TSFM Meets LLM: Context as Covariates

Hemanth Ram* **Sai Shankar Narasimhan*** **Shubhankar Agarwal** **Sandeep Chinchali**

The University of Texas at Austin, Austin, TX, 78712.
{hemanthramgk,nsaishankar,somi.agarwal,sandeepc}@utexas.edu

Abstract

Textual information is ubiquitous in real-world time series datasets, appearing as news, reports, and annotations that hold key predictive signals not deducible from numerical history alone. Large Language Models (LLMs) interpret text but lack numerical precision, while Time Series Foundation Models (TSFMs) are quantitatively accurate but cannot reason about the effects of text. Moreover, the lack of large-scale paired multimodal datasets limits progress in bridging this gap. In this work, we address this challenge by providing a scalable approach for generating realistic multimodal datasets by annotating real-world time series with LLMs. Building on this, we propose **Co**ntext as **Co**variates (CoCo), a forecasting framework in which an LLM distills its textual reasoning into numerically grounded forecast and confidence covariates that guide a TSFM backbone, thereby exploiting the strengths of both LLMs and TSFMs. We align the two models in stages, first fine-tuning the LLM via Group Relative Policy Optimization (GRPO) with a reward tied to covariate-induced improvement in forecasting performance, and then fine-tuning the TSFM on covariates generated by the aligned LLM. Our experiments show that models trained on our synthetic multimodal corpus generalize to real benchmarks spanning finance, economics, weather, traffic, and security. Notably, CoCo with `Qwen3-4B-Instruct-2507` as the LLM and Chronos2 as the TSFM achieves up to 12% MAE improvement over state-of-the-art TSFMs and LLM forecasters, such as GPT-4o. Code[1], checkpoints, and the evaluation datasets[2] are publicly available.

1 Introduction

"Forecast Apple's stock price if revenue from iPhone sales is 15% higher than the target"
"Forecast crude oil prices given the recent tariffs on the OPEC countries"

These contextual forecasting examples across domains show that the textual context is deeply intertwined with real-world time series and can provide signals about future changes that are not captured by past numerical values alone (check Figure 1). For example, in the finance domain, analyst notes, government policies, and even social media sentiments influence market dynamics, and are commonly used to predict stock price movements [1–4]. Similarly, in consumer markets, online reviews and social media discussions are used to forecast future product demand [5, 6].

Despite the availability of textual information, the state-of-the-art (SOTA) univariate (TimesFM [7]) and multivariate (Chronos2 [8]) time series foundation models (TSFMs) operate solely on unimodal numerical inputs. Figure 5 shows that incorporating text embeddings into SOTA multivariate

*Equal Contribution
[1]https://github.com/UTAustin-SwarmLab/CoCo
[2]https://huggingface.co/collections/SWARM-Lab/coco

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

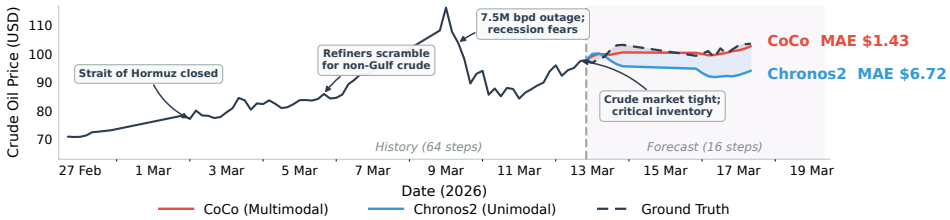


Figure 1: **Effective use of context is essential for accurate time series forecasting.** In crude oil price forecasting, textual context (e.g., recession fears on March 9) aligns with sudden price shifts. Our approach, **Co**ntext as **Co**variates (CoCo), leverages this context to produce accurate forecasts, while unimodal models such as Chronos2 fail to capture context-induced trends.

forecasting models does not necessarily yield multimodal gains, even when the text clearly contains useful information. Given the substantive progress in the understanding and reasoning capabilities of large language models (LLMs) [9, 10], we address the following key question in this work: *How can we leverage LLMs to process textual annotations or context and improve the performance of TSFMs?* Current efforts to build such context-aware TSFMs struggle with two major barriers:

- **Insufficient Multi-Domain Multimodal Training Data:** Learning complex and lagged relationships between text and future numerical values requires large volumes of aligned multimodal data from different domains. However, existing multi-domain datasets such as TimeMMD [11] and Context-Is-Key [12] contain only a few thousand aligned samples. Large-scale datasets [13] are heavily restricted to a single sub-domain, limiting cross-domain generalization.
- **LLM-Centric Multimodal Forecasting approaches neglect the predictive power of TSFMs:** Numerous prior works [14–20] serialize time series as tokens and rely primarily on LLM prediction. While this enables the use of textual context, it fails to leverage the forecasting inductive biases and uncertainty modeling strengths of dedicated TSFMs.

Motivated by these gaps, we make three contributions.

