CHRONICLE: A MULTIMODAL FOUNDATION MODEL FOR JOINT LANGUAGE AND TIME SERIES UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world time series come with text: metadata, descriptions, news, reports. Yet time series foundation models process numerical sequences in isolation, and the multimodal text-and-time-series models that attempt to bridge the two all adapt a pretrained language model post hoc, inheriting representations shaped without ever seeing temporal data. These models are also evaluated almost exclusively against other multimodal baselines, not against the strongest unimodal foundation models in either domain, leaving open whether joint training is needed at all. We present **Chronicle**, a compact 324M-parameter decoder-only transformer trained from scratch on natural language and time series within a single unified architecture. Both modalities share the same transformer blocks, attention mechanism, and residual stream; the bulk of pretraining uses unimodal batches so cross-modal capability emerges purely from shared parameters, with a short alignment stage that interleaves the two. To our knowledge, Chronicle is the first model jointly pretrained on text and time series from scratch, and the first multimodal model evaluated against dedicated foundation models in both domains. It matches Gemma-3-270M-PT on 19 NLU tasks, sets a new bar for frozen-embedding time series classification on 24 UCR/UEA datasets, and produces multimodal forecasts on Time-MMD that beat every supervised fusion baseline, all from a single backbone.

1 INTRODUCTION

Time series foundation models (TSFMs) have transformed forecasting into an inference-only pipeline: a single pretrained model can be applied zero-shot across diverse domains (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Wen et al., 2026). Models such as Chronos-2 (Ansari et al., 2025) and PatchTST-FM (Wen et al., 2026) set a high bar on standardized benchmarks like GIFT-Eval (Aksu et al., 2024). Yet these models remain narrow specialists that process numerical sequences in isolation, with no mechanism to incorporate the textual context (metadata, domain knowledge, anomaly descriptions) that accompanies virtually every real-world time series.

A growing body of work has attempted to bridge this gap by connecting language models to time series, but these efforts share three systematic limitations. First, every existing approach starts from a pretrained language model and adapts it post hoc. LLMTIME (Gruver et al., 2023) and GPT4MTS (Jia et al., 2024) query frozen LLMs directly; Time-LLM (Jin et al., 2024) and GPT4TS (Zhou et al., 2023) add lightweight adapters; ChatTS (Xie et al., 2025), ChatTime (Wang et al., 2024), and MSE-ITT (Koval et al., 2025) fine-tune large backbones (Qwen2.5-14B, LLaMA-2, LLaMA-3-8B); and MoAT (Lee et al., 2024) and TaTs (Li et al., 2026) fuse separate encoders via learned heads (Table 1). Because these models were pretrained on text alone, their internal representations were shaped without any exposure to temporal data, and the time series modality must adapt to a representational space not designed for it. No prior work has trained a single model from scratch on both modalities, allowing text and time series to shape each other's representations from the beginning of training.

Second, these models are evaluated almost exclusively against other multimodal or task-specific baselines, not against state-of-the-art unimodal foundation models in either domain. Recent surveys (Liu et al., 2025) have noted this gap, and Zhang et al. (2025) found that the benefits of multimodality are "highly condition-dependent," underscoring the need for rigorous unimodal baselines. Third, many of

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.

these models, particularly ChatTS, ChatTime, Chat-TS (Quinlan et al., 2026), and MSE-ITT, target conversational reasoning about time series (question answering, summarization, explanation) and require large backbones (8 to 14B parameters) to support instruction following. This conflates two distinct goals: building general-purpose temporal *representations* versus building temporal *reasoning agents*. The question of whether a compact model can learn high-quality representations for both text and time series, without catastrophic interference, has not been addressed. Evidence from Tan et al. (2024), who showed that LLMs do not meaningfully improve forecasting as backbones, and Merrill et al. (2024), who found that LLMs struggle with text-encoded series, suggests that a different training paradigm may be required.

We present **Chronicle**, a 324M-parameter decoder-only transformer trained from scratch on natural language and time series within a single architecture. Both modalities share the same transformer blocks, attention mechanism, and residual stream; modality-specific components are limited to the input and output interfaces. The bulk of pretraining uses *unimodal* batches: each micro-batch contains either text tokens or time series patches, and the two modalities shape the backbone only through the shared parameters they both update. A short second stage at extended context length introduces a small fraction of interleaved text and time series sequences for explicit cross-modal alignment. At inference, text embeddings and patch embeddings can be freely interleaved within a single sequence, and cross-modal information flow arises naturally from causal self-attention without any architectural changes. Unlike prior work, our goal is not conversational reasoning but learning *general-purpose representations* that serve forecasting, classification, and embedding extraction, while retaining language understanding as a first-class capability. Simultaneous unimodal competence matters in practice: deployed systems routinely face missing or delayed modalities, and unimodal evaluation is the control that multimodal gains do not conceal a weak component. We evaluate Chronicle against dedicated foundation models in *each* modality on *their own benchmarks*: 19 NLU tasks against GPT-2 through LLaMA-3.2-1B; GIFT-Eval against the full public leaderboard; 24 UCR/UEA datasets against supervised models and frozen TSFM embeddings; and multimodal classification and forecasting on TimeCAP (Lee et al., 2025) and Time-MMD (Liu et al., 2024a), following the MM-TSFlib (Liu et al., 2024a) fusion evaluation protocol. Throughout, Chronicle refers to the final aligned checkpoint; the effect of the alignment stage is isolated in a dedicated ablation (Section 5.3). Our contributions are:

1. **Joint pretraining from scratch.** To our knowledge, Chronicle is the first model to learn text and time series end-to-end from random initialization within a single shared transformer backbone, rather than adapting a pretrained LLM post hoc.
2. **Cross-domain evaluation against unimodal foundation models.** We benchmark Chronicle against scale-matched LLMs on 19 NLU tasks *and* dedicated TSFMs on GIFT-Eval and UCR/UEA, addressing a longstanding gap in the multimodal time series literature.
3. **Strong frozen-backbone downstream performance.** Without per-dataset retraining, Chronicle sets a new bar for frozen-embedding TS classification, beats every supervised fusion baseline on Time-MMD, and matches Gemma-3-270M-PT on language understanding.

2 Related Work

We situate Chronicle within three lines of work and summarize the architectural landscape in Table 1; a comprehensive discussion appears in Appendix G.

Time series foundation models. TSFMs target zero-shot generalization across domains (Liang et al., 2024). Recent models span scalar tokenization (Chronos (Ansari et al., 2024), Chronos-2 (Ansari et al., 2025)) and patch-based encoding (PatchTST (Nie et al., 2023), TimesFM (Das et al., 2024), PatchTST-FM (Wen et al., 2026), Moirai (Woo et al., 2024), MOMENT (Goswami et al., 2024), UniTS (Gao et al., 2024)). We compare against these models on GIFT-Eval and UCR.

Multimodal text and time series models. Prior work falls into two categories (Table 1). *Reasoning-focused* models including ChatTS (Xie et al., 2025), Chat-TS (Quinlan et al., 2026), and MSE-ITT (Koval et al., 2025) fine-tune large backbones (8–14B) on synthetic QA data and target conversational benchmarks rather than standard forecasting or classification. ChatTime (Wang et al., 2024) instruction-tunes LLaMA-2 with discretized series but does not evaluate on the full GIFT-Eval suite or against scale-matched LLMs. *Forecasting-focused* models fuse text with time series via frozen LLMs (LLMTIME (Gruver et al., 2023), GPT4MTS (Jia et al., 2024)), adapters (Time-LLM (Jin et al.,

Table 1: **Positioning of Chronicle relative to prior multimodal text and time series models.** *Base model*: the pretrained backbone ("—" = trained from scratch). Chronicle is the only model that trains from scratch on both modalities and evaluates against unimodal foundation models in both domains.

Model	Params	Base model	TS input	Adaptation	Primary goal	Eval vs. TSFMs	Eval vs. LLMs	MM eval
LLMTIME (Gruver et al., 2023)	7–175B	GPT-3 / LLaMA	Digits	Frozen	Forecasting	Partial	✗	✗
GPT4MTS (Jia et al., 2024)	>7B	GPT-3.5 / GPT-4	Digits + prompt	Frozen	Forecasting	✗	✗	✓
Time-LLM (Jin et al., 2024)	7B	LLaMA-7B	Patch → text proto.	Adapter	Downstream Adaptation	Partial	✗	✗
GPT4TS (Zhou et al., 2023)	124M	GPT-2	Patch	Adapter (norm)	Downstream Adaptation	✗	✗	✗
MoAT (Lee et al., 2024)	Varies	Separate enc.	Patch (decomposed)	Late fusion	Downstream Adaptation	✗	✗	✓
TaTs (Li et al., 2026)	Varies	Separate enc.	Patch (+ text var.)	Late fusion	Downstream Adaptation	✗	✗	✓
ChatTS (Xie et al., 2025)	14B	Qwen2.5-14B	Patch (MLP enc.)	Full FT	Reasoning	✗	✗	✓
ChatTime (Wang et al., 2024)	7B	LLaMA-2	Scalar (discretized)	Full FT	Reasoning	Partial	✗	✓
Chat-TS (Quinlan et al., 2026)	8B	LLaMA-3-8B	Vocab expansion	Full FT	Reasoning	✗	✓	✓
MSE-ITT (Koval et al., 2025)	8B	LLaMA-3-8B	Patch (MoE experts)	Full FT + MoE	Reasoning	✗	✗	✓
Chronicle (ours)	**324M**	**—**	**Patch (interleaved)**	**Joint (from scratch)**	**Foundation model**	✓	✓	✓

2024), GPT4TS (Zhou et al., 2023)), or late-fusion heads (MoAT (Lee et al., 2024), TaTs (Li et al., 2026)). Time-MMD (Liu et al., 2024a) contributes both a benchmark and the MM-TSFlib fusion library, which we adopt as our multimodal baseline protocol. Unimodal-batch multimodal pretraining has precedent in vision–language models such as VLMo (Bao et al., 2022a) and VL-BEiT (Bao et al., 2022b), which combine image-only, text-only, and paired data; we do not claim unimodal batching is novel across multimodal research, but no prior work applies it to text and time series trained from scratch.

Across both categories, no prior work evaluates against dedicated TSFMs *and* dedicated LLMs on their respective benchmarks. Tan et al. (2024) showed that LLM pretraining does not transfer to forecasting, and Merrill et al. (2024) found that LLMs struggle with text-encoded series, motivating our modality-native joint training. Chronicle differs from all prior work in three respects: training from scratch on both modalities, using a compact 324M backbone focused on representation quality rather than dialogue, and evaluating against unimodal foundation models in both domains.

Small language models. GPT-2 (Radford et al., 2019) demonstrated that decoder-only transformers produce capable few-shot learners; subsequent compact models (Qwen2 (Yang et al., 2024), LLaMA-3.2 (Grattafiori et al., 2024), Gemma-3 (Gemma Team, 2025), LFM-2 (Amini et al., 2025)) push zero-shot understanding to strong levels at sub-1B scale. We compare against five such models on 19 NLU tasks to verify that joint training preserves language capability.

3 Methodology

Chronicle departs from prior multimodal text-and-time-series work: rather than adapting a pretrained LLM post hoc, we design an architecture in which both modalities shape a shared backbone from random initialization. The result is deliberately minimal—a decoder-only transformer (Vaswani et al., 2017; Radford et al., 2019) in which text tokens and time series patches occupy positions in a single sequence and flow through the same blocks, attention mechanism, and residual stream. Modality-specific components are confined to the interfaces: a text embedding table over a 131,072-entry BPE vocabulary and a patch projection on the input side; a tied language model head and a quantile head over $Q{=}21$ levels on the output side. Almost all parameters are therefore exercised by both modalities, and cross-modal information flow arises naturally from causal self-attention with no architectural additions.

