ADAPTING TIME SERIES FOUNDATION MODELS FOR EVENT-CONDITIONED MULTIMODAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Unimodal Time series foundation models (TSFMs) have demonstrated strong zero-shot forecasting capabilities through large-scale pretraining on heterogeneous time series data. However, they cannot natively incorporate natural-language descriptions of external events, such as promotional events, holidays, and policy changes, that provide predictive information not recoverable from historical observations alone. Existing work either relies on LLM-based reasoning that does not guarantee accurate numerical forecasts, or requires costly joint multimodal pretraining from scratch. To address this, we propose **MM-Time**, a parameter-efficient framework that equips pretrained TSFMs with event-aware forecasting through multimodal continued pretraining. To bridge the semantic–effect gap in which semantically distinct events exert identical numerical impacts, we introduce an Event-Effect Prototype Router. It maps event descriptions and temporal attributes to sparse combinations of shared learnable prototypes under forecasting supervision, encouraging the sharing of forecast-relevant information across events. Furthermore, to address temporal alignment over event durations, we propose Event Interval Rotary-Bias Attention, which conditions temporal attention on event boundary positions and incorporates an interval-distance bias to favor temporally relevant events. We adapt three state-of-the-art TSFMs, Chronos-2, Falcon-2, and TiRex-2, and show that MM-Time consistently improves event-conditioned forecasting across multiple downstream datasets.

1 INTRODUCTION

Unimodal Time series foundation models (TSFMs) learn transferable temporal patterns through large-scale pretraining on heterogeneous data, enabling zero-shot forecasting across domains (Ansari et al., 2025; Liu et al., 2026a; Khwaja et al., 2026; Das et al., 2024; Liu et al., 2025c; 2024b). However, historical observations alone may not fully reflect future changes associated with external events, such as promotions, holiday activities, and policy changes. Natural-language descriptions of these events can provide additional predictive information before their effects appear in numerical observations. Incorporating event information available at forecast time can therefore help unimodal TSFMs make predictions more responsive to upcoming changes.

Existing work has explored combining language and time series through LLM-based reasoning or forecasting (Guan et al., 2026; Xie et al., 2025; Lu et al., 2026; Jin et al., 2024) and joint multimodal pretraining (Wu et al., 2025; Chen et al., 2026; Quinlan et al., 2026). LLM-based methods can interpret event semantics, but semantic understanding alone does not necessarily translate into accurate numerical forecasts. Joint multimodal pretraining requires learning both numerical dynamics and cross-modal interactions, typically demanding large-scale multimodal training data and computational resources. A more efficient and promising paradigm is to reuse the numerical representations of pretrained unimodal TSFMs and incorporate event information through multimodal continued pretraining. This motivates our central question:

> *How can parameter-efficient multimodal continued pretraining equip existing TSFMs with transferable event-aware forecasting capabilities?*

Adapting pretrained TSFMs for event-aware forecasting presents two key challenges. ❶ The first is the **semantic–effect gap**. Natural language semantics do not directly align with numerical forecast-

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.

ing targets: different event descriptions can lead to the same temporal patterns, while identical events might trigger vastly different temporal patterns. Since standard semantic similarity fails to capture these quantitative effects, it is crucial to learn latent effect representations that capture the underlying predictive impact of surface text, enabling robust knowledge transfer across diverse events. ❷ The second is **event–forecast temporal alignment**. Real-world events, such as promotions or holidays, span continuous intervals rather than single time points, possessing two key properties: explicit *start and end boundaries*, and time-varying *locality decay* over the forecast horizon. A single timestamp fails to capture either aspect, leaving the model unable to determine where an event applies or how its influence dissipates. This necessitates encoding interval boundaries and explicitly modeling their relative temporal relationship to individual forecast steps.

To address these challenges, we propose **MM-Time**, a parameter-efficient framework for adapting pretrained unimodal TSFMs through multimodal continued pretraining. *To bridge the semantic–effect gap*, we introduce a shared dictionary of latent effect prototypes as an intermediate representation between event descriptions and numerical forecasts. The proposed Event-Effect Prototype Router maps event text and structured temporal features to sparse combinations of these prototypes, yielding event-specific effect tokens. Rather than relying on superficial semantic matching, the router and prototypes are jointly trained under forecasting supervision to capture predictive structures shared across events, effectively decoupling surface text from its underlying numerical impact. *To address temporal alignment*, the proposed Event Interval Rotary-Bias Attention aligns these tokens with future forecast patches through cross-attention. To explicitly handle interval-based events, Interval RoPE encodes the relative timing between forecast patches and event boundaries, while Interval Bias introduces a soft locality prior by penalizing attention according to distance outside the event interval. This design allows the pretrained numerical backbone to incorporate event information according to both its forecast-relevant content and temporal relevance.

We initialize each backbone from its pretrained checkpoint and perform multimodal continued pretraining on time series paired with event descriptions and temporal information. MM-Time incorporates event representations through gated residual connections at intermediate or late backbone layers. We apply Weight-Decomposed Low-Rank Adaptation (DoRA) (Liu et al., 2024a) to selected linear projections in the final backbone layers after event fusion and fine-tune the pretrained forecasting head. The event-conditioning modules, DoRA adapters, and forecasting head are jointly optimized, while the remaining backbone parameters are kept fixed. We evaluate MM-Time using Chronos-2 (Ansari et al., 2025), Falcon-2 (Xia et al., 2026), and TiRex-2 (Podest et al., 2026) on different downstream forecasting datasets to demonstrate both cross-backbone applicability and robust downstream generalization. Our contributions are summarized as follows:

1. **Parameter-efficient adaptation of pretrained unimodal TSFMs.** We introduce MM-Time, a multimodal continued pretraining framework that reuses pretrained numerical representations for event-conditioned forecasting. By leveraging event injection and parameter-efficient adaptation, we provide a unified, plug-and-play adaptation design across heterogeneous TSFM backbones.
2. **Latent event-effect modeling and interval-aware alignment.** We propose an Event-Effect Prototype Router that represents events through sparse combinations of shared prototypes to model latent numerical impacts. We further propose Event Interval Rotary-Bias Attention, which combines rotary encoding of event boundaries with a distance-based bias to incorporate temporal relationships into event attention
3. **Evaluation across backbones and downstream datasets.** We evaluate MM-Time with three prominent unimodal TSFMs across different downstream datasets. The results demonstrate the effectiveness of event-conditioned forecasting and the applicability of the framework across different backbone architectures.

2 Related Work

2.1 Time Series Foundation Models

Unimodal Time series foundation models (TSFMs) learn transferable temporal patterns through large-scale pretraining, supporting forecasting across domains (Liu et al., 2024b; Das et al., 2024; Shi et al., 2025; Wang et al., 2025b; Woo et al., 2024; Ekambaram et al., 2024; Liu et al., 2025c;

Cohen et al., 2025; Ansari et al., 2024). Recent studies further extend their capabilities. Chronos-2 (Ansari et al., 2025) introduces group attention for multivariate and covariate-informed forecasting, while Toto 2.0 (Khwaja et al., 2026) combines temporal and cross-variable attention with scalable training. TiRex-2 (Podest et al., 2026) supports multivariate forecasting and efficient streaming inference through a recurrent architecture. Falcon-2 (Xia et al., 2026) improves pretraining through controlled data sampling and varying context lengths and prediction horizons.

Recent efforts also extend foundation models to multimodal settings. ChatTime (Wang et al., 2025a) adapts a pretrained LLM to jointly process time series and text. Aurora (Wu et al., 2025) combines modality-guided attention with prototype-guided flow matching for multimodal probabilistic forecasting. HORAI (Chen et al., 2026) integrates time series, derived text and images, and external news through frequency-enhanced multimodal pretraining. MM-Time focuses on event-conditioned forecasting by adapting pretrained numerical TSFMs through parameter-efficient multimodal continued pretraining. It maps event descriptions to shared effect representations and uses explicit event intervals to guide their fusion with time series representations.

