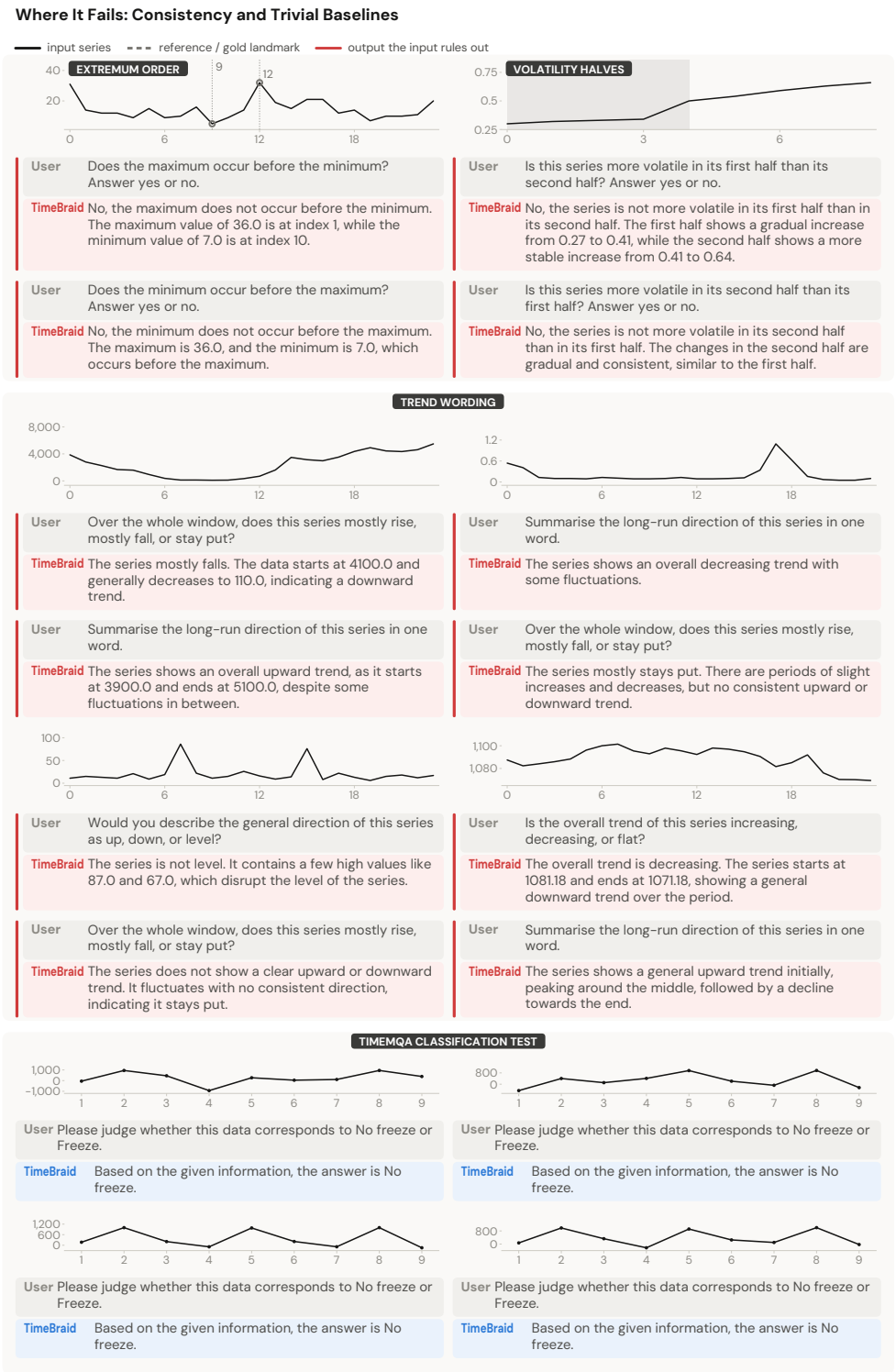


Figure 19: Question answering, paraphrase consistency, and time-series classification.