3.1 Time Series Representation

Following PatchTST-FM (Wen et al., 2026) and Chronos-2 (Ansari et al., 2025), we first standardize each input using statistics computed *only* over visible (non-NaN, unmasked) values, then apply the inverse hyperbolic sine transform to suppress outliers while preserving sign:

$$x_{\text{norm}} = \operatorname{arcsinh}\Big(\frac{x - \mu_{\text{vis}}}{\sigma_{\text{vis}}}\Big), \tag{1}$$

where μ_{vis} and σ_{vis} are the visible-value mean and standard deviation. During autoregressive inference these statistics are computed once from the original context and cached, preventing distribution drift as

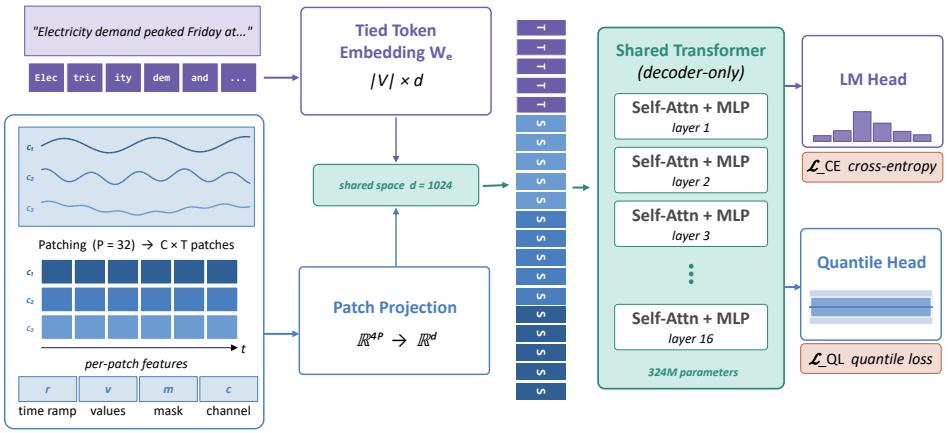

Figure 1: **The Chronicle architecture.** Text tokens and time series patches share a 16-layer decoder-only transformer, modality-specific components are limited to the input and output interfaces. Modality-specific output heads produce quantile forecasts ($\mathcal{L}_{\text{QL}}$) and next-token predictions ($\mathcal{L}_{\text{CE}}$).

model-generated predictions accumulate. The normalized series is then segmented into $T = \lceil L/P \rceil$ non-overlapping patches of length P=32; patching reduces effective sequence length by a factor of P and gives each token access to local temporal structure, matching the de facto input format of recent TSFMs (Nie et al., 2023; Ansari et al., 2025; Liu et al., 2026).

Each patch is represented by a $4P$-dimensional feature vector $\mathbf{f} = [\mathbf{r}; \mathbf{v}; \mathbf{m}; \mathbf{c}]$ obtained by concatenating four P-dimensional components: a *time ramp* $\mathbf{r}$ encoding the patch's normalized position within its channel (running from approximately -1 at the start of a channel to 0 at its end and resetting at channel boundaries for multivariate inputs); the *normalized values* $\mathbf{v}$ produced by Eq. 1; a binary *validity mask* $\mathbf{m} \in \{0, 1\}^P$ that distinguishes observed values from missing or masked positions; and a *channel ramp* $\mathbf{c}$ that encodes channel identity in multivariate inputs. For a sample with C channels, channel $j \in \{0, \ldots, C-1\}$ is assigned the scalar value $j/\max(C-1, 1)$, which is expanded across all P positions of each patch from that channel. Thus multivariate channel values are evenly spaced in $[0, 1]$, while univariate inputs use $\mathbf{c} = \mathbf{0}$. If channel identifiers are unavailable, the channel ramp defaults to zero. Appendix J.1 shows that channel-aware multivariate handling improves classification performance over mean-channel pooling on average on the multivariate UEA datasets. The feature vector is projected to the transformer embedding dimension d via a single bias-free linear layer followed by RMSNorm, and the resulting patch embedding is placed into the shared input space.

$$\mathbf{e}_{\text{ts}} = \text{RMSNorm}(\mathbf{W}_p \mathbf{f}) \in \mathbb{R}^d, \tag{2}$$

Text tokens are embedded via a learned table $\mathbf{W}_e \in \mathbb{R}^{|\mathcal{V}| \times d}$ that is tied with the language model output head (Press & Wolf, 2017), an important saving given the 131k-entry vocabulary. During pretraining, batches are either text-only or time-series-only (with the small interleaved fraction in stage 2 described in Section 4).

3.2 Output Heads and Training Objective

At text positions, transformer hidden states are projected to vocabulary logits via the tied embedding matrix, with logit soft-capping (Gemma Team, 2025) $\ell \leftarrow \alpha \tanh(\ell/\alpha)$, α=15, applied to prevent extreme pre-softmax values. The text loss is standard autoregressive cross-entropy $\mathcal{L}_{\text{CE}}$. At time series positions, RMSNorm-projected hidden states are mapped through a single bias-free linear layer to $P \times Q$ outputs, where Q=21 quantile levels are spaced uniformly over $\tau \in [0.05, 0.95]$. We minimize the masked quantile loss (Gneiting & Raftery, 2007):

$$\mathcal{L}_{\text{QL}} = \frac{\sum_{b,t,p,q} z_{b,t,p}\, \rho_{\tau_q}\big(y_{b,t,p} - \hat{q}_{b,t,p,q}\big)}{Q \sum_{b,t,p} z_{b,t,p}}, \quad \rho_\tau(u) = \max\big(\tau u,\ (\tau - 1)u\big), \tag{3}$$

where $z_{b,t,p} \in \{0, 1\}$ is a target-validity mask that is one only for finite, observed target values inside the prediction horizon and zero for padded or otherwise invalid positions. The mask is applied to each

per-position quantile loss term before normalization, so padded targets in partial forecast patches do not contribute to the objective.

At inference time, we denormalize predictions by inverting Eq. 1: $\hat{x} = \sinh(\hat{q}) \cdot \sigma_{\text{vis}} + \mu_{\text{vis}}$.

The overall training objective is the weighted sum $\mathcal{L} = w_{\text{text}}\mathcal{L}_{\text{CE}} + w_{\text{TS}}\mathcal{L}_{\text{QL}}$ with w_{text}=1.0 and w_{TS}=2.5; the asymmetric weighting reflects the substantially smaller scale of the per-element quantile loss relative to cross-entropy and balances gradient contributions from the two modalities.

4 Implementation

Chronicle is a 16-layer, 324M-parameter decoder-only transformer (d=1024, 8 GQA heads with 4 KV heads, RoPE (Su et al., 2024), SwiGLU (Shazeer, 2020), pre-norm RMSNorm (Zhang & Sennrich, 2019)), with patch length P=32 and a Q=21-quantile head over $\tau \in [0.05, 0.95]$. Pretraining runs on 2×H100 80GB GPUs in BF16. **Stage 1** trains at sequence length 2048 for 47,683 steps (∼3.1M tokens/batch), yielding ∼138B text and ∼12B TS patches; each micro-batch is text-only (p=0.92, from FineWeb-Edu (Penedo et al., 2024) and Dolmino-mix-1124 (OLMo Team & Allen Institute for AI, 2024)) or time-series-only (p=0.08, from GiftEvalPretrain plus KernelSynth augmentation; Appendix H). **Stage 2** extends context to 4096 and replaces 5% of TS tokens with interleaved alignment data from ChatTS (Xie et al., 2025) and Merrill et al. (2024), establishing cross-modal correspondences while preserving stage-1 capabilities. The text-heavy 92/8 mix is a compute constraint: matching text-only baselines trained on trillions of tokens requires devoting most of our budget to text (Section 5.1.1). The Stage-2 aligned checkpoint is the released Chronicle model used throughout the evaluation; the pre-alignment Stage-1 checkpoint is retained for the ablation in Section 5.3. Full details appear in Appendix C.

5 Evaluation

We organize the evaluation around four explicit questions. *Q1: Does joint pretraining preserve language understanding at matched scale? Q2: Does the time series stream produce competitive zero-shot probabilistic forecasts? Q3: Do the frozen temporal representations transfer to downstream time series classification? Q4: Does native joint pretraining outperform post-hoc fusion when both modalities are present?* Q1 and Q2 probe each training objective in isolation (Sections 5.1.1 and 5.1.2) and serve as the control that the multimodal gains in Q4 do not conceal a weak component; Q3 and Q4 assess downstream transfer (Section 5.2). Section 5.3 ablates the effect of the alignment stage, and Appendix D documents the pretraining–evaluation data separation and leakage controls.

Table 2: Language understanding (19 NLU tasks). Models ordered by parameter count. Subscripts indicate shot count.

Task	**GPT-2**	**Gemma-3**	**Chronicle**	**LFM-2**	**Qwen2**	**LLaMA-3.2**
Params	*124M*	*270M*	*324M*	*350M*	*500M*	*1.2B*
Tokens	∼10*B*	*6T*	∼153*B*	*10T*	*12T*	*9T*
HellaSwag_0	0.310	0.401	0.435	0.483	0.480	**0.629**
HellaSwag_{10}	0.308	0.397	0.429	0.473	0.482	**0.648**
ARC-E_{10}	0.417	0.583	0.651	**0.715**	0.595	0.678
ARC-C_{10}	0.224	0.289	0.325	**0.445**	0.311	0.376
COPA_0	0.630	0.670	0.660	0.690	0.670	**0.760**
CSQA_{10}	0.230	0.207	0.193	0.541	**0.582**	0.370
PiQA_{10}	0.624	0.676	0.694	0.698	0.700	**0.757**
LAMBADA_0	0.322	0.429	0.382	0.398	0.494	**0.627**
Winograd_0	0.575	0.652	0.663	0.608	0.696	**0.799**
WinoGrande_0	0.507	0.536	0.516	0.558	0.557	**0.609**
BoolQ_{10}	0.547	0.517	0.566	0.572	0.614	**0.657**
CoQA_0	0.136	0.223	0.211	0.303	0.324	**0.360**
SQuAD_{10}	0.058	0.250	0.290	0.318	**0.492**	0.479
Jeopardy_{10}	0.003	0.130	0.101	0.069	0.139	**0.344**
BB WikiQA_{10}	0.283	0.548	0.538	0.406	0.594	**0.643**
BB CSAlg_{10}	0.423	0.436	0.405	0.405	0.442	**0.458**
BB Ops_{10}	0.090	0.210	0.167	0.319	0.305	**0.405**
AGIE LSAT_3	0.209	**0.300**	0.239	0.243	0.252	0.239
BB LangID_{10}	0.258	0.254	0.248	0.282	**0.318**	0.253
Average	0.324	0.406	0.406	0.449	0.476	**0.531**

5.1 Pretraining Effectiveness

5.1.1 Q1: Language Understanding

We evaluate on 19 NLU tasks drawn from the DCLM evaluation suite (Li et al., 2024), spanning commonsense reasoning (HellaSwag (Zellers et al., 2019), COPA (Roemmele et al., 2011), PiQA (Bisk et al., 2020), CommonsenseQA (Talmor et al., 2019), WinoGrande (Sakaguchi et al., 2019)), reading comprehension (ARC-Easy, ARC-Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), CoQA (Reddy et al., 2019), SQuAD (Rajpurkar et al., 2016)), cloze and completion (LAMBADA (Paperno et al., 2016), Winograd (Levesque et al., 2012)), knowledge (Jeopardy,

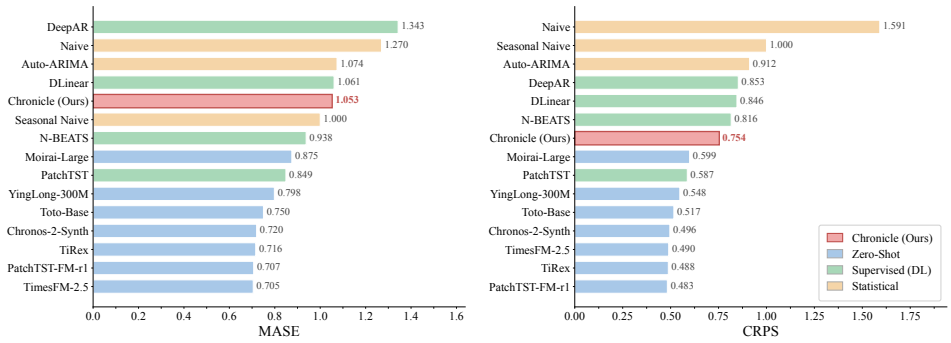

Figure 2: **GIFT-Eval leaderboard (97 tasks; lower is better).**

BB WikiQA (Srivastava et al., 2023)), and algorithmic reasoning (BB CS Algorithms, BB Operators, BB Language ID (Srivastava et al., 2023); AGI-Eval LSAT (Zhong et al., 2023)). All tasks use zero-shot or few-shot in-context learning with no fine-tuning. We compare against five text-only decoder-only language models: GPT-2 (Radford et al., 2019) (124M), Gemma-3-270M-PT (Gemma Team, 2025) (270M), LFM-2-350M (Amini et al., 2025) (350M), Qwen2-0.5B (Yang et al., 2024) (500M), and LLaMA-3.2-1B (Grattafiori et al., 2024) (1.2B).