2.2 Adaptation for Time Series Models

Existing adaptation methods mainly improve forecasting on downstream datasets through channel modeling or covariate integration. The correlation-aware CoRA (Cheng et al., 2026) captures cross-channel dependencies through lightweight adapters, while ChronosX (Arango et al., 2025) incorporates past and future covariates into pretrained forecasting models. TaTS (Li et al., 2025) represents timestamp-aligned texts as auxiliary variables, and UniCA (Han et al., 2025) converts heterogeneous inputs, including text and images, into continuous covariate sequences for attention-based fusion. The covariate-aware CoRA (Qin et al., 2025) uses Granger Causality Embedding to weight multimodal covariates and incorporates them through zero-initialized condition-injection modules.

While these methods primarily focus on channel or covariate adaptation on target datasets, MM-Time aims to learn transferable event-conditioned forecasting capabilities through parameter-efficient multimodal continued pretraining. It explicitly addresses the semantic–effect gap and event–forecast temporal alignment by learning shared predictive effects across event descriptions and modeling the relationship between event intervals and forecast positions.

3 Method

3.1 Problem Formulation

Given historical observations $\mathbf{X} \in \mathbb{R}^{L\times C}$, where L is the input length and C is the number of channels, we aim to predict the next F time steps $\mathbf{Y} \in \mathbb{R}^{F\times C}$ using both the historical observations and available event information. Let t_0 denote the forecast origin. Each event is represented as $\mathbf{e}_i = (d_i, a_i, s_i, u_i)$, where d_i is its textual description, a_i is the time when the event information becomes available, and $[s_i, u_i]$ is its time interval. The available event set is defined as

$$\mathcal{E}_{t_0} = \{\mathbf{e}_i \mid a_i \le t_0\}. \tag{1}$$

The availability time determines whether an event can be used for forecasting, while the event interval provides temporal information for modeling its effects. All event attributes used by the model must be available at t_0, even if the event interval extends into the future.

Starting from a pretrained time series foundation model (TSFM) with parameters θ, we introduce event-conditioning modules and parameter-efficient adapters with trainable parameters ϕ. Through multimodal continued pretraining, we model

$$p_{\theta,\phi}\left(\mathbf{Y}_{t_0+1:t_0+F} \mid \mathbf{X}_{t_0-L+1:t_0}, \mathcal{E}_{t_0}\right). \tag{2}$$

We keep the original backbone parameters θ fixed and optimize ϕ to incorporate event information into the forecasting process.

3.2 Framework Overview

To incorporate event information into pretrained TSFMs, we propose MM-Time. As shown in Figure 1(a), the framework consists of three main components: the Event-Effect Prototype Router, Event Interval Rotary-Bias Attention, and parameter-efficient adapters in the final backbone layers.

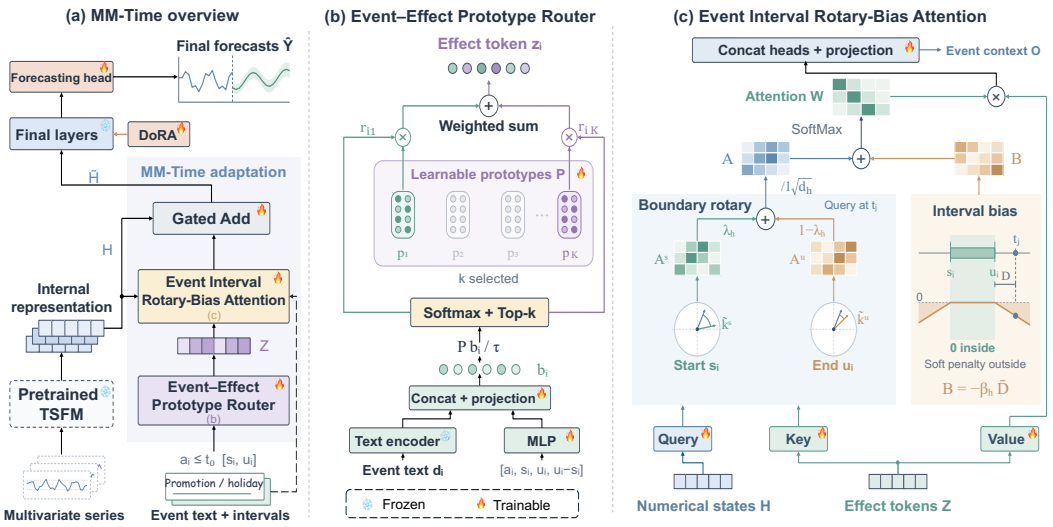


Figure 1: Overview of MM-Time. Event descriptions and temporal features are mapped to effect representations through a shared prototype dictionary. Interval-aware cross-attention combines these representations with numerical hidden states, and adapters in the final backbone layers support event-conditioned forecasting.

The **Event-Effect Prototype Router** (Section 3.3) maps event descriptions and temporal features to sparse combinations of learnable effect prototypes. These shared prototypes capture common effect patterns across events. Based on the resulting event representations, **Event Interval Rotary-Bias Attention** (Section 3.4) models their relationships with numerical hidden states through both representation similarity and relative time. It combines rotary encoding at the two event boundaries with a distance-based attention bias. The attention output is added to the backbone hidden states through a gated residual connection:

$$\widetilde{\mathbf{H}} = \mathbf{H} + \mathbf{g} \odot \mathrm{IntervalAttn}(\mathbf{H}, \mathbf{Z}, \mathcal{T}), \tag{3}$$

where $\mathbf{H}$ denotes the numerical hidden states at the fusion layer, $\mathbf{Z}$ denotes the event representations, $\mathcal{T}$ contains the event intervals, and $\mathbf{g}$ is a learnable gate that controls the contribution of event information. Event fusion is applied at an intermediate or late backbone layer. Finally, **parameter-efficient adapters** (Section 3.5) in the subsequent layers adapt the combined representations for prediction using the original forecasting head. We jointly train the event-conditioning modules and adapters while keeping the original backbone weights fixed.

3.3 Event-Effect Prototype Router

Event descriptions mainly convey semantic information, whereas forecasting requires modeling their effects on numerical observations. Events with similar descriptions may have different effects, while events with different descriptions may share common effect patterns. To bridge this gap, we introduce a learnable dictionary of effect prototypes. Each prototype represents a latent effect pattern, and each event is represented by a sparse weighted combination of prototypes. This allows events with similar effects to share representations even when their descriptions differ.

Event encoding. As shown in Figure 1(b), for each event $\mathbf{e}_i = (d_i, a_i, s_i, u_i)$, we first encode its textual description and temporal features. The two outputs are concatenated and projected into a shared feature space:

$$\mathbf{b}_i = \mathbf{W}_e \left[\mathrm{TextEncoder}(d_i)\,;\, \mathrm{MLP}([a_i, s_i, u_i, u_i - s_i])\right], \tag{4}$$

where $\mathbf{b}_i \in \mathbb{R}^d$ is the event embedding and $\mathbf{W}_e$ is a learnable projection matrix. We use a frozen Qwen3.5-4B (Team, 2026) as the text encoder. The temporal features describe the availability time, interval boundaries, and event duration.