- **Scalable Synthetic Multimodal Data.** We introduce a pipeline that annotates real-world time series with LLM-generated text, producing large-scale, text-aligned datasets with varied lagged correlations. We empirically show that augmenting scarce and often domain-restricted real multimodal data with synthetic annotations yields strong zero-shot forecasting performance.
- **CoCo Architecture and Staged Fine-tuning.** We propose CoCo (**Co**ntext as **Co**variates), where an LLM distills text into forecast and confidence covariates that guide a Chronos2 backbone. The LLM is fine-tuned via Group Relative Policy Optimization (GRPO) using a reward tied to the forecasting performance, after which Chronos2 is fine-tuned with the LLM-generated covariates.
- **Multimodal Forecasting Gains.** CoCo outperforms SOTA TSFMs and LLMs across 30 datasets and 95 settings (Figure 5) spanning finance, weather, macroeconomics, traffic, and security domains, achieving up to 70% improvement over GPT-4o in traffic forecasting (Figure 2).

2 Related Works

Multimodal datasets for text-aided forecasting. Prior works largely study *text-aided* time series prediction, where auxiliary textual context improves forecasts across domains [21, 22, 16, 23–25]. However, the nature of text supervision varies significantly across domains. For example, finance offers abundant dynamic or *time-varying* annotations (*e.g.,* news/social media/search trends) enabling large-scale datasets for stock price and movement predictions [13, 21, 22, 16, 23–25]. On the other hand, healthcare datasets typically provide *static/global* patient information in the form of medical records aligned with physiological signals and are commonly used for classification or anomaly detection [26, 27]. Other settings (*e.g.,* weather, system monitoring, e-commerce) similarly incorporate static descriptors or dynamic textual context, but remain largely domain-specific [28–31].

While TimeMMD [11] and Context-Is-Key [12] provide a collection of multimodal datasets across domains, they are primarily used for benchmarking purposes, and not large-scale training, due to their small sizes. To address these gaps, **we focus on leveraging LLMs to generate synthetic multimodal**

datasets with realistic time-varying textual annotations aligned to real-world time series. This scales the volume of multimodal datasets, enabling large-scale training.

Multimodal forecasting approaches. Prior methods can be broadly grouped into *zero-shot (training-free)* and *dataset-specific, training-based* approaches. Zero-shot methods primarily use LLMs by encoding time series and other modalities as textual prompts, enabling LLMs to reason over temporal patterns and contextual information [32, 33, 16, 34, 15]. Such approaches offer flexibility across domains, but Park et al. [14] show that they are often sensitive to noise and typically underperform with respect to lightweight or domain-specific forecasting models [35] on standard benchmarks.

Training-based approaches learn joint representations of time series and text by fine-tuning LLMs [36–38, 18], aligning text and time series embeddings [39–42], or employing fusion architectures [43, 37, 44–46]. Several works explore vision-based [47–49] or vision–language-based [50] representations by encoding time series as images or plots, or closed-loop reasoning frameworks that iteratively align text, predictions, and explanations [20]. These methods are either domain-specific or require per-dataset training.

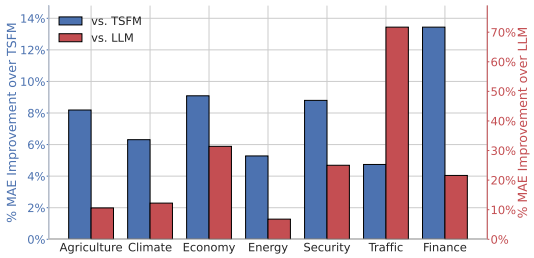


Figure 2: **CoCo outperforms SOTA TSFMs (Chronos2) and closed-source LLMs (GPT-4o) across datasets from diverse domains.** CoCo effectively utilizes LLMs to distill textual annotations into usable forecast covariates that guide powerful TSFMs to generate accurate forecasts.

Overall, previous works primarily (i) focus on LLMs as the key forecasters, (ii) rely on dataset-specific multimodal training, and (iii) do not exploit SOTA TSFMs [8, 7, 51]. In contrast, **we develop a zero-shot, domain-agnostic setup** that combines the strengths of LLMs and TSFMs to generate context-aware forecasts. We refer the readers to Appendix A for a detailed discussion.

3 Problem Formulation

Unimodal time series forecasting models take historical numerical series $X^H \in \mathcal{X}^H = \mathbb{R}^l$, where l is the length of the context window, as input and generate the forecast $X^F \in \mathcal{X}^F = \mathbb{R}^h$, where h is the horizon or forecast length. Typically, the sample at timestamp t is denoted as (X_t^H, X_t^F), where $X_t^H = [x_{t-l+1}, x_{t-l+2}, \ldots, x_t] \in \mathcal{X}^H$ and $X_t^F = [x_{t+1}, x_{t+2}, \ldots, x_{t+h}] \in \mathcal{X}^F$. Thus, the unimodal forecasting model parameterized by θ is given by $f_\theta^{\text{uni}} : \mathcal{X}^H \to \mathcal{X}^F$.