Table 2 reports all 19 tasks. Chronicle achieves an average accuracy of 0.406, matching Gemma-3-270M-PT (0.406) at comparable scale, and sits between GPT-2 (0.324) and Qwen2-0.5B (0.476). On ARC-Easy, Chronicle (0.651) closely approaches LLaMA-3.2-1B (0.678), a model roughly $4\times$ larger trained exclusively on text, and outperforms both Gemma-3-270M-PT (0.583) and Qwen2-0.5B (0.595). The training budget context is an important consideration: Chronicle sees only $\sim$138B text tokens during the main pretraining stage ($\sim$153B including the alignment stage), roughly $43\times$ fewer than Gemma-3-270M (6T), $72\times$ fewer than LFM-2-350M (10T), and $87\times$ fewer than Qwen2-0.5B (12T). These results demonstrate that devoting $\sim 8\%$ of training compute to time series does not cause catastrophic interference in the shared transformer backbone, and validate the text-heavy token mix described in Section 4.

5.1.2 Q2: Zero-Shot Probabilistic Forecasting

GIFT-Eval (Aksu et al., 2024) comprises 97 zero-shot forecasting tasks drawn from 55 datasets across 7 domains at three horizon lengths (short, medium, long). All metrics are standardized by dividing by the Seasonal Naive baseline and aggregated via geometric mean; we report MASE (point forecast quality using the median quantile) and WQL (weighted quantile loss, equivalent to CRPS). We compare against published scores from the public GIFT-Eval leaderboard, representing the strongest dedicated TSFMs currently evaluated: the leading zero-shot models PatchTST-FM-r1 (Wen et al., 2026), TimesFM-2.5 (Das et al., 2024), TiRex (Auer et al., 2025), Toto-Base (Cohen et al., 2025), YingLong-300M (Wang et al., 2025), Chronos-2-Synth (Ansari et al., 2025), and Moirai-Large (Woo et al., 2024); supervised baselines PatchTST (Nie et al., 2023), N-BEATS (Oreshkin et al., 2020), DLinear (Zeng et al., 2023), and DeepAR (Salinas et al., 2020); and statistical baselines Seasonal Naive and Auto-ARIMA. We exclude models trained with potentially leaky data from test-set distributions. Chronicle forecasts autoregressively, generating one patch per step.

Figure 2 places Chronicle within the full leaderboard. Chronicle reaches 1.053 MASE and 0.754 CRPS, and despite allocating only $\sim 8\%$ of training compute to time series, it improves over several supervised and statistical baselines on CRPS, including Seasonal Naive, N-BEATS (0.816), DLinear (0.846), DeepAR (0.853) and Auto-ARIMA (0.912); the tradeoff introduced by the alignment stage is analyzed in Section 5.3. The remaining gap to dedicated TSFMs reflects two principled design choices: (i) only $\sim 8\%$ of training compute is allocated to time series versus 100% for dedicated models; and (ii) to align with our text setup we use causal next-patch prediction, while PatchTST-FM uses contiguous patch masking in an otherwise similar architecture, which their ablations show meaningfully reduces MASE.

5.2 Downstream Applications

We now evaluate whether Chronicle's learned representations transfer to three downstream tasks—time series classification (Q3), and multimodal classification and forecasting (Q4)—probing temporal discriminability, cross-modal fusion, and text-conditioned prediction without retraining the backbone.

Shared baselines. All downstream evaluations draw from a common pool of baselines. *Supervised DL models* (Informer (Zhou et al., 2021), TimesNet (Wu et al., 2023), Autoformer (Wu et al., 2021), iTransformer (Liu et al., 2024b), DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), and FEDformer (Zhou et al., 2022)) are trained independently per dataset for TS classification. *TS foundation models* (Chronos-2 (Ansari et al., 2025), Moirai-2 (Liu et al., 2026), and TimesFM (Das et al., 2024)) are evaluated with a learned linear probe on frozen embeddings for classification and via fusion heads or direct prediction for forecasting.

Multimodal fusion baselines follow the **MM-TSFlib** (Liu et al., 2024a) protocol, the standard fusion library introduced alongside the Time-MMD benchmark and subsequently adopted by multiple text-augmented time series studies. Under this protocol, each baseline pairs a *trainable* time series encoder (DLinear, PatchTST, or TimesNet) with a *frozen* pretrained text encoder (BERT (Devlin et al., 2019) or GPT-2 (Radford et al., 2019)) and a trainable two-layer MLP fusion head. The TS encoder is fine-tuned end-to-end on each dataset together with the head, so the temporal representation adapts to the task. We additionally report *FM Fusion* baselines that pair the same frozen text encoders with frozen TS foundation models (Chronos-2, Moirai-2, TimesFM) as encoders and train only the fusion head.

5.2.1 Q3: Time-Series Classification

We evaluate time-series classification on 24 datasets: 14 univariate datasets from the UCR Time Series Archive (Dau et al., 2019) and 10 multivariate datasets from the UEA archive, using official train/test splits. Supervised DL baselines are trained per-dataset with Adam (lr$=10^{-3}$, batch size 16). Foundation model baselines and Chronicle use a learned linear probe on frozen embeddings, directly testing whether pretrained temporal representations are linearly separable without backbone adaptation.

Table 3: Time series classification on 24 UCR/UEA datasets. Linear probes on frozen embeddings for TS foundation models and Chronicle; supervised DL baselines are trained per dataset. Full results, including the pre-alignment ablation checkpoint, in Table 13. Results are averaged over 5 different seeds.

Category	Model	Acc ↑	F1 ↑
Supervised DL	Informer	0.565 ± 0.213	0.483 ± 0.241
	TimesNet	0.645 ± 0.228	0.575 ± 0.267
	Autoformer	0.628 ± 0.221	0.566 ± 0.259
	DLinear	0.637 ± 0.228	0.606 ± 0.248
	iTransformer	0.628 ± 0.221	0.582 ± 0.249
	FEDformer	0.723 ± 0.229	0.666 ± 0.283
	PatchTST	0.668 ± 0.231	0.618 ± 0.267
TS Foundation Models (LP)	Chronos-2	0.376 ± 0.225	0.230 ± 0.167
	TimesFM	0.611 ± 0.238	0.563 ± 0.262
	Moirai-2	0.714 ± 0.238	0.692 ± 0.257
Chronicle	LP	$\mathbf{0.729 \pm 0.199}$	$\mathbf{0.700 \pm 0.220}$

Table 3 shows accuracy and F1 across all 24 datasets. Chronicle achieves the strongest overall results among frozen-backbone models, reaching 0.729 accuracy and 0.700 F1, compared to $0.714/0.692$ for Moirai-2, $0.611/0.563$ for TimesFM, and $0.376/0.230$ for Chronos-2. Chronicle also exceeds all supervised DL baselines trained per-dataset, including FEDformer (0.723 accuracy), from a single frozen backbone with no per-dataset retraining. The full results are given in Table 13.

5.2.2 Q4: Multimodal Classification

We evaluate multimodal classification on TimeCAP (Lee et al., 2025) across three domains: *Weather* (binary rain/no-rain), *Finance* (three-way market direction), and *Healthcare* (mean of two binary tasks: in-hospital mortality and disease test-positive prediction). All four underlying tasks are class-imbalanced (majority class 61–69%); we train every method with class-balanced cross-entropy, cap text inputs at 384 tokens, and report macro-F1 and AUC. Chronicle adds a 2-layer head on a single fully frozen backbone with joint text+TS input; full training settings appear in Appendix F.2 and we follow the training splits from Lee et al. (2025). Because Chronicle is causally autoregressive and TimeCAP series are short (often a single patch), the paired text caption can dominate the mean-pooled representation; we therefore optionally repeat the TS-token block r times within the input, rebalancing the modality ratio without altering the series or the backbone, and report the fair-comparison setting

Table 5: **Multimodal forecasting MAE on Time-MMD.** MM-TSFlib baselines finetune the TS encoder end-to-end with a frozen text encoder and a trained MLP head. FM Fusion baselines pair a frozen pretrained TS foundation model with a frozen text encoder and a trained fusion head. Chronicle reports zero-shot and finetuned head variants; average ranks are computed over the full pool of evaluated configurations. Abbreviations: Agri.=Agriculture, Clim.=Climate, Econ.=Economy, Enrg.=Energy, Env.=Environment, P.Hlth=Public Health, Sec.=Security, Soc.G=Social Good, Traf.=Traffic; DLin=DLinear, PTST=PatchTST, TNet=TimesNet, Chr2=Chronos-2, Moi2=Moirai-2, TFM=TimesFM.

Cat.	Model	Agri.	Clim.	Econ.	Enrg.	Env.	P.Hlth	Sec.	Soc.G	Traf.	NMAE	Avg Rank
MM-TSFlib (TS enc. trainable, text frozen)	BERT+DLin	**0.181**	0.901	**0.069**	0.409	0.447	0.839	2.067	0.499	0.241	0.648	7.44
	BERT+PTST	0.194	0.902	0.075	0.459	0.452	0.825	2.179	0.526	0.220	0.666	10.22
	BERT+TNet	0.186	0.913	0.073	0.437	0.438	0.818	1.707	0.535	0.226	0.621	8.56
	GPT2+DLin	0.184	0.890	0.073	0.424	0.447	0.790	1.871	0.523	0.225	0.626	6.78
	GPT2+PTST	0.195	0.884	0.076	0.449	0.465	0.794	1.772	0.562	**0.209**	0.627	9.00
	GPT2+TNet	**0.181**	0.924	0.076	0.464	0.450	0.833	1.780	0.591	0.211	0.640	9.78
FM Fusion (both enc. frozen, head trained)	BERT+Chr2	0.211	0.894	0.103	0.505	0.493	1.274	1.700	0.527	0.386	0.719	12.78
	BERT+Moi2	0.185	**0.873**	0.073	0.453	0.507	0.790	1.413	0.464	0.227	0.591	6.89
	BERT+TFM	0.190	0.881	0.071	0.403	0.491	0.765	1.686	0.441	0.228	0.601	6.11
	GPT2+Chr2	0.213	0.894	0.103	0.505	0.494	1.273	1.616	0.523	0.386	0.712	12.22
	GPT2+Moi2	0.189	0.878	0.073	0.448	0.514	0.785	1.394	0.453	0.223	0.588	6.44
	GPT2+TFM	0.187	0.878	0.079	0.396	0.491	0.777	1.575	0.471	0.229	0.595	6.89
Chronicle	ZS	0.222	0.890	0.100	0.648	1.270	1.490	1.082	0.519	0.386	0.835	11.89
	LP	**0.181**	0.875	0.070	**0.370**	**0.410**	**0.690**	**1.056**	**0.399**	0.245	**0.514**	**2.56**

r=1 and the tuned setting r=64 (full sweep in Appendix I). Table 4 reports results averaged over Weather, Finance, and Healthcare. Several baselines achieve inflated accuracy via majority-class collapse but perform poorly on these imbalance-aware metrics. Across macro-F1 and AUC, Chronicle is the strongest entry: LoRA (r=64) achieves the best average macro-F1 ($\mathbf{0.613}$) and AUC ($\mathbf{0.757}$), with the frozen LP (r=64) at 0.605/0.750; the best baselines reach 0.590 (TNet+BERT) and 0.754 (TNet+GPT2), trailing Chronicle on both metrics.

5.2.3 Q4: Multimodal Forecasting

We evaluate multimodal forecasting on the 9 Time-MMD (Liu et al., 2024a) domains (agriculture through traffic; textual fact reports; chronological 70/10/20 splits), reporting MAE averaged over all forecast horizons per domain (monthly: 6 to 12 steps; weekly: 12 to 48; daily: 48 to 336). For Chronicle, we report two variants: *ZS*, zero-shot forecasting, and *LP*, where a forecasting head is trained on top of the frozen backbone with joint text+TS input.

Table 4: Multimodal classification on TimeCAP. Scores are averaged over Weather, Finance, and Healthcare. Values are mean ± standard deviation over 3 seeds. Chronicle rows report both LP and LoRA scores with TS token repeat $r \in \{1, 64\}$ (Appendix I); the pre-alignment ablation appears in Section 5.3 and Table 12.