Prototype routing. We define a learnable prototype dictionary $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]^\top \in \mathbb{R}^{K \times d}$, where K is the number of prototypes. Based on the similarity between an event embedding and the prototypes, the router computes assignment weights and selects the top-k prototypes:

$$\mathbf{r}_i = \mathrm{TopK}\left(\mathrm{softmax}\left(\frac{\mathbf{P}\mathbf{b}_i}{\tau}\right), k\right), \qquad \mathbf{z}_i = \sum_{j=1}^{K} r_{i,j}\mathbf{p}_j, \tag{5}$$

where τ is a temperature parameter. The TopK operation retains the k largest weights and sets the remaining weights to zero. The weighted sum $\mathbf{z}_i$ forms the event-specific effect representation. During training, we use a Gumbel-Softmax relaxation for the routing step. For M available events, we stack their representations as $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_M]^\top \in \mathbb{R}^{M \times d}$ for the subsequent attention module.

Prototype regularization. The forecasting loss jointly supervises the event encoder, router, and prototype dictionary. However, routing may repeatedly select only a few prototypes, leaving the others insufficiently trained. We therefore introduce a load-balancing loss (Shazeer et al., 2017; Chen et al., 2024) to encourage more balanced prototype usage. Let

$$f_j = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}[r_{i,j} > 0], \qquad \bar{r}_j = \frac{1}{M}\sum_{i=1}^{M} r_{i,j} \tag{6}$$

denote the selection frequency and average routing weight of prototype j, respectively. We also regularize the prototype similarities to reduce redundancy. The two objectives are defined as

$$\mathcal{L}_{\mathrm{load}} = K\sum_{j=1}^{K} f_j \bar{r}_j, \quad \mathcal{L}_{\mathrm{ortho}} = \left\| \frac{\mathbf{P}\mathbf{P}^\top}{\|\mathbf{P}\|_F^2} - \mathbf{I}_K \right\|_F^2, \tag{7}$$

where $\mathbb{I}[\cdot]$ is the indicator function, $\mathbf{I}_K$ is the identity matrix, and $\|\cdot\|_F$ denotes the Frobenius norm.

3.4 Event Interval Rotary-Bias Attention

The Prototype Router provides effect representations for the available events. To use these representations for forecasting, the model also needs to account for when each event occurs. An event may affect the time series over an interval, and its influence may extend beyond the interval boundaries. We therefore propose Event Interval Rotary-Bias Attention, which combines rotary encoding of both boundaries with an interval locality bias.

Boundary-aware rotary attention. Let $\mathbf{h}_j$ be a numerical hidden state and t_j its associated time. For attention head h, we obtain the query from $\mathbf{h}_j$ and the key and value from event representation $\mathbf{z}_i$ through learnable projections:

$$\mathbf{q}_j^{(h)} = \mathbf{W}_Q^{(h)}\mathbf{h}_j, \qquad \mathbf{k}_i^{(h)} = \mathbf{W}_K^{(h)}\mathbf{z}_i, \qquad \mathbf{v}_i^{(h)} = \mathbf{W}_V^{(h)}\mathbf{z}_i. \tag{8}$$

We apply rotary encoding to the query using t_j, and encode the event key separately using the interval start s_i and end u_i:

$$\widetilde{\mathbf{q}}_j^{(h)} = R(t_j)\mathbf{q}_j^{(h)}, \quad \widetilde{\mathbf{k}}_i^{s,(h)} = R(s_i)\mathbf{k}_i^{(h)}, \quad \widetilde{\mathbf{k}}_i^{u,(h)} = R(u_i)\mathbf{k}_i^{(h)}, \tag{9}$$

where $R(\cdot)$ is the rotary position transformation. All time coordinates use the same physical time unit so that temporal distances are comparable across series with different sampling frequencies. The two query–key scores capture the relative time to the event start and end. We combine them using a learnable weight for each head:

$$A_{ji}^{(h)} = \frac{\lambda_h (\widetilde{\mathbf{q}}_j^{(h)})^\top \widetilde{\mathbf{k}}_i^{s,(h)} + (1-\lambda_h)(\widetilde{\mathbf{q}}_j^{(h)})^\top \widetilde{\mathbf{k}}_i^{u,(h)}}{\sqrt{d_h}}, \qquad \lambda_h = \sigma(\gamma_h), \tag{10}$$

where d_h is the head dimension, γ_h is a learnable scalar, and $\sigma(\cdot)$ is the sigmoid function. Different heads can place different weights on the start and end boundaries, capturing their respective relationships with the numerical hidden states.

Interval locality bias. Rotary encoding models relative time to the boundaries, but does not directly distinguish positions inside and outside an event interval. We introduce an additional bias based on the distance from t_j to $[s_i, u_i]$:

$$D(t_j, [s_i, u_i]) = \max(s_i - t_j, 0) + \max(t_j - u_i, 0). \tag{11}$$

This distance is zero inside the interval and increases linearly with the distance to the nearest boundary outside it. Let $\overline{D}(t_j, [s_i, u_i])$ denote the normalized distance. The bias for head h is

$$B_{ji}^{(h)} = -\beta_h \overline{D}(t_j, [s_i, u_i]), \qquad \beta_h \geq 0, \tag{12}$$

where the learnable parameter β_h controls the strength of the distance penalty. A larger β_h gives greater preference to events whose intervals are closer to the query time. Since this bias does not mask events outside their intervals, the model can still attend to them when their representations are relevant.

Event fusion. We augment event attention with a null slot whose attention logit and value are fixed to zero. After adding the interval bias, we normalize over both the available events and the null slot:

$$w_{ji}^{(h)} = \frac{\exp\left(A_{ji}^{(h)} + B_{ji}^{(h)}\right)}{1 + \sum_{k=1}^{M} \exp\left(A_{jk}^{(h)} + B_{jk}^{(h)}\right)}, \qquad o_j^{(h)} = \sum_{i=1}^{M} w_{ji}^{(h)} v_i^{(h)}. \tag{13}$$

The null slot allows the model to suppress event injection, so boundary-aware scores and interval locality can modulate the event contribution even when only one event is available. We concatenate the outputs of all heads and apply an output projection to match the dimension of the backbone hidden states. The resulting event information is incorporated through the gated residual connection in Equation 3. The two attention components work together: rotary encoding captures relative time to both event boundaries, while the interval bias guides attention toward events with nearby or overlapping intervals.

3.5 Parameter-Efficient Multimodal Continued Pretraining

Updating all backbone parameters on event-annotated data may lead to overfitting and affect the forecasting capability learned during pretraining. We adopt a parameter-efficient strategy to adapt the backbone to event information. Specifically, we initialize MM-Time from a pretrained TSFM checkpoint and apply DoRA (Liu et al., 2024a) to selected linear projections in the final backbone layers after event fusion. These adapters allow the subsequent layers to process the combined representations. We also fine-tune the pretrained forecasting head to map these representations to event-conditioned predictions. During multimodal continued pretraining, we jointly optimize the event-conditioning modules, DoRA adapters, and forecasting head, denoted by $\phi = \{\phi_{\text{event}}, \phi_{\text{DoRA}}, \phi_{\text{head}}\}$, while keeping the remaining backbone parameters fixed.

Forecasting objective. For each backbone, let $\mathcal{Q} \subset (0, 1)$ be the set of output quantile levels, and let $\widehat{y}_{h,c}^{(\tau)}$ denote the predicted τ-quantile for channel c at forecast step h. We use the quantile loss averaged over quantile levels and observed target positions:

$$\mathcal{L}_{\text{pred}} = \frac{1}{|\mathcal{Q}|\,|\Omega|} \sum_{\tau \in \mathcal{Q}} \sum_{(h,c) \in \Omega} \rho_\tau \left(y_{h,c} - \widehat{y}_{h,c}^{(\tau)} \right), \tag{14}$$

where $\Omega \subseteq \{1, \ldots, F\} \times \{1, \ldots, C\}$ contains the observed target positions, and $\rho_\tau(r) = \max\{\tau r, (\tau - 1)r\}$ is the pinball loss. This objective trains the event-conditioning modules and adapters through their contribution to the final forecasts.