In the multimodal setting, the textual annotations, denoted by $S \in \mathcal{S} = \mathbb{N}^{l \times N_{\max}^{\text{token}}}$, are available for each timestamp in the context for the most general setting, with up to $N_{\max}^{\text{token}}$ tokens in each annotation. As with the unimodal case, the sample at timestamp t in a dataset is denoted as (X_t^H, S_t, X_t^F), where $S_t = [s_{t-l+1}, s_{t-l+2}, \ldots, s_t] \in \mathcal{S}$. Note that at any given timestamp t, s_t can contain a summary of the past numerical values and the predictive cues to accurately forecast the future values. To this end, the multimodal forecasting model, parameterized by θ, is denoted as $f_\theta^{\text{multi}} : \mathcal{X}^H \times \mathcal{S} \to \mathcal{X}^F$.

Problem Statement: We aim to develop a multimodal forecasting model f_θ^{multi} such that the inclusion of textual context S considerably improves the forecasting accuracy compared to the unimodal approach f_θ^{uni} during zero-shot inference. More formally, given a dataset $D = \{(X_t^H, S_t, X_t^F)\}_{t=1}^{t=N}$, where N is the total number of samples, and a metric $\mathcal{M}(\hat{X}^F, X^F)$ that evaluates the error in the forecast $\hat{X}^F$ relative to the ground truth X^F, we aim to develop f_θ^{multi} such that $\mathcal{M}(f_\theta^{\text{multi}}(X^H, S), X^F) \leq \mathcal{M}(f_\theta^{\text{uni}}(X^H), X^F)$.

4 Synthetic Multimodal Dataset Generation

We propose a scalable approach to address the scarcity of multi-domain multimodal datasets by generating textual annotations for real-world time series using LLMs, motivated by the following **key intuition**: given numerical trajectories and domain descriptions, LLMs can produce semantically meaningful explanations for observed temporal trends (see Figure 11).

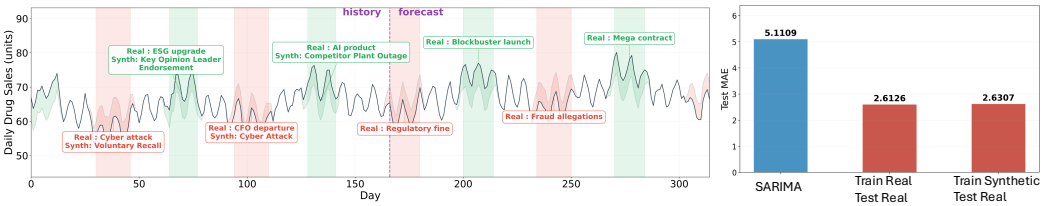


Figure 3: **Training on synthetic annotations yields real-world multimodal gains.** Using pharmaceutical sales data annotated with both real news and synthetic text, a SARIMA model with sentiment features achieves similar MAE when trained on synthetic versus real annotations, as both encode the same domain-agnostic forecast trends, thereby enabling **sim-to-real** transfer.

For each dataset, we randomly sample windows of fixed length and provide the numerical values, timestamps, and dataset description to an LLM (`Qwen3-30B-A3B-Instruct-2507` [9], referred to as Qwen3-30B). The model is prompted to generate context-rich annotations for each timestamp t describing plausible events or exogenous indicators explaining the observations up to t. Additionally, the LLM produces speculative descriptions of the future trajectory without generating explicit numerical forecasts. We added key indications to the prompt to ensure sufficient diversity in the generated annotations, and refer the readers to Appendix I.1 for the actual prompt input to the LLM.

To assess the realism of the generated annotations, we compare the cosine similarity between the normalized BERT [52] embeddings of the true future and the predicted forecast description, for real and synthetic cases. The significant overlap in resulting distributions (Figure 8) indicates synthetic annotations capture the forecast with levels of uncertainty similar to those observed in real context.

To this end, we make the following observation: **sim-to-real transfer**, *i.e.,* training on synthetic and testing on real-world text annotations, is feasible because both annotation types encode similar domain-agnostic descriptions of forecast trends (e.g., `argmax` in the second forecast timestamp) in domain-specific terms (e.g., due to tariffs, crude oil prices are expected to peak in the next two days). We validate this observation using a toy example (Figure 3). Consider the pharmaceutical sales of a company following a Seasonal Autoregressive Integrated Moving Average (SARIMA) process with shocks, explained by analyst notes. For each shock, an LLM generates a plausible domain-specific annotation. We fit SARIMA models with extracted sentiment features (scalar, domain-agnostic value ranging from -1 to 1) from both real and synthetic annotations and observe similar MAE on real annotations during the test period, demonstrating that synthetic text preserves forecast-relevant semantics useful for accurate prediction. A detailed description is provided in Appendix C.