Category	Model	F1↑	AUC↑
MM-TSFlib	DLin+BERT	0.588 ± 0.016	0.739 ± 0.024
	DLin+GPT2	0.564 ± 0.026	0.724 ± 0.017
	PTST+BERT	0.578 ± 0.022	0.719 ± 0.022
	PTST+GPT2	0.539 ± 0.021	0.707 ± 0.034
	TNet+BERT	0.589 ± 0.021	0.750 ± 0.026
	TNet+GPT2	0.577 ± 0.019	0.754 ± 0.028
FM Fusion	BERT+Chr2	0.590 ± 0.021	0.726 ± 0.023
	BERT+Moi2	0.588 ± 0.004	0.751 ± 0.025
	BERT+TFM	0.498 ± 0.006	0.659 ± 0.023
	GPT2+Chr2	0.455 ± 0.043	0.673 ± 0.056
	GPT2+Moi2	0.542 ± 0.038	0.739 ± 0.015
	GPT2+TFM	0.480 ± 0.034	0.628 ± 0.018
Chronicle	LP (r=1)	0.594 ± 0.025	0.731 ± 0.029
	LP (r=64)	0.605 ± 0.011	0.750 ± 0.014
	LoRA (r=1)	0.595 ± 0.030	0.731 ± 0.032
	LoRA (r=64)	$\mathbf{0.613 \pm 0.011}$	$\mathbf{0.757 \pm 0.014}$

Table 5 presents per-domain MAE and normalized mean absolute error (NMAE) to account for differing data scales. Chronicle (LP) achieves the best overall NMAE ($\mathbf{0.514}$) and average rank ($\mathbf{2.56}$), outperforming the strongest MM-TSFlib baseline (BERT+TNet, NMAE 0.621, rank 8.56) and the strongest FM Fusion baseline (GPT2+Moi2, NMAE 0.588, rank 6.44). At the domain level, Chronicle (LP) leads on 5 of 9 domains (Energy, Environment, Public Health, Security, and Social Good) and is within 0.002 of the best method on Agriculture and Climate. Linear probing unlocks large gains over the zero-shot setting (e.g., Environment −0.860 MAE, Public Health −0.800, Energy −0.278). Because the head is common to all trained protocols, the difference lies in the frozen representation.

5.3 Ablation: The Effect of the Alignment Stage

Chronicle throughout the paper is the Stage 2 checkpoint produced by the alignment stage (Section 4); here we isolate what that stage adds and what it costs by comparing it against the pre-alignment Stage 1 checkpoint under a single fixed protocol (zero-shot where applicable, frozen LP/head, and LoRA). Figure 3 summarizes the comparison; the complete matrix appears in Table 7 (Appendix E). The pattern is consistent: Stage 1 is slightly stronger on every purely unimodal benchmark (NLU 0.411 vs. 0.406; GIFT-Eval $0.978/0.690$ vs. $1.053/0.754$ MASE/CRPS; UCR/UEA 0.736 vs. 0.729 accuracy), consistent with Stage 2 replacing a fraction of unimodal tokens with alignment data, while Stage 2 wins wherever both modalities are present (Time-MMD zero-shot NMAE $1.040 \rightarrow 0.835$ and frozen head $0.524 \rightarrow 0.514$; TimeCAP LoRA macro-F1 $0.584 \rightarrow 0.613$). Unimodal training gives the best isolated performance; a small interleaved alignment slice trades a little of it for consistent multimodal transfer.

TimeCAP AUC differs by only 0.006 with overlapping uncertainty, consistent with a calibration-level difference rather than a representational one. The strongest adaptation method is task-dependent, and frozen evaluation, Chronicle's baseline setting, is strong throughout. Every TimeCAP and Time-MMD input fits within the Stage-1 2048-token context, so both checkpoints receive identical sequences and the multimodal gains isolate the alignment data, not the context extension.

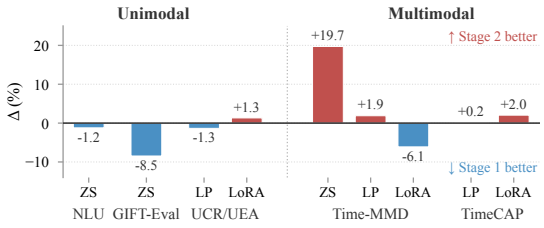

Figure 3: **Effect of the alignment stage.** Relative change from the Stage 1 checkpoint to the Stage 2 checkpoint. Alignment trades a small unimodal cost for multimodal gains. Full matrix in Table 7.

6 Limitations and Future Work

Several limitations of the current work suggest directions for future research. First, the forecasting gap to dedicated TSFMs reflects a compute trade-off: our text-heavy 92/8 mix was chosen to keep language understanding competitive with scale-matched text-only models, and closing this gap likely requires more compute or a curriculum strategy (e.g., TS-only pretraining followed by joint continued pretraining) rather than a different architecture. Second, our causal next-patch objective unifies the text and TS streams but compounds errors over long horizons; hybrid schemes that retain causal attention for text while applying bidirectional attention and contiguous patch masking for time series (Wen et al., 2026) could substantially improve long-horizon zero-shot forecasting within our architecture. Third, Stage 2 introduces explicit cross-modal supervision for only 5% of TS tokens, yet the consistent zero-shot to linear-probe gain on Time-MMD indicates that substantial cross-modal information remains latent in the frozen backbone; a larger interleaved alignment stage with millions of paired examples is the single most promising direction for improving it. Finally, we targeted a frozen representation backbone rather than a conversational agent, leaving open whether Chronicle can serve as a retrieval encoder for time series or, after instruction tuning, as the basis for temporal reasoning in the spirit of ChatTS (Xie et al., 2025) or MSE-ITT (Koval et al., 2025).

7 Conclusion

We presented Chronicle, a 324M-parameter decoder-only transformer trained from scratch on natural language and time series within a single shared backbone. Across five benchmarks—NLU, GIFT-Eval, UCR/UEA, Time-MMD, and TimeCAP—Chronicle matches scale-matched LLMs on language understanding, sets a new bar for frozen-embedding time series classification, and outperforms every supervised fusion baseline on multimodal forecasting, demonstrating that text and time series can share a transformer backbone without catastrophic interference. Our results challenge the prevailing assumption that multimodal time series models must adapt a pretrained LLM, and suggest that joint pretraining from scratch is a more direct path to general-purpose temporal representations. The remaining forecasting gap to dedicated TSFMs is attributable to compute allocation and autoregressive inference, both addressable with scaling and objective refinements; the architecture itself supports both modalities cleanly.

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, 2024. URL https://arxiv.org/abs/2410.10393.

Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, Ramin Hasani, Saniya Karwa, Yuri Khrustalev, Maxime Labonne, Mathias Lechner, Valentine Lechner, Simon Lee, Zetian Li, Noel Loo, Jacob Marks, Edoardo Mosca, Samuel J. Paech, Paul Pak, Rom N. Parnichkun, Alex Quach, Ryan Rogers, Daniela Rus, Nayan Saxena, Bettina Schlager, Tim Seyde, Jimmy T. H. Smith, Aditya Tadimeti, and Neehal Tumma. Lfm2 technical report, 2025. URL https://arxiv.org/abs/2511.23404.

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. In *International Conference on Machine Learning*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

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

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts, 2022a. URL https://arxiv.org/abs/2111.02358.

Hangbo Bao, Wenhui Wang, Li Dong, and Furu Wei. Vl-beit: Generative vision-language pretraining, 2022b. URL https://arxiv.org/abs/2206.01127.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. 2018.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, Jean Ogier du Terrail, Anna-Monica Toon, Kan Wang, Stephan Xie, Zongzhe Xu, Viktoriya Zhukova, David Asker, Ameet Talwalkar, and Othmane Abou-Amal. This time is different: An observability perspective on time series foundation models, 2025. URL https://arxiv.org/abs/2505.14766.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024. URL https://arxiv.org/abs/2310.10688.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

Shanghua Gao, Teddy Koker, Owen Queen, Tom Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: Building a unified time series model. In *Advances in Neural Information Processing Systems*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The LLaMA-3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2023.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. GPT4MTS: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 23343–23351, 2024.

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

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

Ross Koval, Nicholas Andrews, and Xifeng Yan. Multimodal language models with modality-specific experts for financial forecasting from interleaved sequences of text and time series. *arXiv preprint arXiv:2509.19628*, 2025.

Geon Lee, Wenchao Yu, Wei Cheng, and Haifeng Chen. MoAT: Multi-modal augmented time series forecasting. 2024. ICLR 2024 Workshop.

Geon Lee et al. TimeCAP: Learning to contextualize, augment, and predict time series events with large language model agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. arXiv:2502.11418.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pp. 552–561, 2012.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. DataComp-LM: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37, 2024.

Zihao Li, Xiao Lin, Zhining Liu, Jiaru Zou, Ziwei Wu, Lecheng Zheng, Dongqi Fu, Yada Zhu, Hendrik Hamann, Hanghang Tong, and Jingrui He. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative, 2026. URL `https://arxiv.org/abs/2502.08942`.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. *arXiv preprint arXiv:2403.14735*, 2024.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting, 2026. URL https://arxiv.org/abs/2511.11698.

Chenxi Liu, Shaowen Zhou, Qianxiong Xu, Hao Miao, Cheng Long, Ziyue Li, and Rui Zhao. Towards cross-modality modeling for time series analytics: A survey in the llm era, 2025. URL https://arxiv.org/abs/2505.02583.

Haoxin Liu et al. Time-MMD: A new multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37, 2024a.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2024b.

Mike A. Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*, 2024.

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

OLMo Team and Allen Institute for AI. Dolmino mix 1124: A dataset for the mid-training of OLMo 2. Hugging Face Datasets, 2024. URL https://huggingface.co/datasets/allenai/dolmino-mix-1124.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.10437.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2025/.

Paul Quinlan, Qingguo Li, and Xiaodan Zhu. Chat-ts: Enhancing multi-modal reasoning over time-series and natural language data, 2026. URL https://arxiv.org/abs/2503.10883.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.

Siva Reddy, Danqi Chen, and Christopher D Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium Series*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting world knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4149–4158, 2019.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

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, 2024. URL https://arxiv.org/abs/2412.11376.

Xue Wang, Tian Zhou, Jinyang Gao, Bolin Ding, and Jingren Zhou. Output scaling: YingLong-delayed chain of thought in a large pretrained time series forecasting model. *arXiv preprint arXiv:2506.11029*, 2025. URL https://arxiv.org/abs/2506.11029.

Yunshi Wen, Wesley M. Gifford, Chandra Reddy, Lam M. Nguyen, Jayant Kalagnanam, and Anak Agung Julius. Revisiting the generic transformer: Deconstructing a strong baseline for time series foundation models, 2026. URL https://arxiv.org/abs/2602.06909.

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

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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

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. *Proceedings of the VLDB Endowment*, 18(8):2385–2398, April 2025. ISSN 2150-8097. doi: 10.14778/3742728.3742735. URL http://dx.doi.org/10.14778/3742728.3742735.

An Yang, Baosong Yang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, 2023.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Xiyuan Zhang et al. When does multimodality lead to better time series forecasting? *arXiv preprint arXiv:2506.21611*, 2025.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting, 2025. URL https://arxiv.org/abs/2502.04395.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*, 2022.

Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all:power general time series analysis by pretrained lm, 2023. URL https://arxiv.org/abs/2302.11939.

A Model and Code

All model checkpoints and evaluation code will be publicly available upon publication at the links below. The Chronicle Stage 1 and Stage 2 checkpoints are hosted on Hugging Face at `[HUGGINGFACE_LINK]`. Evaluation code is available at `[GITHUB_LINK]`. We cannot provide direct links to them here to preserve anonymity throughout the review process. In addition, we provide comprehensive implementation and data preparation details below to ensure that these results are reproducible.

B Use of Large Language Models

In accordance with the ICLR 2027 policy on large language model (LLM) usage, we disclose the following. LLM-based assistants were used in the preparation of this work in the following capacities. *(i) Writing:* LLMs were used to edit, condense, and restructure the manuscript, to improve clarity and flow, to assist in preparing text for some sections from author-provided results and notes. *(ii) Code:* LLMs were used as coding assistants for portions of the training, evaluation, and plotting code. *(iii) Retrieval and discovery:* LLMs with web search were used to help identify related work and to verify bibliographic details of cited papers. The research questions, model architecture, training design, experiments, and analysis of results originate with the authors. All LLM-assisted text, code, and citations were reviewed and verified by the authors, who take full responsibility for the content of this paper.