Overall objective. We combine the forecasting loss with the two prototype regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{ortho}} \mathcal{L}_{\text{ortho}} + \lambda_{\text{load}} \mathcal{L}_{\text{load}}, \tag{15}$$

where λ_{ortho} and λ_{load} control the strength of prototype diversity and load-balancing regularization.

4 EXPERIMENT

Datasets. We use real-world business time series to evaluate forecasting under events whose effects span specific time intervals. We divide the data into a continued-pretraining corpus and downstream evaluation datasets, with no overlap between them. The continued-pretraining corpus contains approximately 1 billion time series data points. The downstream datasets cover two domains: E-commerce, comprising 24 forecasting scenarios at hourly and weekly frequencies, and Retail, comprising 7 forecasting scenarios at 15-minute and daily frequencies. Detailed descriptions of the datasets are provided in the Appendix A.

Baselines. We compare MM-Time with four groups of baselines: (i) *unimodal TSFMs*, including Chronos-2 (Ansari et al., 2025), TiRex-2 (Podest et al., 2026), Toto 2.0 (Khwaja et al., 2026), TimesFM 2.5 (Das et al., 2024), Falcon-2 (Xia et al., 2026), Timer-S1 (Liu et al., 2026b), Sundial (Liu et al., 2025c), and Moirai-2 (Liu et al., 2025a); (ii) *multimodal TSFMs*, including Aurora (Wu et al., 2025) and ChatTime (Wang et al., 2025a); (iii) *task-specific multimodal models*, including VoT (Wang et al., 2026), GPT4MTS (Jia et al., 2024), TaTS (Li et al., 2025), and TESS (Li et al., 2026); and (iv) *LLM-based time series models*, including Time-LLM (Jin et al., 2024) and TimeCMA (Liu et al., 2025b). MM-Time and all foundation model baselines are evaluated in a zero-shot setting, without training or fine-tuning on the downstream datasets. Task-specific multimodal and LLM-based time series models are trained in a full-shot setting on the downstream training split, with validation-based model selection. Specific descriptions of baselines are in Appendix B.

Settings. We perform multimodal continued pretraining for 100,000 iterations on two NVIDIA B200 GPUs, with a maximum context length of 2048 and a maximum prediction length of 96. For downstream evaluation, we specify three prediction horizons for each sampling frequency, corresponding to short-, medium-, and long-term forecasting. We evaluate performance using mean absolute scaled error (MASE), with scores aggregated geometrically at the term level. More specific settings are in Appendix C.

4.1 MAIN RESULTS

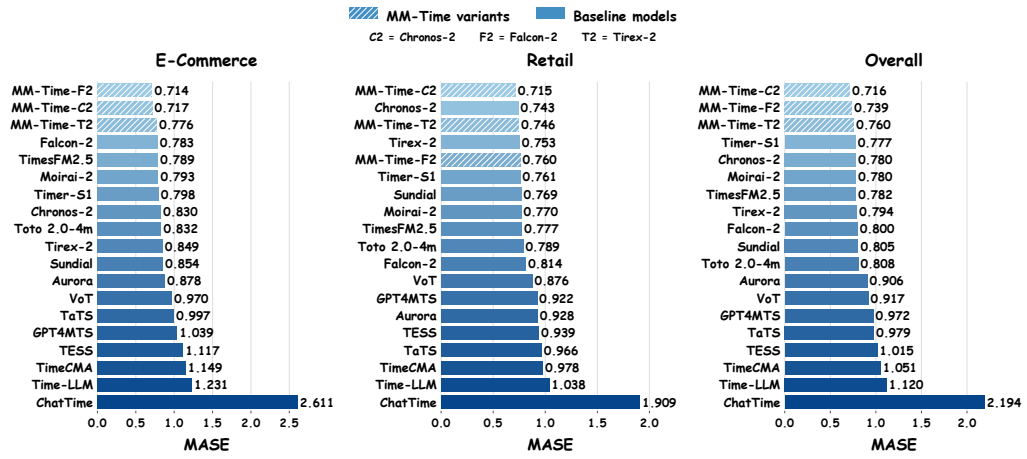


Figure 2: Forecasting results on E-Commerce, Retail, and Overall, measured by geometric-mean MASE (↓).

Figure 2 presents the forecasting results on E-Commerce and Retail, measured by the geometric mean of MASE across dataset–term settings. Overall aggregates all nine settings with equal weight. MM-Time-Falcon-2 achieves the best performance on E-Commerce with a MASE of 0.714, while MM-Time-Chronos-2 performs best on Retail and Overall, with MASE values of 0.715 and 0.716, respectively. All three MM-Time variants outperform their corresponding pretrained backbones on both domains and occupy the top three positions on E-Commerce and Overall. On E-Commerce, MM-Time-Chronos-2, MM-Time-Falcon-2, and MM-Time-Tirex-2 reduce MASE by 13.6%, 8.79%, and 8.56% relative to their respective backbones. Compared with the multimodal

foundation models Aurora and ChatTime, MM-Time-Chronos-2 reduces MASE by 18.35% and 72.53% on E-Commerce, and by 22.98% and 62.56% on Retail, respectively. All three MM-Time variants also outperform the task-specific multimodal models VoT, GPT4MTS, TaTS, and TESS on both domains. Relative to VoT, the strongest baseline in this category, MM-Time-Chronos-2 achieves MASE reductions of 26.08% on E-Commerce and 18.38% on Retail. These results further demonstrate that MM-Time provides an effective approach to enhancing time series foundation models for event-conditioned multimodal forecasting.

4.2 Ablation Study

To evaluate the contribution of each component in MM-Time, we compare the complete model with three ablated variants: (i) **w/o Prototype Router**, which bypasses prototype routing and directly uses the encoded event embeddings for fusion; (ii) **w/o Interval RoPE**, which removes boundary-aware rotary encoding from event attention while retaining the locality bias; (iii) **w/o Locality Bias**, which removes the interval distance penalty while retaining rotary encoding; The event encoder and gated residual connection are retained in all variants. Each variant follows the same continued-pretraining setting and downstream zero-shot evaluation protocol.

As shown in Table 1, MM-Time achieves the lowest aggregated MASE across both domains and Overall for both backbones. Removing the Prototype Router increases Overall MASE by 3.38% on Chronos-2 and 2.64% on Falcon-2, supporting the benefit of shared prototype-based event representations. Removing Interval RoPE increases Overall MASE by 2.50% and 1.08%, while removing the locality bias leads to increases of 3.30% and 2.35%, respectively. These results support incorporating both boundary-aware encoding and interval locality priors.

Table 1: Ablation study on Chronos-2 and Falcon-2. *MASE scores are geometric means over the corresponding dataset–term settings, with Overall aggregating all settings equally*. Proto., RoPE, and Bias denote the Prototype Router, Interval RoPE, and Interval locality bias, respectively.

Dataset	Chronos-2				Falcon-2			
	MM-Time	w/o Proto.	w/o RoPE	w/o Bias	**MM-Time**	w/o Proto.	w/o RoPE	w/o Bias
E-Commerce	**0.717**	0.745	0.744	0.752	**0.714**	0.730	0.717	0.724
Retail	**0.715**	0.736	0.726	0.729	**0.760**	0.783	0.772	0.784
Overall	**0.716**	0.740	0.734	0.739	**0.739**	0.759	0.747	0.757

4.3 Model Analysis

Effects of Continued Pretraining and Multimodal Conditioning. To distinguish the effects of continued pretraining from those of incorporating event information, we compare four configurations: (i) **Original TSFM**, the pretrained backbone without further training; (ii) **Numerical CPT**, which performs continued pretraining using numerical observations alone; (iii) **Vanilla Multimodal CPT**, which incorporates event descriptions and time series through standard cross-attention, without prototype routing or interval-aware attention; and (iv) **MM-Time**, which uses the complete event-conditioning framework. All configurations start from the same TSFM checkpoint.