To formalize this observation, we introduce a domain-agnostic contextual variable $Z \in \mathcal{S}$ (e.g., spikes or gradual trends). Let the history, forecast, and the forecast's domain-agnostic descriptions be sampled from a joint data distribution $p(X^H, Z, X^F)$, where the future X^F depends on history X^H and domain-agnostic context Z. The real annotations are naturally aligned with and describe the underlying future trends Z in domain-specific terms ($S^r \sim p_r(S^r \mid X^H, Z)$). The synthetic annotations are generated as $S^s \sim p_s(S^s \mid X^H, Z, X^F)$, reflecting the use of future information by the LLM. To this end, for a multimodal forecaster $f_\theta^{\text{multi}}(X^H, S) = h_\theta^{\text{multi}}(X^H, \phi(S))$, we define the real and synthetic population risks as

$$R^r_{\text{multi}}(\theta) = \mathbb{E}_{p,p_r}\left[\mathcal{M}\big(h_\theta^{\text{multi}}(X^H, \phi(S^r)), X^F\big)\right], \; R^s_{\text{multi}}(\theta) = \mathbb{E}_{p,p_s}\left[\mathcal{M}\big(h_\theta^{\text{multi}}(X^H, \phi(S^s)), X^F\big)\right].$$

Here, $\phi : \mathcal{S} \to \mathbb{R}^d$ is a text encoder with embedding dimension d, and $\mathbb{E}$ is the expectation operator. $\mathcal{M}(\hat{X}^F, X^F)$ can be the mean squared error (MSE) or the mean absolute error (MAE) between the ground truth and the forecast.

Assumption 4.1 (Approximate semantic representation)**.** There exists a semantic embedding $\psi(Z) \in \mathbb{R}^d$ of the latent context and representation error bounds $\varepsilon_r, \varepsilon_s \geq 0$ such that

$$\|\phi(S^r) - \psi(Z)\| \leq \varepsilon_r, \qquad \|\phi(S^s) - \psi(Z)\| \leq \varepsilon_s. \tag{1}$$

Assumption 4.2 (Lipschitz dependence on text representation)**.** There exists a constant $L > 0$ such that for all X^H, X^F, and $h_\theta^{\text{multi}} : \mathcal{X}^H \times \mathbb{R}^d \to \mathcal{X}^F$, and all $u, v \in \mathbb{R}^d$,

$$|\mathcal{M}(h_\theta^{\text{multi}}(X^H, u), X^F) - \mathcal{M}(h_\theta^{\text{multi}}(X^H, v), X^F)| \leq L\|u - v\|. \tag{2}$$

Here, $|\cdot|$ represents the absolute value and $\|\cdot\|$ represents the l_2 norm functions. **Assumption 4.1** states that both real (S^r) and synthetic (S^s) annotations lie close to the domain-agnostic forecast description Z in a latent space. Now, we provide empirical evidence to back this assumption. First, real-world text annotations (S^r) are typically speculations that encode the future (Z) and thus imply a small ε_r. To justify small ε_s, we refer to Figure 8. The strong overlap in cosine similarity distributions between the embeddings of forecast predictions derived from real ($\phi(S^r)$) and synthetic annotations ($\phi(S^s)$) with the true future descriptions ($\psi(Z)$), *i.e.,* $\phi(S^r)^T\psi(Z)$ and $\phi(S^s)^T\psi(Z)$, indicates that synthetic annotations recover the forecast nearly as well as real ones, suggesting $\varepsilon_s \approx \varepsilon_r$. Additionally, the generation process grounds synthetic text in numerical trajectories and domain descriptions. Here, $\phi(S)$ and $\psi(Z)$ correspond to normalized BERT embeddings of predicted and true domain-agnostic forecast descriptions, respectively.

Assumption 4.2 ensures that small perturbations in the text embedding produce limited changes in forecasting errors. Tables 5, 6, and 21 show that perturbing real-world textual annotations causes negligible variation in forecasting error when the perturbations largely preserve forecast-relevant semantics. This empirical result points towards local smoothness in the loss space with respect to small perturbations in text annotations for reasonably good forecasters.

Lemma 4.3 (Representation mismatch bound)**.** *Under **Assumptions 4.1** and **4.2**,*

$$|R^r_{\text{multi}}(\theta) - R^s_{\text{multi}}(\theta)| \leq L(\varepsilon_r + \varepsilon_s). \tag{3}$$

Theorem 4.4 (Synthetic-to-real transfer)**.** *Let* $\hat{\theta}_s \in \arg\min_{\theta\in\Theta} R^s_{\text{multi}}(\theta)$ *and* $\theta^\star_r \in \arg\min_{\theta\in\Theta} R^r_{\text{multi}}(\theta)$*. Under **Assumptions 4.1** and **4.2**,*

$$R^r_{\text{multi}}(\hat{\theta}_s) \leq R^r_{\text{multi}}(\theta^\star_r) + 2L(\varepsilon_r + \varepsilon_s). \tag{4}$$

Lemma 4.3 states that for a forecaster, if real and synthetic embeddings are close to the same latent semantic representation (small ε_r and ε_s), their risks remain close. **Theorem 4.4** states that a model trained with synthetic text achieves near-optimal performance under real text, with a bounded gap of $2L(\varepsilon_r + \varepsilon_s)$ relative to the best achievable real-text risk. In practice, this gap depends on (i) how effectively the encoder ϕ aligns real and synthetic text in a shared semantic space (whether $\varepsilon_r \simeq \varepsilon_s$), and (ii) how well synthetic annotations preserve the forecast-relevant semantics of Z (whether $\|\phi(S^s) - \psi(Z)\|$ is small). Our results in Section 6 demonstrate that models trained on synthetic text generalize effectively to real-world multimodal datasets. Proofs of **Lemma 4.3** and **Theorem 4.4** and a detailed discussion on the assumptions are provided in Appendix B.