C Full Implementation Details

Architectural summary. The model is a 16-layer decoder-only transformer with d=1024, 8 GQA query heads, 4 KV heads (head dim 128), and SwiGLU MLPs (Shazeer, 2020) of hidden dimension $\lceil 8d/3 \rceil$ rounded up to the nearest multiple of 256. Each block applies pre-norm RMSNorm before the attention and MLP submodules and uses standard residual connections; no additional residual-stream modifications are introduced. RoPE (Su et al., 2024) positional encodings use base frequency 5×10^5. QK normalization is applied within attention before the dot product. We use FlashAttention (Dao et al., 2022) for efficient causal self-attention; KV-caching support is included for autoregressive inference. Logit soft-capping via $15 \cdot \tanh(\ell/15)$ is applied to text outputs.

Input/output interfaces. Text embeddings are produced by a single learned table ($|\mathcal{V}|$=131,072, dimension d) and tied with the language model output head. Time series patch embeddings are produced by a single bias-free linear layer $\mathbf{W}_p \in \mathbb{R}^{d \times 4P}$ followed by RMSNorm, applied to the $4P$-dimensional patch features $[\mathbf{r}; \mathbf{v}; \mathbf{m}; \mathbf{c}]$ described in Section 3.1. The quantile head consists of an RMSNorm followed by a bias-free linear projection $\mathbb{R}^d \to \mathbb{R}^{P \cdot Q}$ with Q=21.

Tokenizer. The text tokenizer is a byte-level BPE vocabulary of 131,072 tokens trained from scratch with RustBPE on a 50B-character training corpus. The training mixture consists of 62.5% FineWeb-Edu (Penedo et al., 2024) and 37.5% Dolmino-mix-1124 (OLMo Team & Allen Institute for AI, 2024) (`dolmino_ratio`=0.375), matching the proportions used during model pretraining.

Optimizer groups. Parameters are partitioned into a Muon group and three AdamW groups. Muon (Jordan et al., 2024) (Newton–Schulz, 5 steps, momentum 0.95) is applied to all 2D weight matrices in the transformer blocks (attention Q, K, V, output projection, and SwiGLU w_1, w_2, w_3) at LR $= 0.02$. AdamW is used for: the token embedding table at LR $= 0.2$; the (untied) lm_head at LR $= 0.004$ when applicable; and the patch projection, quantile head, and all RMSNorm scales (including the post-embedding norm, the final norm, and the per-block pre-norms) at LR $= 0.002$. AdamW uses β_1=0.8, β_2=0.95, ϵ=10^{-10}, and weight decay 0. All AdamW learning rates are scaled by $\sqrt{768/d}$ to preserve update magnitudes across model dimensions.

Stage 1: schedule and batching. The learning rate follows a three-phase schedule: a 40-step linear warmup, a constant phase, and linear decay over the final 65% of training. We train for 47,683 steps at sequence length 2048 with a device micro-batch of 48 and 16 gradient accumulation steps, giving a global batch size of 3,145,728 tokens, yielding ∼150B total tokens (∼138B text, ∼12B time

series patches). At each step, the batch composition (text-only vs. time-series-only) is sampled and broadcast across all data-parallel ranks before any forward computation, so the gradient at every step is computed exclusively over a single modality.

Stage 2: extended context and multimodal alignment. Stage 2 reloads the stage-1 checkpoint, extends sequence length to 4096, and continues training. Within the time-series stream, 5% of tokens are drawn from interleaved alignment data: the alignment subset of ChatTS (Xie et al., 2025) (synthetic series paired with descriptive text labels) and the time series description corpus of Merrill et al. (2024) (natural-language descriptions of temporal patterns). The remaining 95% of TS tokens use the stage-1 unimodal corpus. For multimodal alignment batches, the loss combines $\mathcal{L}_{\mathrm{CE}}$ at text positions and $\mathcal{L}_{\mathrm{QL}}$ at TS positions, with the same global weighting (w_{text}=1.0, w_{TS}=2.5). The text/TS micro-batch ratio remains $0.92/0.08$, the optimizer state is preserved, and the learning rate continues its decay schedule from the end of stage 1.

Pretraining data. Text comes from a mixture of FineWeb-Edu (Penedo et al., 2024) and the Dolmino-mix-1124 sub-mixture (OLMo Team & Allen Institute for AI, 2024) (DCLM, FLAN, math, peS2o, Wikipedia, StackExchange) with the dolmino mix ratio set to 0.334 during pretraining. Time series data comes from GiftEvalPretrain ($\sim$900GB), augmented online with KernelSynth (2–5 kernels from 33 generators) and per-batch jitter, scaling, and mixup (Appendix H). We do *not* include explicit multimodal batches during stage 1. Stage 2 introduces the small alignment slice described above. For pretraining we use two H100 80GB GPUs. Total training for both stages takes roughly one week.

Weight initialization. Linear layers in the transformer use a fan-scaled normal init, $\mathcal{N}(0,\,\sigma)$ with $\sigma = \min(1, \sqrt{\mathrm{fan_out}/\mathrm{fan_in}})/\sqrt{\mathrm{fan_in}}$. The token embedding table is initialized with $\mathcal{N}(0, 0.02)$ when weight tying is enabled. RMSNorm scales are initialized to one. The output projections of attention and the SwiGLU w_3, as well as the (untied) lm_head when present, are zero-initialized to keep the residual stream near identity at initialization. The patch projection is initialized with the standard fan-scaled normal; the quantile head's linear projection is initialized to zero.

D Pretraining–Evaluation Data Separation

Table 6 consolidates the relationship between the pretraining corpora and every evaluation benchmark. All pretraining sources are public and enumerated in Section 4 and Appendix C; no proprietary pretraining data is used, and there is no overlap or leaking between the training and evaluation datasets.

Table 6: Relationship between pretraining data and each evaluation benchmark, with leakage and split controls.

Evaluation	Relationship to pretraining	Leakage/split control
GIFT-Eval	GiftEvalPretrain spans related domains	The benchmark's own curated non-leaking corpus; excludes GIFT-Eval test data
Time-MMD	Entire benchmark excluded	All splits, incl. downstream training split, excluded from pretraining; head trained only afterward, same as baselines
TimeCAP	Excluded	Downstream only; fixed stratified 70/10/20 split per the TimeCAP authors' implementation
UCR/UEA	No classification labels in pretraining	Official train/test splits; only the probe sees the training split, no training data from this benchmark in pretraining
NLU	Standard public suite	Same protocol as the text-model comparisons with no overlap between pretraining text corpus and evaluation corpus

E Stage 1 vs. Stage 2: Full Ablation Matrix

Table 7 reports the complete Stage 1 vs. Stage 2 matrix summarized in Figure 3 (Section 5.3): zero-shot where applicable, frozen LP/head, and LoRA, for both stages regardless of outcome. LoRA and seed settings are documented in Appendix F.

Table 7: Stage 1 (pre-alignment) vs. Stage 2 (aligned) under a fixed protocol. TimeCAP values are F1/AUC at r=64; LoRA uncertainty is over 3 seeds; UCR/UEA LP uncertainty is computed across seed-level benchmark aggregates (5 seeds).

Benchmark (metric)	**Setting**	**Stage 1**	**Stage 2**
NLU, 19 tasks (avg acc ↑)	zero/few-shot	0.411	0.406
GIFT-Eval (MASE ↓ / CRPS ↓)	zero-shot	0.978 / 0.690	1.053 / 0.754
UCR/UEA, 24 datasets (Acc ↑)	frozen LP	0.736 ± 0.001	0.729 ± 0.057
	LoRA	0.699 ± 0.013	0.708 ± 0.021
TimeCAP, 3 domains (F1/AUC ↑)	frozen LP	$0.608 \pm .010$ / $0.745 \pm .014$	$0.605 \pm .011$ / $0.750 \pm .014$
	LoRA	$0.584 \pm .024$ / $0.763 \pm .011$	$0.613 \pm .011$ / $0.757 \pm .014$
Time-MMD, 9 domains (NMAE ↓)	zero-shot	1.040	0.835
	frozen head	0.524	0.514
	LoRA	0.548 ± 0.003	0.581 ± 0.015

F Downstream Evaluation Setup

This appendix consolidates the downstream evaluation protocol for TS classification (UCR), multimodal classification (TimeCAP), and multimodal forecasting (Time-MMD). All settings here apply to every method in the corresponding tables, baselines and Chronicle alike, unless explicitly noted otherwise.

F.1 Common Settings

Table 8: Settings shared across all downstream evaluations.

Setting	**Value**
Seed	1337
TS normalization	instance z-score
Chronicle patch length	32
TimeCAP split	stratified $70/10/20$ train/val/test
Time-MMD forecasting split	chronological $70/10/20$, no shuffle
UCR TS-CLS split	aeon default train/test split
TimeCAP MM-CLS max text length	384 tokens
TimeCAP MM-CLS trainable loss	class-balanced cross-entropy

The TimeCAP class-balanced cross-entropy weights each class by the inverse of its training-set frequency, normalized to sum to the number of classes. This is applied uniformly to every trainable head in the MM classification table (MM-TSFlib supervised fusion, FM Fusion with frozen encoders, and the Chronicle head) so that no method gains an artificial advantage from majority-class collapse.

F.2 TimeCAP Multimodal Classification

TimeCAP (Lee et al., 2025) pairs short multivariate time series with GPT-4-generated text summaries. We evaluate on three reporting domains. Weather labels are collapsed from the original city-specific labels into binary rain / no-rain. Healthcare is reported as the mean of two underlying binary tasks (in-hospital mortality and disease test-positive prediction); each is evaluated separately under the same protocol and the per-domain numbers in Table 12 are their average. Dataset statistics for the four underlying classification tasks are summarized in Table 9.

Baselines. MM-TSFlib fusion baselines pair a trainable time series encoder (DLinear, PatchTST, or TimesNet) with a frozen pretrained text encoder (BERT or GPT-2) and a trainable two-layer MLP fusion head. The TS encoder and head are trained end-to-end for 100 epochs at learning rate 10^{-3} with batch size 8. FM Fusion baselines replace the trainable TS encoder with a frozen pretrained TS foundation model (Chronos-2, Moirai-2, or TimesFM) and train only the fusion head under the

Table 9: TimeCAP MM classification dataset statistics. "TS shape" is (steps, channels) for multivariate series and (steps,) for univariate. Caption length is reported in whitespace- delimited word counts.

Task	Samples	Train / Val / Test	Classes (counts)	TS shape	Caption words (mean / max)
Weather	5,652	3955 / 566 / 1131	no rain (4149), rain (1503)	(24, 5)	132.6 / 196
Finance	1,238	866 / 124 / 248	class 1 (857), 2 (211), 0 (170)	(9,)	160.6 / 228
Healthcare mortality	375	262 / 38 / 75	False (260), True (115)	(4,)	153.8 / 212
Healthcare positive	427	298 / 43 / 86	False (294), True (133)	(6,)	154.2 / 199

same schedule. All baselines use the corrected TimeCAP labels, class-balanced cross-entropy, and a maximum text length of 384 tokens.

Chronicle. For the linear-probe setting, we feed the joint text–time-series input to a single frozen Chronicle backbone and train only a two-layer MLP classification head. The head uses the same optimizer settings as the fusion baselines: 100 epochs, learning rate 10^{-3}, batch size 8, dropout 0.1, class-balanced cross-entropy, and `mean_full` pooling over the backbone outputs. Because Chronicle is causally autoregressive, short TimeCAP series can be underrepresented relative to the accompanying text. We therefore repeat the TS-token block r times within the input, without changing the underlying time series or updating the backbone, and sweep $r \in \{1, 2, 4, 8, 16, 32, 48, 64, 96, 128\}$ in Appendix I. Macro-F1 peaks at r=32, while average AUC peaks at r=64 (0.792 versus 0.788 at r=32); we therefore report both the fair-comparison setting r=1 and the tuned setting r=64 in the main results.

For LoRA experiments, the pretrained backbone weights remain fixed and we train only the LoRA adapters together with the classification head. We report these rows separately from the linear-probe results to distinguish frozen-backbone evaluation from parameter-efficient adaptation.