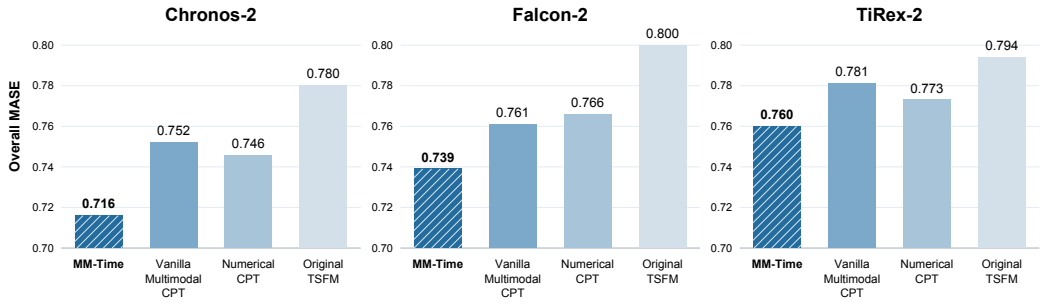


Figure 3: Forecasting results on the effects of continued pretraining and multimodal conditioning, measured by geometric-mean MASE (↓).

As illustrated in Figure 3, numerical CPT improves over Original TSFM across all three backbones, reducing Overall MASE by 2.64–4.38%, showing that continued pretraining alone contributes to the gains. Vanilla Multimodal CPT improves over Numerical CPT only on Falcon-2 and performs worse on Chronos-2 and TiRex-2, indicating that standard cross-attention does not consistently translate event information into better forecasts. MM-Time achieves the lowest Overall MASE on all three backbones. These comparisons support the additional benefits of the proposed event-conditioning framework beyond numerical continued pretraining and standard multimodal fusion.

Robustness to Event Information. We examine how MM-Time responds to changes in event information through three inference-time tests: (i) **Semantic Paraphrasing**, which rewrites event descriptions while preserving their meaning and temporal attributes; (ii) **Event-text Shuffling**, which shuffles descriptions among available events while keeping their temporal attributes fixed; and (iii) **Boundary Perturbation**, which adds noise to event start and end times while maintaining valid intervals. Model parameters and numerical inputs remain fixed throughout these tests. As shown in Figure 4(a), semantic paraphrasing increases Overall MASE by only 0.34%, whereas event-text shuffling and boundary perturbation increase it by 4.50% and 2.26%, respectively. These results suggest that MM-Time is relatively stable under meaning-preserving rewrites, while its forecasting accuracy benefits from correctly matched event descriptions and accurate temporal boundaries.

Comparison with Covariate Adaptation Methods. To compare different ways of incorporating event information under a common training protocol, we integrate UniCA (Han et al., 2025) and covariate-aware CoRA (Qin et al., 2025) into our continued-pretraining framework. Specifically, we replace MM-Time's event-conditioning modules with their respective covariate encoding and fusion mechanisms, yielding **UniCA-CPT** and **CoRA-CPT**. The variants share the same TSFM checkpoint, available event information, continued-pretraining data, and training schedule, while retaining the same DoRA and forecasting-head adaptation strategy. As shown in Figure 4(b), MM-Time achieves the lowest Overall MASE of 0.716, outperforming UniCA-CPT and CoRA-CPT. Under the matched training and evaluation protocol, these gains support the effectiveness of the proposed event-conditioning design compared with general covariate adaptation.

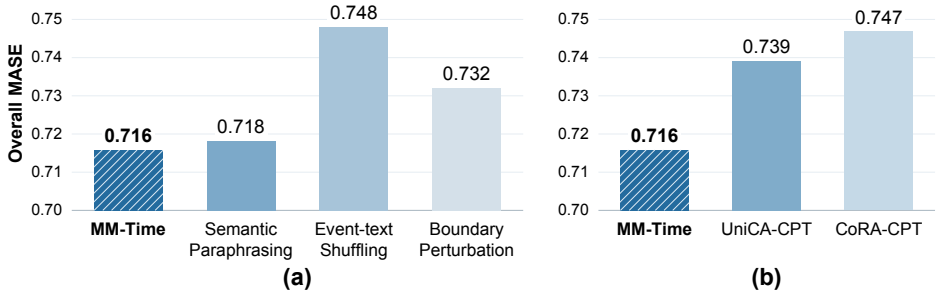


Figure 4: Overall MASE (↓): (a) robustness to event-information perturbations; (b) comparison with UniCA-CPT and CoRA-CPT.

Due to space limitations, we provide the sensitivity analysis of prototype configuration and qualitative analysis of event-prototype routing in Appendix E.1 and Appendix E.2.

5 Conclusion

We propose MM-Time, a parameter-efficient framework for adapting pretrained time series foundation models to event-conditioned forecasting. Its Event-Effect Prototype Router maps event descriptions and temporal attributes to sparse combinations of shared prototypes learned under forecasting supervision. Event Interval Rotary-Bias Attention combines boundary-aware rotary encoding with an interval-distance bias to guide the fusion of event information and numerical representations. Through gated residual fusion, DoRA adapters, and forecasting-head adaptation, MM-Time learns transferable event-conditioning capabilities via multimodal continued pretraining while preserving the remaining backbone parameters. Experiments across three backbones on e-commerce and retail datasets support its effectiveness for zero-shot forecasting.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and code development and debugging. All AI-assisted text and code were carefully reviewed and verified by the authors. The research questions, methodology, experimental design, analysis, and scientific claims were developed independently by the authors. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work focuses on methodological research for multimodal time series forecasting. Our experiments use existing benchmark datasets and pretraining corpora, without involving human subjects or the intentional collection of personally identifiable information. The proposed method is intended for general-purpose forecasting research, and we are not aware of any direct ethical risks arising from the methodology itself.

REPRODUCIBILITY STATEMENT

The architecture and training objectives of MM-Time are described in Section 3, with the parameter-efficient adaptation procedure detailed in Section 3.5. The experimental setup and evaluation protocol are presented in Section 4. Descriptions of the baseline models and illustrative examples of event perturbations are provided in Appendix B and Appendix D, respectively. The dataset, source code, and scripts for training and evaluation will be released.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda-Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *CoRR*, abs/2403.07815, 2024.

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

Sebastian Pineda Arango, Pedro Mercado, Shubham Kapoor, Abdul Fatir Ansari, Lorenzo Stella, Huibin Shen, Hugo Senetaire, Ali Caner Türkmen, Oleksandr Shchur, Danielle C. Maddix, Michael Bohlke-Schneider, Bernie Wang, and Syama Sundar Rangapuram. Chronosx: Adapting pretrained time series models with exogenous variables. In *AISTATS*, volume 258, pp. 2242–2250, 2025.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *ICLR*, 2024.

Peng Chen, Siyuan Wang, Shiyan Hu, Xingjian Wu, Yang Shu, Zhongwen Rao, Meng Wang, Yijie Li, Bin Yang, and Chenjuan Guo. Empowering time series analysis with large-scale multimodal pretraining. *CoRR*, abs/2602.05646, 2026.

Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Cora: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. *CoRR*, abs/2603.21828, 2026.

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. *CoRR*, abs/2505.14766, 2025.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *ICML*, volume 235, pp. 10148–10167, 2024.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *NeurIPS*, 2024.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 152139–152170, 2026.