Our formulation is closely related to domain adaptation and invariant representations [53], where domain generalization is achieved by aligning representations or assuming a shared latent variable. Similarly, sim-to-real transfer [54] relies on learning representations robust to shifts between synthetic and real data. We adapt these concepts to multimodal forecasting and realistic synthetic annotation generation for domains where there is a lack of real textual annotations, with valid assumptions. Without any loss of generality, the theory and algorithms developed in this work can be extended to the multivariate case.

5 Context as Covariates

We introduce CoCo (**Co**ntext as **Co**variates), a multimodal forecasting framework that combines LLMs' ability to interpret text with TSFMs' numerical precision (Figure 4). CoCo uses the LLM to generate a text-aware forecast capturing broad contextual trends and a per-step confidence estimate, which are provided as covariates to guide a multivariate TSFM (M-TSFM), such as Chronos2, toward precise numerical forecasts. We first describe the covariate construction.

LLM Forecast Covariate: Our main intuition is to model the forecast covariate as a noisy version of the next timestep. The forecast covariate, $M_t \in \mathbb{R}^{(l+h)}$, is constructed from the numerical history $X^H_t = [x_{t-l+1}, \ldots, x_t]$ and the text-aware LLM forecast $\theta_{\text{LLM}}(X^H_t, S_t) = [m_{t+1}, \ldots, m_{t+h}]$ as:

$$M_t = \big[x_{t-l+2} + \epsilon_2,\ x_{t-l+3} + \epsilon_3,\ \ldots,\ x_t + \epsilon_l,\ m_{t+1},\ m_{t+2},\ \ldots,\ m_{t+h},\ m_{t+h} + \epsilon_{\text{end}}\big], \tag{5}$$

where $\epsilon_i, \epsilon_{\text{end}} \sim \mathcal{N}(0, \sigma^2_\epsilon)$ are independent and identically distributed (i.i.d.) noise terms and we assign a very small value to σ^2_ϵ. In the history portion, the covariate at position i is a noisy version

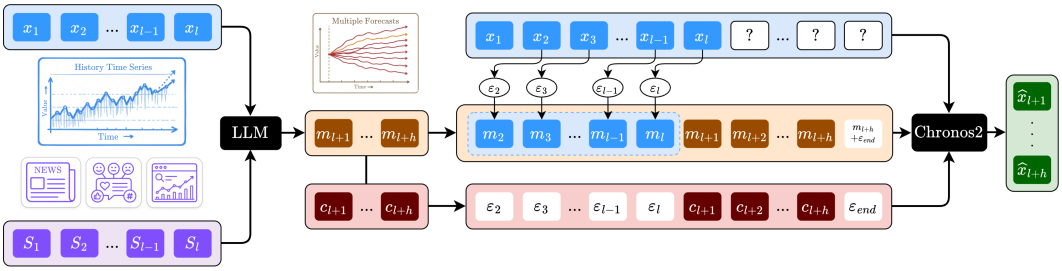


Figure 4: **Context as Covariates (CoCo) Description.** The LLM θ_{LLM} takes the numerical history X_t^H (blue) and textual context S_t (violet) as input and produces a forecast covariate M_t (Equation 5, brown) and a rollout-based confidence covariate C_t (Equation 6, maroon), which are concatenated with X_t^H and passed to Chronos2 for multivariate forecasting. Our **key technical intuition** is to utilize the confidence covariate (obtained from the per-step variance in the LLM's predictions) to teach Chronos2 when to trust the LLM and when not to trust it (low variance $\rightarrow$ trust the LLM, as text induces a more decisive forecast).

of the next observed value x_i. The future portion carries the LLM's median forecast directly. This construction allows M-TSFMs to learn through in-context learning that M_t provides a noisy version of the true forecast, enabling text-induced trends in $[m_{t+1}, \ldots, m_{t+h}]$ to be included in the final forecast. In other words, this setup allows M-TSFMs to interpret M_t as a noisy next-step predictor.