F.3 Time-MMD Multimodal Forecasting

Time-MMD (Liu et al., 2024a) pairs each of nine domain-specific multivariate time series with aligned textual fact reports. Each domain is a single chronological sequence; we use the MM-TSFlib chronological $70/10/20$ split with frequency-specific context and horizon settings (Table 10).

Table 10: Time-MMD MM forecasting dataset statistics. "Test windows by horizon" lists the number of evaluation windows produced by each horizon length in the same order as the "Horizons" column.

Domain	Freq.	Rows	Channels	Train / Val / Test rows	Context	Horizons	Test windows by horizon
Agriculture	monthly	496	3	347 / 50 / 99	8	6, 8, 10, 12	94, 92, 90, 88
Climate	monthly	496	2	347 / 50 / 99	8	6, 8, 10, 12	94, 92, 90, 88
Economy	monthly	423	3	296 / 43 / 84	8	6, 8, 10, 12	79, 77, 75, 73
Energy	weekly	1,479	9	1035 / 149 / 295	36	12, 24, 36, 48	284, 272, 260, 248
Environment	daily	15,979	2	11185 / 1599 / 3195	96	48, 96, 192, 336	3148, 3100, 3004, 2860
Public Health	weekly	1,389	8	972 / 140 / 277	36	12, 24, 36, 48	266, 254, 242, 230
Security	monthly	297	1	207 / 31 / 59	8	6, 8, 10, 12	54, 52, 50, 48
Social Good	monthly	900	1	630 / 90 / 180	8	6, 8, 10, 12	175, 173, 171, 169
Traffic	monthly	531	1	371 / 54 / 106	8	6, 8, 10, 12	101, 99, 97, 95

Baselines. MM-TSFlib forecasting baselines train a trainable TS encoder (DLinear, PatchTST, or TimesNet) jointly with a frozen text encoder (BERT or GPT-2) and a trainable forecasting head. FM Fusion baselines substitute a frozen pretrained TS foundation model (Chronos-2, Moirai-2, or TimesFM) for the trainable encoder and train only the fusion head. All baselines minimize MSE on the Time-MMD training split and report MAE on the held-out test windows; per-domain numbers in Table 5 are averaged across the four horizon lengths in Table 10.

Chronicle. We report two variants. *ZS* is autoregressive next-patch forecasting, with no head training; predictions are denormalized via the inverse of the patch-level standardization in Eq. 1. *LP* adds a forecasting head on top of the frozen backbone with joint text and TS input; only the head's parameters are updated, using MSE loss with the same horizon and split settings as the baselines.

LoRA ablation settings. For the Stage 1 vs. Stage 2 ablation (Section 5.3), Time-MMD LoRA uses 3 seeds with $1{,}147{,}552$ trainable parameters (frozen head: $688{,}800$), and UCR/UEA LoRA uses 3 seeds with $460{,}802$ trainable parameters; UCR/UEA LP results use 5 seeds and reproduce Table 3, with uncertainty computed across seed-level benchmark aggregates.

F.4 UCR Time Series Classification

The 14 UCR datasets used in the main paper are GunPoint, Coffee, ECG200, FaceFour, OSULeaf, SwedishLeaf, SyntheticControl, Trace, TwoPatterns, Wafer, Earthquakes, ShapeletSim, Chinatown, and ItalyPowerDemand, with the official aeon train/test splits in all cases. Supervised DL baselines (Autoformer, DLinear, FEDformer, Informer, iTransformer, PatchTST, TimesNet) are trained per-dataset for 30 epochs at lr 10^{-3} with batch size 16. TS foundation model baselines (Chronos-2, Moirai-2, TimesFM) and Chronicle are evaluated with a learned linear probe on frozen embeddings under the same aeon split; we train for 200 epochs at lr 10^{-2} with weight decay 0 and batch size 64 on patch-32, joint multivariate, channel-aware, instance z-scored embeddings, while the foundation-model probes use the published linear-probe protocol from each model's reference implementation.

G Extended Related Work

This appendix provides a comprehensive discussion of the three research threads that Chronicle builds upon and extends.

G.1 Time Series Foundation Models

Foundation models for time series aim to generalize zero-shot across domains and frequencies, analogous to how language models generalize across tasks (Liang et al., 2024). The field has coalesced around two main input representations.

Scalar tokenization. Chronos (Ansari et al., 2024) tokenizes real-valued series via scaling and quantization and trains a T5-family encoder–decoder with cross-entropy loss. LLMTIME (Gruver et al., 2023) and Chat-TS take scalar tokenization to its extreme by representing values as digit strings and querying frozen LLMs, demonstrating useful numerical priors at the cost of verbosity and computational overhead.

Patch-based tokenization. A complementary line represents series as *patches*, contiguous windows projected to dense embeddings. PatchTST (Nie et al., 2023) introduced patch-based tokenization for supervised forecasting, demonstrating that "a time series is worth 64 words." TimesFM (Das et al., 2024) scaled a decoder-only patch transformer to 200M parameters with pretraining on large-scale corpora. PatchTST-FM (Wen et al., 2026) revisited the generic transformer as a foundation model baseline, adding gated residual projections, a 99-quantile output head, and cumulative patch masking (CPM), achieving state-of-the-art on GIFT-Eval at 260M parameters. Moirai (Woo et al., 2024) addressed heterogeneous frequencies with frequency-specific projections within a masked encoder; Moirai-2 (Liu et al., 2026) extended this with improved architectures and training. MOMENT (Goswami et al., 2024) trains a backbone with lightweight task-specific decoders for multiple tasks simultaneously, while UniTS (Gao et al., 2024) pursues multi-task generalization via unified token representations. TiRex (Auer et al., 2025), Toto (Cohen et al., 2025), and YingLong (Wang et al., 2025) represent further entries on the GIFT-Eval leaderboard.

Our architecture draws on PatchTST-FM and TimesFM (patch-based, decoder-only, quantile output) but differs fundamentally in being trained jointly with natural language from scratch. We compare against these models on GIFT-Eval (Section 5.1.2), where published leaderboard scores provide a direct zero-shot comparison, and on UCR classification (Section 5.2.1), where Chronos-2 and Moirai-2 frozen embeddings serve as foundation model baselines.

G.2 Multimodal Text and Time Series Models

A rapidly growing body of work connects language models to time series. We organize these approaches by their architectural paradigm and highlight their evaluation limitations.

Frozen LLM approaches. LLMTIME (Gruver et al., 2023) queries frozen GPT-3/LLaMA with digit-string representations of time series, demonstrating zero-shot forecasting capability but inheriting the full computational cost of large language models and producing no learnable temporal representations. GPT4MTS (Jia et al., 2024) constructs multimodal prompts combining textual context with numerical time series data and feeds them to frozen LLMs. Zhou et al. (2023) showed that frozen LLMs, fine-tuned only at input/output projections, yield competitive forecasting performance. However, these approaches treat the language model as a black box; the temporal representations are constrained to the text embedding space, which was never designed for continuous numerical data.

Adapted LLM approaches. Time-LLM (Jin et al., 2024) reprograms patch embeddings into text prototypes with a frozen LLM backbone. GPT4TS (Zhou et al., 2023) fine-tunes only the normalization layers of GPT-2. ChatTS (Xie et al., 2025) encodes time series patches through a shallow MLP and concatenates them with text embeddings before feeding a fine-tuned Qwen2.5-14B backbone, using synthetic QA pairs (TSEvol) to address data scarcity. ChatTime (Wang et al., 2024) instruction-fine-tunes a decoder-only LLM for bidirectional text and time series generation, achieving 99.9% of Chronos's zero-shot accuracy with only 4% of the pretraining data by leveraging the pretrained LLM's existing representations. MoAT (Lee et al., 2024) introduces a two-stage framework: first optimizing forecasts from decomposed time series and text embeddings, then fusing via an offline MLP synthesis. TaTs (Li et al., 2026) treats text embeddings as auxiliary time series variables, capturing what the authors call "chronological textual resonance," periodic patterns in text representations that mirror the numerical series. MSE-ITT (Koval et al., 2025) extends LLaMA-3-8B with modality-specific expert layers for financial forecasting from interleaved text and time series. Time-VLM (Zhong et al., 2025) bridges temporal, visual, and textual modalities using frozen vision-language models.

All of these approaches share a fundamental limitation: *they start from a pretrained language model*, meaning the backbone's representations were shaped entirely by text before any exposure to temporal data. The time series modality must adapt to a representational space that was not designed for it, and the resulting models inherit the language model's parameter count, vocabulary, and computational requirements, even when the downstream task is purely temporal.

Unimodal-batch multimodal pretraining. Combining modality-specific unimodal batches with a smaller paired component has precedent in vision–language pretraining: VLMo (Bao et al., 2022a) and VL-BEiT (Bao et al., 2022b), among others, combine image-only, text-only, and paired data. We do not claim unimodal batching is novel across multimodal research; our claim is specific to text and time series, where no prior model is trained from scratch on both modalities within one backbone, and where web-scale paired corpora comparable to image–text data do not exist.

Fusion and benchmark approaches. Time-MMD (Liu et al., 2024a) provides a multi-domain benchmark pairing time series with textual reports across nine domains and introduces **MM-TSFlib**, a fusion library that has become a standard reference protocol for text-augmented time series. Under MM-TSFlib, a trainable time series encoder (e.g., DLinear, PatchTST, TimesNet) is paired with a frozen pretrained text encoder (BERT or GPT-2) and a trainable MLP fusion head; the TS encoder and head are trained end-to-end on each downstream dataset, while the text encoder remains frozen. Subsequent text-augmented time series studies have adopted MM-TSFlib as a benchmarking baseline; we use it directly as our multimodal fusion comparison in Section 5.2. TimeCAP (Lee et al., 2025) uses LLM agents to generate contextual descriptions and combines predictions from a multimodal predictor with a pretrained LLM. Recent surveys on multimodal time series (Liu et al., 2025) provide comprehensive taxonomies of fusion strategies. Zhang et al. (2025) systematically investigate when multimodal integration yields gains, finding that benefits are "highly condition-dependent" and "neither universal nor always aligned with intuition."

Critical evaluation gap. A striking pattern across this literature is the narrowness of evaluation. ChatTime compares against Chronos and GPT4TS for forecasting but does not evaluate language understanding. ChatTS evaluates time series understanding but not against GIFT-Eval, UCR, or NLU benchmarks. MoAT, TaTs, and GPT4MTS evaluate only multimodal forecasting on their own datasets. MSE-ITT compares against multimodal and financial baselines but does not benchmark against dedicated TSFMs on standard time series tasks. *No prior multimodal text and time series model has been evaluated against both dedicated TSFMs on time series benchmarks and dedicated*

LLMs on language understanding benchmarks. This creates a fundamental ambiguity: when a multimodal model reports improved forecasting, it is unclear whether the improvement stems from genuine cross-modal learning or simply from the text providing complementary information that a strong TSFM baseline would render unnecessary. Our evaluation protocol addresses this gap directly by testing Chronicle against the best models in *each* modality on *their own* benchmarks.

Negative results on LLMs for time series. Several recent works have questioned the value of language model priors for temporal tasks. Tan et al. (2024) ablated three top-tier LLM-for-TS methods and found that LLMs "fail to convincingly improve time series forecasting" while "significantly increasing computational costs." Re-initializing LLM weights prior to forecasting had no impact on performance, suggesting that pretrained language representations do not transfer to temporal modeling. Merrill et al. (2024) found that LLMs struggle to reason about time series encoded as text, motivating modality-native representations. These findings suggest that simply bolting time series onto a language model, the approach taken by all prior multimodal work, is fundamentally limited. Chronicle takes a different path: rather than adapting a language model for time series, we train a single model for both from scratch, allowing the architecture to develop representations suitable for both modalities simultaneously.