Lu Han, Yu Liu, Qiwen Deng, Jian Jiang, Yinbo Sun, Zhe Yu, Binfeng Wang, Xingyu Lu, Lintao Ma, Han-Jia Ye, and De-Chuan Zhan. Unica: Adapting time series foundation model to general covariate-aware forecasting. *CoRR*, abs/2506.22039, 2025.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. GPT4MTS: prompt-based large language model for multimodal time-series forecasting. In *AAAI*, 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 *ICLR*, 2024.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, Chenghao Liu, Ameet Talwalkar, and David Asker. Toto 2.0: Time series forecasting enters the scaling era. *CoRR*, abs/2605.20119, 2026.

Lehui Li, Yuyao Wang, Jisheng Yan, Wei Zhang, Jinliang Deng, Haoliang Sun, Zhongyi Han, and Yongshun Gong. From text to forecasts: Bridging modality gap with temporal evolution semantic space. *arXiv preprint arXiv:2603.12664*, 2026.

Zihao Li, Xiao Lin, Zhining Liu, Jiaru Zou, Ziwei Wu, Lecheng Zheng, Dongqi Fu, Yada Zhu, Hendrik F. Hamann, Hanghang Tong, and Jingrui He. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative. *CoRR*, abs/2502.08942, 2025.

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. *CoRR*, abs/2511.11698, 2025a.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI*, volume 39, pp. 18780–18788, 2025b.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *ICML*, volume 235, pp. 32100–32121, 2024a.

Yiding Liu, Yifan Hu, Hongjie Xia, Peiyuan Liu, Hongzhou Chen, Xilin Dai, Zewei Dong, and Jiang-Ming Yang. Falcon-x: A time series foundation model for heterogeneous multivariate modeling. *CoRR*, abs/2605.27286, 2026a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *ICML*, volume 235, pp. 32369–32399, 2024b.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *ICML*, volume 267, 2025c.

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

Junkai Lu, Peng Chen, Xingjian Wu, Yang Shu, Chenjuan Guo, Christian S. Jensen, and Bin Yang. PATRA: pattern-aware alignment and balanced reasoning for time series question answering. *CoRR*, abs/2602.23161, 2026.

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

Guo Qin, Zhi Chen, Yong Liu, Zhiyuan Shi, Haixuan Liu, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Cora: Covariate-aware adaptation of time series foundation models. *CoRR*, abs/2510.12681, 2025.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *ICLR*, 2025.

Qwen Team. Qwen3.5-omni technical report. *CoRR*, abs/2604.15804, 2026.

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

Siyuan Wang, Peng Chen, Yihang Wang, Wanghui Qiu, Chenjuan Guo, Bin Yang, and Yang Shu. Unlocking the value of text: Event-driven reasoning and multi-level alignment for time series forecasting. *CoRR*, abs/2603.15452, 2026.

Yihang Wang, Yuying Qiu, Peng Chen, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Lightgts: A lightweight general time series forecasting model. In *ICML*, volume 267, 2025b.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *ICML*, volume 235, pp. 53140–53164, 2024.

Xingjian Wu, Jianxin Jin, Wanghui Qiu, Peng Chen, Yang Shu, Bin Yang, and Chenjuan Guo. Aurora: Towards universal generative multimodal time series forecasting. *CoRR*, abs/2509.22295, 2025.

Hongjie Xia, Yiding Liu, Yifan Hu, Peiyuan Liu, and Zewei Dong. Into the ORBIT for time series: Training regimes for foundation models. *CoRR*, abs/2608.13262, 2026.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *Proc. VLDB Endow.*, 18(8):2385–2398, 2025.

A DATASETS

A.1 MULTIMODAL CONTINUED-PRETRAINING CORPUS

Time series and event information. Our continued-pretraining corpus combines general data, spanning six frequencies from one minute to one week. Each series is paired with an entity description and time-stamped contextual records. Event records contain textual descriptions, availability timestamps, and optional start and end times. The timestamps control information access, while the intervals provide temporal priors for event alignment. For Chronos-2, the resulting pool contains 1,442 entity–frequency series and 415,438 indexed series–event associations (Table 2). The general corpus provides 53,142,850 and 2,619,416 candidate forecast origins. Training windows sampled from these origins may overlap.

Table 2: Continued-pretraining data used by the Chronos-2 configuration. Series are counted separately at each frequency. Event records denote indexed series–event associations before window-specific filtering. $H_{\max}$ is the maximum supervised prediction length.

Frequency	#Series	#Event records	$H_{\max}$
1 minute	62	17,057	96
15 minutes	115	32,073	96
30 minutes	127	34,911	96
1 hour	334	99,155	96
1 day	570	163,073	12
1 week	234	69,169	12
Total	1,442	415,438	–

Temporal partitioning. Continued pretraining uses numerical observations before January 1, 2026, 00:00 UTC. Benchmarks are excluded from the continued-pretraining corpus.

A.2 EVALUATION DATASETS

We use four datasets: hourly and weekly e-commerce series, and 15-minute and daily retail series. The numerical observations are accompanied by textual context, including holiday schedules and promotional events. Together, these datasets contain 61 entity–frequency series, 2,931,379 time points, and 6,777 evaluation windows. Table 3 reports their statistics and the short-, medium-, and long-horizon configurations used where available.

Table 3: Statistics and forecasting configurations of the evaluation datasets. H denotes the prediction length in time steps, and W denotes the number of evaluation windows across all eligible series. A dash indicates an unavailable forecasting term.

Dataset	Freq.	#Series	#Points	Short		Medium		Long	
				H	W	H	W	H	W
E-commerce	1H	24	1,108,637	24	3,334	168	460	336	218
E-commerce	1W	23	6,294	12	23	–	–	–	–
Retail	15Min	7	1,797,485	48	2,201	480	217	720	142
Retail	1D	7	18,963	7	147	30	35	–	–

B BASELINES

We compare MM-Time with 16 forecasting baselines, organized into the following four groups: unimodal TSFMs, multimodal TSFMs, task-specific multimodal models, and LLM-based time series models. The unadapted Chronos-2, Falcon-2, and TiRex-2 checkpoints also serve as paired backbone references for their respective MM-Time variants.

B.1 Unimodal time series foundation models

- Chronos-2 (Ansari et al., 2025) employs group attention to exchange information among related series, allowing a single pretrained model to perform zero-shot univariate, multivariate, and covariate-informed forecasting.
- TimesFM (Das et al., 2024) uses a decoder-only Transformer to forecast continuous time-series patches autoregressively. We evaluate version 2.5, which extends the context window and provides an optional quantile forecasting head.
- Timer-S1 (Liu et al., 2026b) combines sparse TimeMoE and TimeSTP blocks with serial-token prediction to improve long-horizon forecasting and reduce autoregressive error accumulation.
- Moirai-2 (Liu et al., 2025a) uses a decoder-only architecture with a single patch size, quantile outputs, and multi-token prediction for efficient probabilistic forecasting.
- Sundial (Liu et al., 2025c) models continuous next-patch distributions using the flow-matching-based TimeFlow loss, enabling zero-shot point and probabilistic forecasting without discrete tokenization or a predefined output distribution.
- Toto 2.0 (Khwaja et al., 2026) combines alternating temporal and variate attention with a quantile head for multivariate probabilistic forecasting.
- Falcon-2 (Xia et al., 2026) uses an encoder-only Transformer for multi-patch quantile forecasting, pretrained with ORBIT to control data sampling, context lengths, and prediction horizons.
- TiRex-2 (Podest et al., 2026) combines xLSTM with asymmetric grouped attention for multivariate and covariate-informed forecasting, while supporting efficient streaming inference.

B.2 Multimodal time series foundation models

- Aurora (Wu et al., 2025) integrates textual or visual context through modality-specific encoding, distillation, and modality-guided attention, and generates probabilistic forecasts via prototype-guided flow matching.
- ChatTime (Wang et al., 2025a) adapts LLaMA-2-7B by normalizing and discretizing time-series values into dedicated tokens in an expanded vocabulary. Continued pretraining enables numerical and textual inputs and outputs for zero-shot and context-guided forecasting.