LLM Confidence Covariate: We model the confidence covariate to provide the M-TSFM with a per-step signal of how much to trust the LLM forecast. To measure confidence, we generate N_{rollouts} independent rollouts for the same multimodal input sample (X_t^H, S_t), where the i-th rollout is denoted as $[m_{t+1}^{(i)}, \ldots, m_{t+h}^{(i)}]$. The confidence at future timestep $t+j$ $(1 \le j \le h)$ is given by $c_{t+j} = \frac{\sigma^2\left(\{m_{t+j}^{(i)}\}_{i=1}^{N_{\text{rollouts}}}\right)}{\sigma^2(X_t^H)}$, where σ denotes the standard deviation operator and dividing by $\sigma^2(X_t^H)$ makes c_{t+j} scale invariant. To this end, we define the confidence covariate as

$$C_t = [\epsilon_2,\ \epsilon_3,\ \ldots,\ \epsilon_l,\ c_{t+1},\ c_{t+2},\ \ldots,\ c_{t+h}, \epsilon_{\text{end}}]. \tag{6}$$

Small c_{t+j} indicates higher LLM confidence at step $t+j$. The history portion of the confidence covariate is filled with $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, where we assign a very small value to σ_ϵ^2 to teach the M-TSFM that the history portion of M_t resembles an LLM forecast with a very high confidence score.

Conditioned on M_t and C_t, an M-TSFM ($\theta_{\text{M-TSFM}}$) produces the forecast $\hat{X}_t^F = \theta_{\text{M-TSFM}}(X_t^H \oplus \emptyset \| M_t \| C_t)$. Here, $X_t^H \oplus \emptyset$ indicates padding the history X_t^H for h forecast timestamps and $\|$ denotes column-wise concatenation, such that $X_t^H \oplus \emptyset \| M_t \| C_t \in \mathbb{R}^{(l+h)\times 3}$ stacks the padded history and covariate channels along the feature dimension. For ease of notation, we drop $\emptyset$ and use $\hat{X}_t^F = \theta_{\text{M-TSFM}}(X_t^H \| M_t \| C_t)$ in the rest of the manuscript.

We fine-tune both the LLM (θ_{LLM}) and the M-TSFM ($\theta_{\text{M-TSFM}}$) in sequence instead of end-to-end joint fine-tuning. The LLM is fine-tuned first against a pretrained M-TSFM, and the M-TSFM is subsequently fine-tuned using covariates pre-generated from the fine-tuned LLM.

LLM Fine-Tuning: We fine-tune the LLM with GRPO [55] to produce forecast covariates with text-induced trends by optimizing it against a pretrained M-TSFM, using forecasting improvement as the reward. Although the M-TSFM is held fixed, it can still respond to the covariate since M_t is constructed as a noisy next-step continuation aligned with its pretraining inductive biases (Table 25). This allows the model to partially utilize M_t via in-context learning, making the relative forecasting error a meaningful reward signal. For a sample (X_t^H, S_t, X_t^F), we generate N_{rollouts} forecasts $\{\theta_{\text{LLM}}^{(g)}(X_t^H, S_t)\}_{g=1}^{N_{\text{rollouts}}}$ and construct covariates $M_t^{(g)}$. The reward and GRPO advantage are:

$$r_t^{(g)} = \frac{1}{1 + \frac{\|\theta_{\text{M-TSFM}}(X_t^H \| M_t^{(g)}) - X_t^F\|_1}{\|\theta_{\text{M-TSFM}}(X_t^H) - X_t^F\|_1 + \delta}}, \quad A_t^{(g)} = \frac{r_t^{(g)} - \frac{1}{N_{\text{rollouts}}} \sum_{i=1}^{N_{\text{rollouts}}} r_t^{(i)}}{\sigma(\{r_t^{(i)}\}_{i=1}^{N_{\text{rollouts}}}) + \delta}, \tag{7}$$

where a very small $\delta > 0$ prevents division by zero ($\delta = 10^{-8}$). The reward lies in $(0, 1)$, approaching 1 when the covariate improves forecasting and 0.5 when it provides no benefit. The

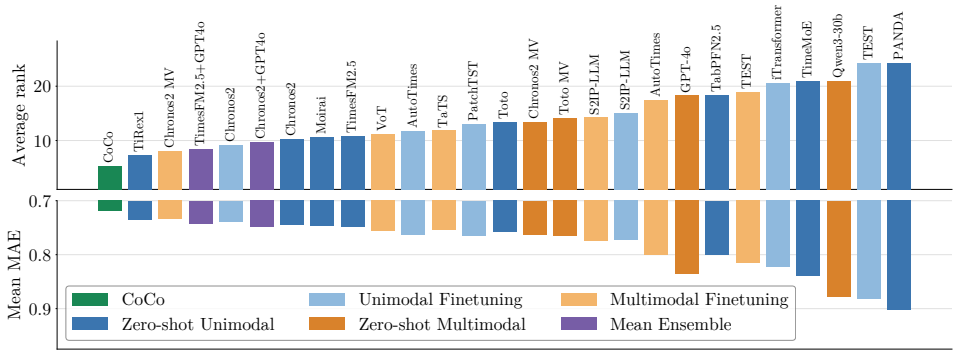


Figure 5: **CoCo improves over unimodal and multimodal fine-tuning approaches, including SOTA TSFMs on real-world multimodal datasets.** CoCo (ours) achieves the best average rank and mean MAE across 30 datasets (95 settings) spanning diverse domains and context lengths against 26 (TimeOmni omitted for scale) approaches. These gains demonstrate the effectiveness of GRPO-based fine-tuning along with our novel two-step fine-tuning.

confidence covariate C_t is excluded from reward computation due to the cost of additional rollouts. We denote $\theta_{\text{M-TSFM}}(X_t^H)$, $\theta_{\text{M-TSFM}}(X_t^H \| M_t)$, and $\theta_{\text{M-TSFM}}(X_t^H \| M_t \| C_t)$ as the unimodal, forecast-covariate, and full multimodal predictions, respectively.