G.3 Small Language Models

GPT-2 (Radford et al., 2019) demonstrated that decoder-only transformers trained with next-token prediction produce capable few-shot learners. Subsequent models have pushed zero-shot language understanding to strong levels at sub-1B scale: Qwen2 (Yang et al., 2024) at 500M parameters achieves 0.476 average accuracy on our NLU suite; LLaMA-3.2 (Grattafiori et al., 2024) at 1.2B achieves 0.531; Gemma-3-270M-PT (Gemma Team, 2025) at 270M achieves 0.406; and LFM-2-350M (Amini et al., 2025) at 350M achieves 0.449. These models represent the current frontier of what is achievable with compact transformer architectures trained exclusively on text. They are also typically trained on hundreds of billions to several trillion text tokens, substantially more than our total compute budget allows. We compare Chronicle against all five to verify that, under our text-heavy 92/8 mix, devoting approximately 8% of training compute to time series does not cause catastrophic interference. The fact that Chronicle matches Gemma-3-270M-PT despite its dual training objective establishes an important proof point: a shared transformer backbone can accommodate both text and time series without degrading either modality's performance relative to scale-matched specialists, provided the token mix is chosen to keep the language stream competitive.

H Synthetic Training Data

Online synthetic augmentation is applied to time-series-only batches during training, controlled by a per-batch probability of 0.20. Series are generated on-the-fly in a background worker thread; generation takes approximately 1 to 3 ms per series at length 32k, introducing no data-loading bottleneck.

A bank of 33 kernel generators is defined at module load time, spanning smooth trends, periodic patterns, stochastic processes, discrete waveforms, and noise models (Table 11). For each synthetic series, 2–5 kernels are sampled without replacement and combined via one of two modes:

- **Additive** (80%): $x(t) = \sum_i k_i(t)$.
- **Mixed multiplicative** (20%): kernels are combined iteratively; each subsequent kernel is either added or multiplied (after shifting to a positive range) with probability 0.40 per kernel.

All kernels operate on normalized time $t_n = \mathrm{linspace}(0, 1, L)$ and are vectorized. Duplicate entries in the bank increase sampling frequency for empirically useful kernels (RBF short/long, periodic short/long, rational quadratic, damped oscillation), following the emphasis in Chronos KernelSynth (Ansari et al., 2024). After composition, Inf values are clipped to ± 5 before combination and $\pm 10^7$ after; output is cast to float32. Beyond KernelSynth, with 50% probability per time-series batch we additionally apply jitter (additive Gaussian noise), scaling (multiplicative perturbation), and intra-batch mixup.

Table 11: KernelSynth generator bank (33 entries). Duplicates are listed in the "# entries" column and increase sampling weight for empirically useful kernels.

Category	Implementation	# entries
RBF smooth	$\frac{1}{R}\sum_r \cos(\omega_r t + \phi_r)$, $\omega_r \sim \mathcal{N}(0, 1/\ell_s)$, R=32 RFF	5
Periodic	$A\sin(2\pi t/p + \phi)$, $A \sim \mathrm{Unif}(0.5, 2)$	5
Periodic + harmonics	Base + 2 overtones at amplitudes $A/2$, $A/3$ with independent phases	1
Rational Quadratic	RFF with Gamma-distributed scales: $\omega_r \sim \mathrm{scale} \cdot \mathcal{N}(0,1)/\ell_s$, $\mathrm{scale} \sim \Gamma(\alpha, 1/\alpha)$	2
Linear trend	$at + b$, $a \sim \mathrm{Unif}(-3, 3)$	1
Polynomial	$\mathrm{polyval}(\mathbf{c}, t_{\mathrm{norm}})$, coefficients $\sim$ Unif	2
Log trend	$c \cdot \log(t)$, $c \sim \mathrm{Unif}(-2, 2)$	1
Random walk	Cumulative sum of Gaussian steps; drift $\mu \sim \mathrm{Unif}(-0.01, 0.01)$	2
Level shifts	1–3 abrupt shifts at random positions in the middle 80%	1
Discrete waves	Period $\in [0.05, 0.40]$, amplitude $\in [0.5, 2.0]$, random phase/offset	3
Damped oscillation	$Ae^{-\gamma t}\sin(2\pi t/p + \phi)$, $\gamma \sim \mathrm{Unif}(1, 8)$	2
White noise	$\mathcal{N}(0, \sigma)$	3
Heteroskedastic noise	$\epsilon_t \sim \mathcal{N}(0,\ \sigma \cdot e^{0.5k(t)})$, envelope modulated by RBF-drawn signal	1
Periodic noise	$\mathcal{N}(0, 0.3) \cdot (1 + A(\sin(2\pi t/p + \phi) \cdot 0.5 + 0.5))$	1
Step function	3–11 constant-level segments with random transitions	1
Exponential growth/decay	$e^{rt} - 1$, $r \sim \mathrm{Unif}(-3, 3)$	1
Constant	Flat baseline $c \sim \mathrm{Unif}(-2, 2)$	1
Total		**33**

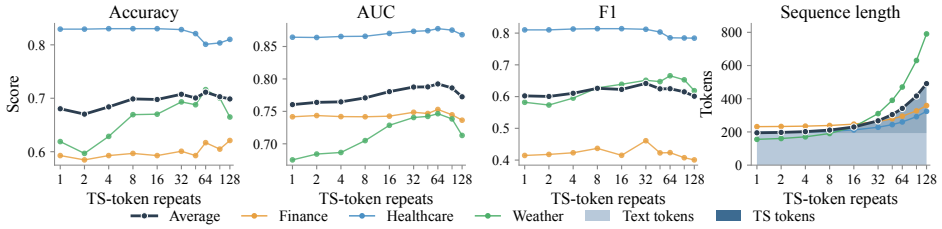

Figure 4: **Effect of TS-token repetition on multimodal classification.** Accuracy (left), AUC (middle), and macro-F1 (right) as a function of TS-token repeats r, evaluated on the three TimeCAP domains and averaged (dashed black). Repetition rebalances the text–TS token ratio in the shared sequence; performance peaks near r=64 then degrades as attention dilutes across identical copies.

I TS-Token Repetition for Short Time Series

Chronicle processes both modalities autoregressively within a shared backbone, so the relative sequence length of each modality directly influences performance. For short-series tasks such as TimeCAP, where the time series is often a single patch, the paired text caption dominates the mean-pooled representation and leaves temporal features underweighted. We address this by repeating the TS-token block r times within the input, appending the same patch embeddings r times without altering the underlying series, rebalancing the modality ratio without any architectural change or backbone retraining. Figure 4 sweeps $r \in [1, 128]$ on the three TimeCAP domains. Averaged across domains, accuracy improves from 0.681 at r=1 to 0.711 at r=64, AUC from 0.760 to 0.790, and macro-F1 from 0.602 to 0.642, before degrading at $r \geq 96$ as attention is diluted across many identical copies. Weather, which has the longest natural TS context, shows the largest gain; Finance and Healthcare are largely flat, limited by class-imbalance ceilings rather than representational quality. The main-paper MM-CLS result (Table 4) uses r=1 and tuned setting r=64 to compare to MM-TSFlib and FM Fusion baselines; this ablation shows roughly 3 accuracy and 4 AUC points of headroom with tuned r.

J Full Multimodal Classification Results

Table 12 reports per-domain accuracy, macro-F1, and AUC for all methods on TimeCAP, including both the primary Stage 2 checkpoint and the Stage 1 ablation checkpoint (Section 5.3). Within each

Table 12: Multimodal classification on TimeCAP by domain. Values are mean ± standard deviation over 3 seeds (0, 1, 2).

Cat.	Model	Weather		Finance		Healthcare		Average	
		F1 ↑	AUC ↑	F1 ↑	AUC ↑	F1 ↑	AUC ↑	F1 ↑	AUC ↑
MM-TSFlib	DLin+BERT	0.600 ± 0.013	0.676 ± 0.011	0.367 ± 0.010	0.641 ± 0.036	0.796 ± 0.051	0.902 ± 0.030	0.588 ± 0.016	0.739 ± 0.024
	DLin+GPT2	0.572 ± 0.009	0.643 ± 0.017	0.326 ± 0.057	0.655 ± 0.018	0.793 ± 0.024	0.875 ± 0.036	0.564 ± 0.026	0.724 ± 0.017
	PTST+BERT	0.566 ± 0.017	0.613 ± 0.021	0.368 ± 0.052	0.643 ± 0.024	0.800 ± 0.004	0.900 ± 0.024	0.578 ± 0.022	0.719 ± 0.022
	PTST+GPT2	0.461 ± 0.066	0.556 ± 0.057	0.363 ± 0.010	0.682 ± 0.019	0.793 ± 0.009	0.882 ± 0.033	0.539 ± 0.021	0.707 ± 0.034
	TNet+BERT	0.597 ± 0.057	0.712 ± 0.013	0.364 ± 0.015	0.635 ± 0.041	0.808 ± 0.017	0.902 ± 0.026	0.589 ± 0.021	0.750 ± 0.026
	TNet+GPT2	0.617 ± 0.037	0.729 ± 0.018	0.330 ± 0.056	0.660 ± 0.048	0.786 ± 0.032	0.871 ± 0.036	0.577 ± 0.019	0.754 ± 0.028
FM Fusion	BERT+Chr2	0.588 ± 0.010	0.631 ± 0.018	0.388 ± 0.054	0.647 ± 0.028	0.794 ± 0.012	0.902 ± 0.029	0.590 ± 0.021	0.726 ± 0.023
	BERT+Moi2	0.617 ± 0.022	0.702 ± 0.023	0.351 ± 0.041	0.651 ± 0.026	0.797 ± 0.019	0.901 ± 0.033	0.588 ± 0.004	0.751 ± 0.025
	BERT+TFM	0.436 ± 0.021	0.577 ± 0.030	0.273 ± 0.000	0.503 ± 0.102	0.786 ± 0.009	0.898 ± 0.030	0.498 ± 0.006	0.659 ± 0.023
	GPT2+Chr2	0.284 ± 0.121	0.542 ± 0.031	0.294 ± 0.037	0.597 ± 0.143	0.785 ± 0.023	0.880 ± 0.035	0.455 ± 0.043	0.673 ± 0.056
	GPT2+Moi2	0.588 ± 0.015	0.690 ± 0.022	0.252 ± 0.150	0.643 ± 0.029	0.786 ± 0.037	0.885 ± 0.024	0.542 ± 0.038	0.739 ± 0.015
	GPT2+TFM	0.423 ± 0.000	0.562 ± 0.029	0.273 ± 0.000	0.451 ± 0.077	0.745 ± 0.101	0.870 ± 0.054	0.480 ± 0.034	0.628 ± 0.018
Chronicle	Stage 1 LP (r=1)	0.564 ± 0.021	0.641 ± 0.034	0.426 ± 0.024	0.683 ± 0.024	0.790 ± 0.028	0.874 ± 0.048	0.593 ± 0.021	0.733 ± 0.030
	Stage 1 LP (r=64)	0.608 ± 0.018	0.705 ± 0.010	0.423 ± 0.009	0.662 ± 0.003	0.793 ± 0.021	0.867 ± 0.042	0.608 ± 0.010	0.745 ± 0.014
	Stage 1 LoRA (r=1)	0.602 ± 0.012	0.641 ± 0.029	0.374 ± 0.032	0.684 ± 0.030	0.828 ± 0.011	0.891 ± 0.017	0.601 ± 0.011	0.739 ± 0.017
	Stage 1 LoRA (r=64)	0.636 ± 0.006	0.717 ± 0.018	0.295 ± 0.038	0.669 ± 0.005	0.820 ± 0.042	0.904 ± 0.039	0.584 ± 0.024	0.763 ± 0.011
	Stage 2 LP (r=1)	0.561 ± 0.039	0.635 ± 0.040	0.428 ± 0.011	0.685 ± 0.014	0.794 ± 0.042	0.872 ± 0.048	0.594 ± 0.025	0.731 ± 0.029
	Stage 2 LP (r=64)	0.606 ± 0.013	0.696 ± 0.008	0.406 ± 0.014	0.683 ± 0.010	0.804 ± 0.025	0.872 ± 0.038	0.605 ± 0.011	0.750 ± 0.014
	Stage 2 LoRA (r=1)	0.599 ± 0.013	0.644 ± 0.024	0.365 ± 0.080	0.656 ± 0.074	0.821 ± 0.013	0.892 ± 0.019	0.595 ± 0.030	0.731 ± 0.032
	Stage 2 LoRA (r=64)	0.629 ± 0.033	0.706 ± 0.066	0.386 ± 0.010	0.667 ± 0.047	0.823 ± 0.016	0.897 ± 0.036	0.613 ± 0.011	0.757 ± 0.014

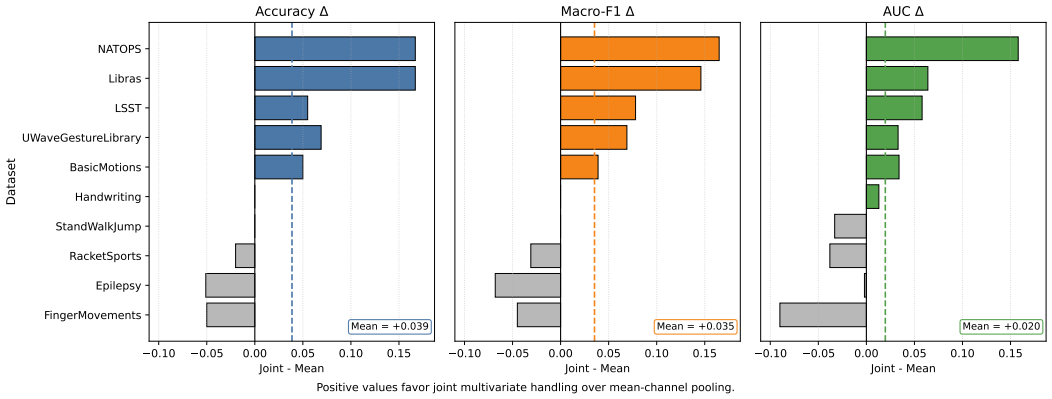

Figure 5: **Effect of channel-aware multivariate handling on UEA classification.** Bars show the per-dataset delta between joint multivariate handling and mean-channel pooling (joint minus mean) for accuracy, macro-F1, and AUC. Positive values favor joint channel-aware handling. Averaged across the 10 multivariate UEA datasets, joint handling improves accuracy by $+0.039$, macro-F1 by $+0.035$, and AUC by $+0.020$.

baseline category, BERT-paired models appear before GPT2-paired models. Summary averages are reported in Table 4.