B.3 Task-Specific multimodal models

- VoT (Wang et al., 2026) reasons over textual events using an LLM and retrieved historical examples, aligns endogenous textual and numerical representations, and adaptively fuses event-driven and numerical forecasts in the frequency domain.
- TaTS (Li et al., 2025) converts temporally paired text into auxiliary variables aligned with numerical observations, allowing existing forecasting backbones to incorporate text with minimal architectural changes.
- TESS (Li et al., 2026) uses an LLM to translate textual events into temporal primitives, including mean shift, volatility, shape, and lag, and incorporates them into forecasting through confidence-aware gating.
- GPT4MTS (Jia et al., 2024) embeds summarized text with BERT and numerical observations through reversible normalization and patching. Both modalities are processed by a pretrained GPT-2 backbone with most attention and feed-forward parameters frozen.

B.4 LLM-based Time Series Models

- TimeCMA (Liu et al., 2025b) aligns representations from a numerical encoder and an LLM-based prompt encoder, using only the final prompt token for downstream prediction to reduce computation.
- Time-LLM (Jin et al., 2024) reprograms numerical patches using pretrained word prototypes for a frozen LLM, while Prompt-as-Prefix provides task information and series statistics as natural-language guidance.

C Experimental Settings

C.1 Model Configuration

We instantiate MM-Time on pretrained Chronos-2, Falcon-2, and TiRex-2 checkpoints, retaining each backbone's native numerical preprocessing, patching mechanism, and forecasting head. Gated event fusion is applied after block 8 in Chronos-2 and TiRex-2 and after block 28 in Falcon-2. We apply DoRA to the final four blocks with rank 8 and scaling coefficient 16. The event-conditioning modules, DoRA parameters, and forecasting head are trainable; the remaining backbone weights are frozen.

The Event-Effect Prototype Router uses 32 learnable prototypes with top-4 routing. The event-context residual gate is initialized to 0.1, and text dropout is set to 0.1. Event descriptions are encoded into 2,560-dimensional representations using Qwen3.5-4B (Team, 2026). The embeddings are cached, and the text encoder remains frozen.

C.2 Continued Pretraining

Training inputs. The maximum history length is 2,048 time steps. We apply variable-length history augmentation, retaining the full context with probability 0.2 and otherwise sampling a shorter context with a four-patch minimum when sufficient history is available. Missing observations and padded positions are masked. The maximum supervised horizon is 96 steps for subdaily series and 12 steps for daily and weekly series.

For text conditioning, each sample uses at most 32 events whose information is available before the forecast origin and within the preceding 90 days. Events with complete start and end times are retained only if their intervals overlap the combined history and forecasting window.

Optimization. We use AdamW and the backbone's multi-quantile forecasting loss, computed over observed target positions. Routing load balancing, prototype diversity are regularized with weights 10^{-2}, 10^{-3}, respectively. The diversity weights are increased from zero to their target values over 4,000 steps following step 1,000.

Continued pretraining runs for 100,000 steps using two GPUs and a global batch size of 512. Learning rates increase linearly during the first 10% of training and then follow a cosine decay to 1% of their initial values.

C.3 Autoregressive Inference

Long-horizon inference. For horizons exceeding 96 steps, or 12 steps for daily and weekly series, we use block-autoregressive inference. Each block's median predictions are appended to the context, which is truncated to the configured length before predicting the next block. The event set remains fixed at the original forecast origin, while relative event-time coordinates are updated as the context advances. Block-wise quantile predictions are concatenated to obtain the requested horizon, without using future ground-truth observations.

C.4 Parameter Efficiency

We report the total and trainable parameter counts of MM-Time with three numerical backbones. During multimodal continued pretraining, we update the multimodal conditioning modules, the DoRA adapters in the final four backbone layers, and the original forecasting head, while keeping the remaining backbone parameters frozen. We calculate the trainable fraction relative to the total parameters of the adapted forecasting network. The frozen Qwen3.5-4B encoder used to obtain cached text embeddings is excluded from both counts.

Table 4 provides a module-wise breakdown. The Event-Effect Prototype Router includes the prototype embeddings, temporal encoders, routing network, and output projection. The gated text fusion comprises the sparse-event attention branch and its associated normalization, gating, and temporal bias parameters. DoRA counts include the low-rank factors and learnable magnitude vectors, excluding the frozen base weights. The forecasting head is an existing backbone component that is

unfrozen rather than newly introduced. Overall, MM-Time updates 10.08%, 4.25%, and 7.79% of the adapted forecasting networks for Chronos-2, Falcon-2, and TiRex-2.

Table 4: Parameter analysis of MM-Time. Counts are reported in millions (M). The first five components are trainable; the remaining backbone parameters are frozen. The external frozen text encoder is excluded.

Component	Chronos-2	Falcon-2	TiRex-2
Text projection	0.656	0.656	0.656
Event-Effect Prototype Router	3.261	5.568	1.610
Gated text fusion	4.734	8.409	2.107
DoRA adapters	0.679	0.778	0.657
Forecasting head	3.653	10.109	1.788
Trainable parameters	12.984	25.520	6.819
Frozen backbone (excluding head)	115.824	575.131	80.713
Total parameters	128.808	600.651	87.532
Trainable fraction (%)	10.08	4.25	7.79

D Illustrative Example of Event Perturbations

We illustrate the three event perturbations using a sample from the `E_commerce` dataset, with a forecast origin of January 1, 2026, and a 24-hour prediction horizon. The selected event context includes (A) New Year's Day, spanning January 1–3, and (B) Christmas and Boxing Day in the United Kingdom, spanning December 25–29. We denote their textual descriptions by d_A and d_B, respectively. Descriptions are translated or shortened for readability.

Table 5 shows the original and perturbed inputs. Semantic paraphrasing changes the wording of a description while preserving its meaning. Event-text shuffling swaps the event names and descriptions between the two event slots while retaining the temporal attributes of each slot. Boundary perturbation changes the start and end times of an event while preserving its description and information availability time. Each perturbation is applied independently to the original sample after event selection; the selected event set, numerical inputs, and model parameters remain fixed.

Table 5: Example transformations for the three event perturbations. Each row represents a separate modification of the same original sample. Paraphrasing and text shuffling preserve all temporal attributes; boundary perturbation preserves the event text and information availability time. All other events are unchanged.

Perturbation	**Original input**	**Perturbed input**
Semantic paraphrasing	**Event A:** "The New Year's Day holiday runs from January 1 to January 3, lasting three days."	**Event A:** "The three-day New Year's Day break begins on January 1 and ends on January 3."
Event-text shuffling	**Slot A:** d_A (New Year's Day) **Slot B:** d_B (Christmas and Boxing Day)	**Slot A:** d_B (Christmas and Boxing Day) **Slot B:** d_A (New Year's Day)
Boundary perturbation	**Event A:** Start: Jan. 1, 00:00 End: Jan. 3, 23:00	**Event A:** Start: Jan. 1, 23:00 End: Jan. 4, 23:00

E Model Analysis

E.1 Sensitivity Analysis

We investigate how prototype capacity and routing sparsity affect forecasting performance. To assess the effect of pool size, we vary the number of prototypes $K \in \{8, 16, 32, 64, 128\}$ while fixing

the number selected per event at $k = 4$. To assess routing sparsity, we fix $K = 32$ and vary $k \in \{2, 4, 8, 16\}$. Together, these sweeps comprise eight configurations, including the shared default $(K, k) = (32, 4)$. For each configuration, we restart multimodal continued pretraining from the same pretrained backbone checkpoint. Data splits, training budgets, optimization settings, and regularization coefficients are held fixed.