M-TSFM Fine-Tuning: Following LLM fine-tuning, we pre-generate the full covariate set $\{M_t, C_t\}_{t=1}^N$ for all N training samples using the fine-tuned LLM. The M-TSFM is then fine-tuned by minimizing the MSE between its covariate-conditioned forecast and the ground truth: $\frac{1}{N}\sum_{t=1}^{N} \left\| \theta_{\text{M-TSFM}}(X_t^H \| M_t \| C_t) - X_t^F \right\|^2$. We note that fine-tuning is essential to teach the M-TSFM to interpret M_t as a soft next-step predictor and to modulate the reliance on M_t according to the per-step confidence signal C_t, producing forecasts that are context-aware and numerically precise.

This two-stage approach avoids the instability of optimizing against a moving target and keeps training tractable. The architecture choices and the hyperparameters are described in detail in Appendix E.1.

6 Experiments

We use `Qwen3-4B-Instruct-2507` [9] (Qwen3-4B) as the LLM and Chronos2 as the M-TSFM in CoCo, with both models fine-tuned as described in Section 5.

Training Datasets and Setup: We generated synthetic annotations (Section 4) for 36,826 time series datasets (∼12.4M rows) spanning retail, macroeconomics, environment, and finance, and curated ∼150,000 rows of real annotations from the finance [13] domain. Details of the datasets, annotation pipeline, and training (Section 5) are provided in Appendices E.2 and E.1.

Evaluation Datasets: We evaluate CoCo on a wide range of 30 datasets (over 16,719 samples) spanning macroeconomics, finance, retail, climate, security, and traffic domains. These datasets are partly curated from existing benchmarks, such as TimeMMD [11] (8 datasets) and FNSPID [13] (11 datasets). Additionally, we curated text annotations for 11 real-world time series datasets with time-stamped news articles extracted using the Media Cloud API [56]. A description of our evaluation datasets is provided in Appendix F.1. Our experiments used 64, 128, and 256 as context lengths and 16 as the forecast length. The TimeMMD dataset is also evaluated under the original paper settings (shown as TimeMMD*).

Baselines. We compare CoCo against 27 methods, categorized by modality and adaptation: eight zero-shot unimodal, seven zero-shot multimodal, six unimodal fine-tuning, and six multimodal fine-tuning approaches. For a controlled comparison, every fine-tuned baseline uses the same synthetic text–time-series pairs as CoCo. Appendix F.2 provides the complete list and implementation details.

Metrics. We report the per-dataset and aggregated (weighted equally to avoid bias from differing sample sizes) mean MAE. To aggregate relative forecasting performance across different settings, we report two complementary metrics: *Average Rank*, for which the models are ranked (best model ranked 1) in each setting, and the average rank across all the settings is reported; and *ELO Score*