J.1 Effect of Channel-Aware Multivariate Handling

To assess the value of preserving channel identity in multivariate time-series inputs, we compare the default joint multivariate representation used by Chronicle, which includes the channel ramp $\mathbf{c}$ in the patch features, against a mean-channel pooling variant that averages channels before encoding and therefore removes channel identity. All values are for the frozen-backbone linear head.

Figure 5 shows the per-dataset deltas (joint minus mean pooling) on the 10 multivariate UEA datasets used in our evaluation. Averaged across datasets, joint channel-aware handling improves accuracy by $+0.039$, macro-F1 by $+0.035$, and AUC by $+0.020$. The largest gains appear on LIBRAS and NATOPS ($+0.167$ accuracy on both), with additional improvements on LSST and UWAVEGESTURELIBRARY. Some datasets favor mean-channel pooling (EPILEPSY, FINGERMOVEMENTS, and RACKETSPORTS), while HANDWRITING and STANDWALKJUMP are effectively unchanged in accuracy and F1. Overall, the results indicate that retaining channel identity is beneficial on average for multivariate classification, supporting the use of channel-aware patch features for multivariate inputs.

Table 13: Per-dataset UCR/UEA time-series classification results, including both the primary Stage 2 checkpoint and the Stage 1 ablation checkpoint. Values are means over five seeds.

Suite	Dataset	Informer			TimesNet			Autoformer			DLinear			iTransformer			FEDformer			PatchTST			Chronos-2			TimesFM			Moirai-2			Chronicle Stage 1			Chronicle Stage 2		
		Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
UCR	GunPoint	0.576	0.573	0.634	0.580	0.539	0.603	0.605	0.599	0.716	0.757	0.756	0.840	0.708	0.701	0.842	0.656	0.650	0.779	0.653	0.648	0.711	0.528	0.397	0.548	0.712	0.682	0.862	0.931	0.931	0.981	0.919	0.919	0.965	0.851	0.850	0.936
	Coffee	0.536	0.349	0.397	0.536	0.349	0.588	0.536	0.349	0.525	0.900	0.871	1.000	0.593	0.441	0.962	0.536	0.349	0.563	0.536	0.349	0.673	0.536	0.410	0.555	0.571	0.426	0.987	0.964	0.964	0.995	0.893	0.892	0.988	0.893	0.870	0.996
	ECG200	0.642	0.441	0.575	0.804	0.791	0.894	0.806	0.779	0.873	0.814	0.796	0.904	0.866	0.855	0.942	0.796	0.764	0.906	0.862	0.846	0.920	0.672	0.468	0.572	0.840	0.818	0.939	0.820	0.799	0.892	0.846	0.826	0.946	0.848	0.833	0.934
	FaceFour	0.186	0.078	0.586	0.395	0.288	0.785	0.227	0.116	0.473	0.541	0.516	0.923	0.627	0.629	0.888	0.223	0.102	0.556	0.350	0.296	0.687	0.236	0.123	0.521	0.609	0.607	0.879	0.582	0.552	0.810	0.864	0.866	0.988	0.759	0.731	0.984
	OSULeaf	0.350	0.270	0.749	0.406	0.319	0.825	0.513	0.457	0.824	0.360	0.313	0.666	0.406	0.350	0.763	0.556	0.485	0.863	0.559	0.510	0.863	0.184	0.055	0.519	0.401	0.311	0.842	0.719	0.693	0.937	0.583	0.555	0.876	0.579	0.542	0.888
	SwedishLeaf	0.533	0.497	0.947	0.381	0.314	0.923	0.675	0.647	0.976	0.763	0.756	0.969	0.843	0.841	0.990	0.711	0.701	0.975	0.859	0.858	0.992	0.096	0.054	0.554	0.599	0.564	0.966	0.854	0.854	0.992	0.848	0.847	0.993	0.834	0.831	0.987
	SyntheticControl	0.835	0.829	0.974	0.591	0.519	0.990	0.680	0.666	0.934	0.878	0.868	0.986	0.833	0.821	0.981	0.992	0.992	0.999	0.926	0.926	0.993	0.239	0.139	0.567	0.903	0.904	0.989	0.860	0.860	0.983	0.727	0.728	0.930	0.727	0.727	0.931
	Trace	0.746	0.703	0.938	0.506	0.420	0.931	0.910	0.903	0.975	0.484	0.444	0.830	0.488	0.385	0.837	1.000	1.000	1.000	0.878	0.868	0.984	0.288	0.189	0.581	0.630	0.573	0.891	0.802	0.804	0.951	0.936	0.932	0.999	0.848	0.827	0.988
	TwoPatterns	0.355	0.253	0.704	0.999	0.999	1.000	0.357	0.304	0.617	0.850	0.850	0.975	0.820	0.819	0.960	0.932	0.932	0.992	0.826	0.825	0.957	0.281	0.153	0.525	0.655	0.627	0.916	0.634	0.632	0.863	0.778	0.777	0.948	0.790	0.787	0.945
	Wafer	0.982	0.950	0.982	0.951	0.863	0.933	0.988	0.969	0.998	0.943	0.834	0.853	0.995	0.987	0.998	0.992	0.979	0.998	0.974	0.931	0.984	0.894	0.488	0.537	0.941	0.813	0.919	0.996	0.990	1.000	0.991	0.977	0.999	0.987	0.966	0.998
	Earthquakes	0.748	0.428	0.680	0.748	0.428	0.666	0.748	0.428	0.681	0.630	0.545	0.583	0.732	0.471	0.500	0.748	0.428	0.688	0.738	0.440	0.619	0.748	0.428	0.506	0.581	0.436	0.507	0.722	0.468	0.656	0.760	0.536	0.645	0.774	0.555	0.682
	ShapeletSim	0.499	0.333	0.536	0.779	0.737	0.947	0.559	0.482	0.662	0.526	0.523	0.521	0.491	0.489	0.496	0.656	0.583	0.972	0.519	0.445	0.546	0.496	0.357	0.502	0.709	0.706	0.757	0.789	0.788	0.856	0.603	0.603	0.623	0.608	0.605	0.682
	Chinatown	0.574	0.388	0.646	0.412	0.363	0.985	0.706	0.671	0.881	0.908	0.876	0.987	0.969	0.962	0.994	0.845	0.835	0.988	0.904	0.880	0.959	0.684	0.487	0.599	0.930	0.913	0.976	0.983	0.978	0.996	0.977	0.971	0.996	0.960	0.953	0.995
	ItalyPowerDemand	0.557	0.542	0.717	0.829	0.819	0.983	0.741	0.740	0.815	0.952	0.952	0.990	0.969	0.969	0.993	0.848	0.848	0.913	0.970	0.970	0.988	0.591	0.458	0.598	0.927	0.926	0.976	0.946	0.946	0.991	0.953	0.953	0.991	0.955	0.955	0.992
UEA	BasicMotions	0.980	0.980	1.000	0.980	0.980	1.000	0.995	0.995	1.000	0.390	0.372	0.666	0.450	0.415	0.725	1.000	1.000	1.000	0.380	0.327	0.626	0.290	0.174	0.543	0.765	0.762	0.938	0.960	0.960	0.998	0.925	0.924	0.995	0.945	0.946	0.990
	Epilepsy	0.478	0.481	0.701	0.848	0.849	0.977	0.642	0.645	0.871	0.370	0.358	0.575	0.309	0.297	0.532	0.922	0.920	0.987	0.877	0.875	0.975	0.272	0.138	0.502	0.897	0.894	0.989	0.943	0.945	0.996	0.945	0.943	0.992	0.939	0.934	0.987
	NATOPS	0.716	0.700	0.953	0.881	0.878	0.982	0.808	0.803	0.970	0.716	0.714	0.933	0.637	0.629	0.900	0.896	0.894	0.988	0.693	0.682	0.918	0.220	0.115	0.551	0.511	0.492	0.823	0.513	0.503	0.827	0.544	0.535	0.839	0.523	0.512	0.826
	RacketSports	0.728	0.739	0.912	0.814	0.829	0.944	0.812	0.824	0.941	0.687	0.697	0.884	0.608	0.615	0.845	0.851	0.864	0.948	0.593	0.597	0.850	0.322	0.183	0.555	0.668	0.659	0.880	0.693	0.697	0.894	0.659	0.662	0.894	0.701	0.711	0.890
	UWaveGestureLibrary	0.410	0.397	0.828	0.606	0.608	0.907	0.520	0.507	0.887	0.801	0.792	0.964	0.676	0.668	0.925	0.652	0.643	0.914	0.785	0.783	0.944	0.139	0.053	0.525	0.221	0.132	0.740	0.500	0.508	0.863	0.627	0.615	0.921	0.636	0.622	0.919
	Handwriting	0.132	0.064	0.692	0.147	0.079	0.774	0.147	0.078	0.685	0.132	0.097	0.613	0.144	0.106	0.614	0.343	0.271	0.883	0.093	0.066	0.595	0.041	0.009	0.510	0.107	0.087	0.626	0.127	0.115	0.648	0.172	0.148	0.661	0.146	0.116	0.659
	Libras	0.569	0.559	0.923	0.813	0.810	0.988	0.692	0.693	0.976	0.554	0.535	0.900	0.472	0.440	0.898	0.816	0.817	0.990	0.697	0.686	0.940	0.124	0.067	0.542	0.431	0.398	0.880	0.598	0.584	0.928	0.527	0.526	0.926	0.543	0.532	0.911
	LSST	0.573	0.328	0.850	0.591	0.323	0.875	0.606	0.337	0.857	0.303	0.055	0.547	0.475	0.191	0.779	0.590	0.324	0.862	0.416	0.144	0.746	0.317	0.042	0.522	0.242	0.111	0.638	0.285	0.138	0.630	0.535	0.317	0.835	0.525	0.286	0.821
	FingerMovements	0.494	0.447	0.511	0.492	0.454	0.474	0.510	0.416	0.531	0.492	0.482	0.517	0.538	0.509	0.555	0.502	0.437	0.508	0.502	0.487	0.506	0.500	0.371	0.502	0.472	0.445	0.468	0.544	0.544	0.588	0.520	0.519	0.531	0.556	0.553	0.563
	StandWalkJump	0.360	0.269	0.619	0.387	0.252	0.572	0.280	0.167	0.468	0.533	0.537	0.673	0.413	0.387	0.572	0.280	0.168	0.447	0.440	0.395	0.569	0.333	0.167	0.500	0.333	0.225	0.499	0.360	0.352	0.603	0.533	0.508	0.805	0.560	0.546	0.771

K PER-DATASET UCR/UEA CLASSIFICATION RESULTS