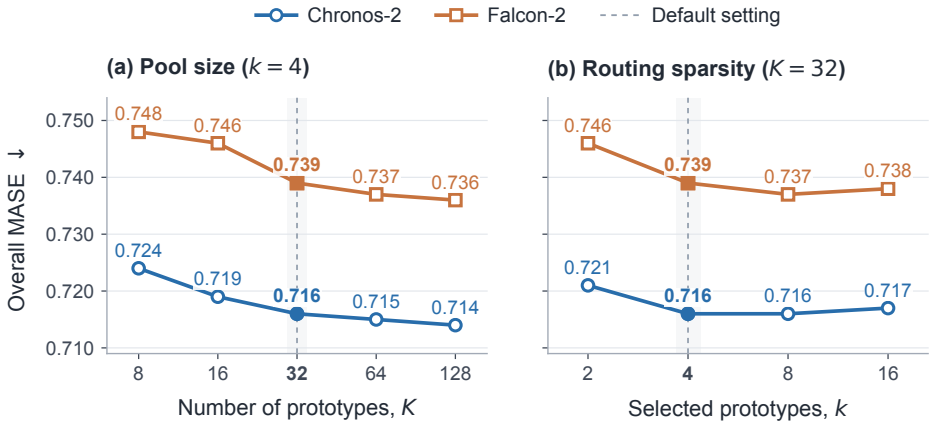


Figure 5: Sensitivity analysis results of the Prototypes.

Results. Both backbones benefit from increasing the prototype pool, although the gains diminish beyond $K = 32$. With $k = 4$, increasing K from 8 to 32 reduces MASE from 0.724 to 0.716 for Chronos-2 and from 0.748 to 0.739 for Falcon-2. Further increasing K to 128 yields smaller improvements, reaching 0.714 and 0.736, respectively. For routing sparsity, increasing k from 2 to 4 improves both backbones. Chronos-2 achieves the same reported MASE of 0.716 at $k = 4$ and $k = 8$, while Falcon-2 improves slightly from 0.739 to 0.737. Increasing k to 16 provides no further improvement for either backbone. These results suggest that larger prototype pools offer diminishing returns, while selecting a moderate number of prototypes is sufficient to achieve competitive performance.

E.2 Qualitative Analysis of Event–Prototype Routing

We study three pairs of promotional events from the E-commerce dataset using MM-Time-Chronos-2, with $K = 32$ prototypes and top-$k = 4$ selection. Model parameters remain fixed during the analysis. Each sample contains a 24-hour forecast window within its event interval. For each pair, we show the event texts, observed time series, baseline residuals, and prototype weights. Observed values are standardized using the historical mean and standard deviation. Baseline residuals are the differences between observed values and the original Chronos-2 median forecasts, divided by the historical standard deviation. We measure routing similarity using cosine similarity between the full 32-dimensional prototype-weight vectors.

Results. Figure 6 compares season-end savings with a Valentine gift campaign. Although the promotional themes differ, their observed trajectories follow similar patterns ($r = 0.953$). The events share three of four selected prototypes, with routing similarity 0.811.

Figure 7 compares tiered discounts with free shipping and an additional percentage discount. The two offers have similar baseline residuals and share three selected prototypes, with routing similarity 0.820. Their routing weights and fourth prototype differ, showing that shared prototypes can be combined differently for individual events.

Figure 8 compares Valentine and anniversary campaigns from different entities. Both events select the same four prototypes with similar weights, giving a routing similarity of 0.996.

Together, these cases illustrate how events with different descriptions can share similar prototype combinations when their associated time series show similar changes. This observation is consistent

with our goal of learning reusable, forecast-relevant event representations. Shared prototypes provide a common basis for representing different events, while differences in prototype selection and routing weights allow event-specific variation.

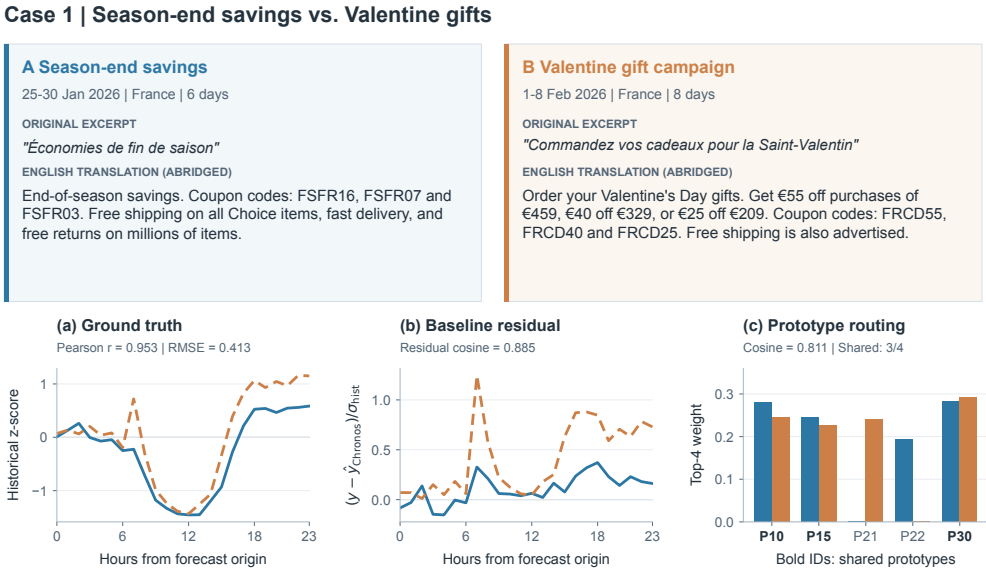


Figure 6: **Qualitative analysis of event–prototype routing: Case 1.**



Figure 7: **Qualitative analysis of event–prototype routing: Case 2.**

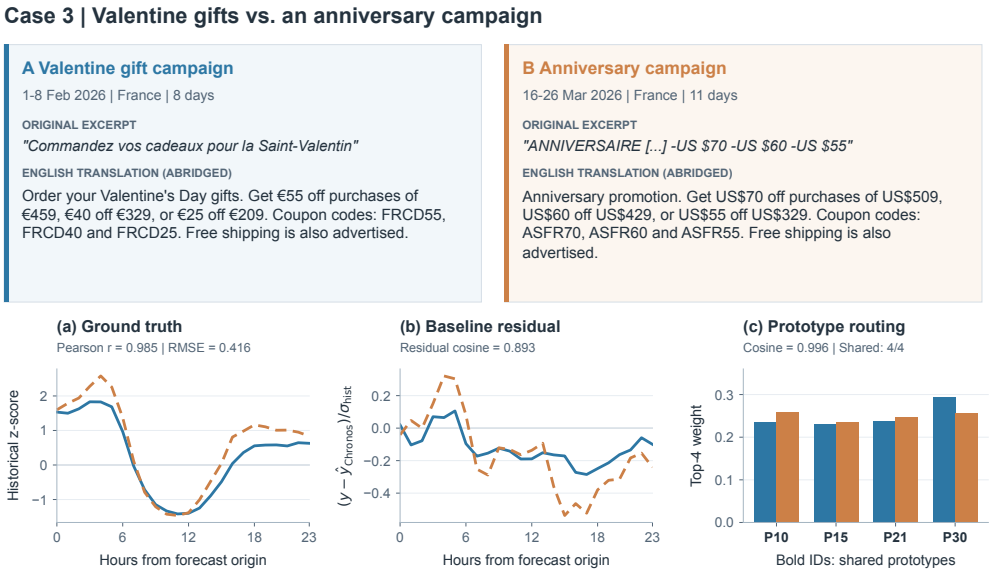


Figure 8: **Qualitative analysis of event–prototype routing: Case 3.**