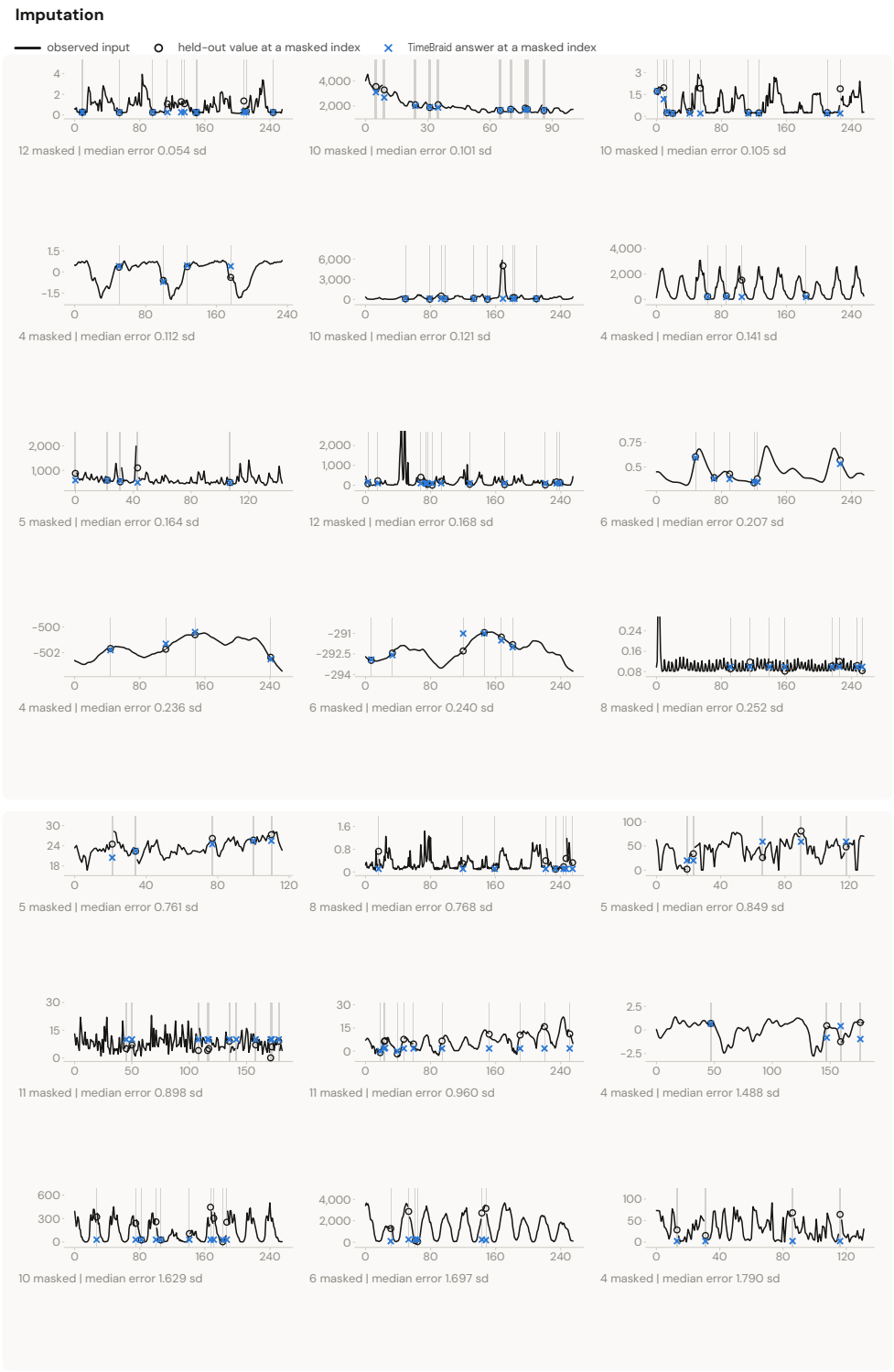


Figure 20: Imputation examples.