	Mean MAE (↓)										Aggregate Performance				
	64–16 Setting			128–16 Setting			256–16 Setting			Native	Overall	Average	ELO	CoCo	Wilcoxon
Model	Media Cloud	FNSPID	TimeMMD	Media Cloud	FNSPID	TimeMMD	Media Cloud	FNSPID	TimeMMD	TimeMMD*	MAE ↓	Rank ↓	Rating ↑	Wins	p
CoCo (Ours)	**0.809**	0.907	**0.811**	**0.567**	**0.613**	**0.672**	**0.446**	0.453	**0.678**	**1.329**	**0.719**	**5.24**	**1838**	–	–
Zero-shot Unimodal															
TiRex1	0.835	0.915	0.824	0.580	0.616	0.696	0.450	**0.450**	0.738	1.352	0.735	7.26	1750	60/95	5.0e-04
Chronos2	0.847	0.942	0.832	0.590	0.631	0.701	0.461	0.462	0.718	1.362	0.745	10.18	1643	81/95	2.0e-12
Moirai	0.843	0.923	0.843	0.576	0.647	0.708	0.461	0.468	0.726	1.379	0.747	10.57	1638	78/95	9.2e-13
TimesFM2.5	0.861	0.941	0.824	0.597	0.643	0.699	0.468	0.469	0.732	1.342	0.749	10.78	1629	72/95	7.8e-09
Toto	0.865	0.955	0.836	0.589	0.660	0.710	0.473	0.484	0.737	1.358	0.758	13.36	1548	80/95	8.1e-13
TabPFN2.5	0.908	0.968	0.864	0.647	0.662	0.724	0.632	0.538	0.770	1.390	0.800	18.27	1385	90/95	1.4e-16
TimeMoE	1.001	1.050	0.939	0.647	0.710	0.768	0.653	0.574	0.750	1.378	0.839	20.82	1292	88/95	3.3e-15
PANDA	1.072	1.133	1.060	0.770	0.790	0.856	0.565	0.547	0.812	1.499	0.903	24.21	1134	93/95	4.1e-17
Zero-shot Multimodal															
TimesFM2.5+GPT-4o	0.829	**0.898**	0.857	0.575	0.625	0.720	0.456	0.456	0.742	1.385	0.742	8.35	1712	60/95	5.0e-03
Chronos2+GPT-4o	0.831	0.910	0.864	0.582	0.629	0.718	0.460	0.465	0.734	1.412	0.748	9.65	1664	65/95	6.3e-05
Chronos2 MV	0.892	0.947	0.870	0.607	0.639	0.715	0.516	0.475	0.721	1.341	0.763	13.42	1542	74/95	1.8e-10
Toto MV	0.902	0.962	0.844	0.600	0.668	0.706	0.499	0.487	0.719	1.348	0.766	14.04	1518	75/95	6.1e-10
GPT-4o	0.913	0.966	1.035	0.644	0.680	0.853	0.513	0.502	0.845	1.586	0.836	18.23	1388	85/95	8.9e-15
Qwen3-30B	0.949	0.985	1.080	0.691	0.717	0.927	0.546	0.539	0.902	1.631	0.878	20.96	1293	94/95	2.8e-17
TimeOmni	1.414	1.126	1.515	1.099	0.736	8.543	0.760	0.564	1.490	1.748	1.756	25.67	1054	95/95	2.6e-17

Table 1: **CoCo outperforms SOTA zero-shot unimodal and multimodal forecasting approaches on real-world multimodal datasets.** CoCo (ours) achieves the best mean MAE across 30 datasets (95 settings) spanning diverse domains and context lengths. It outperforms SOTA TSFMs such as TiRex1 in over 63% (60/95) of settings. Similarly, CoCo surpasses constructed SOTA zero-shot multimodal baselines (TimesFM2.5 + GPT-4o) in 60/95 settings with a significant improvement in the aggregate MAE and the average rank metrics, highlighting effective **sim-to-real** transfer.

	Mean MAE (↓)										Aggregate Performance				
	64–16 Setting			128–16 Setting			256–16 Setting			Native	Overall	Average	ELO	CoCo	Wilcoxon
Model	Media Cloud	FNSPID	TimeMMD	Media Cloud	FNSPID	TimeMMD	Media Cloud	FNSPID	TimeMMD	TimeMMD*	MAE ↓	Rank ↓	Rating ↑	Wins	p
CoCo (Ours)	**0.809**	0.907	0.811	**0.567**	**0.613**	0.672	**0.446**	**0.453**	0.678	**1.329**	**0.719**	**5.24**	**1838**	–	–
Unimodal Fine-tuning															
Chronos2	0.851	0.924	**0.810**	0.583	0.637	**0.670**	0.460	0.469	**0.675**	1.389	0.738	9.18	1677	64/95	1.2e-05
AutoTimes	0.851	0.916	0.901	0.585	0.633	0.773	0.515	0.473	0.784	1.337	0.763	11.75	1588	78/95	3.2e-10
PatchTST	0.858	0.927	0.903	0.603	0.638	0.762	0.469	0.492	0.772	1.332	0.764	13.06	1559	89/95	7.2e-15
S2IP-LLM	0.893	0.939	0.892	0.619	0.635	0.736	0.492	0.500	0.768	1.363	0.773	14.98	1491	87/95	3.5e-15
iTransformer	0.894	0.962	0.920	0.694	0.756	0.812	0.541	0.579	0.808	1.354	0.823	20.49	1300	92/95	6.1e-17
TEST	0.962	1.040	1.026	0.679	0.731	0.860	0.588	0.609	0.880	1.590	0.882	24.14	1145	94/95	2.7e-17
Multimodal Fine-tuning															
Chronos2 MV	0.845	0.905	0.834	0.573	0.625	0.697	0.462	0.461	0.702	1.334	0.734	8.02	1714	68/95	1.8e-05
VoT	0.836	**0.888**	0.885	0.594	0.628	0.753	0.530	0.488	0.745	1.339	0.756	11.06	1614	71/95	1.5e-08
TaTS	0.883	0.936	0.848	0.594	0.637	0.702	0.509	0.473	0.707	1.343	0.754	11.79	1592	70/95	1.3e-08
S2IP-LLM	0.858	0.910	0.896	0.610	0.647	0.758	0.528	0.546	0.771	1.338	0.775	14.18	1518	85/95	1.3e-13
AutoTimes	0.884	0.949	0.973	0.626	0.663	0.835	0.552	0.520	0.790	1.350	0.801	17.40	1414	83/95	2.3e-13
TEST	0.857	0.970	0.977	0.636	0.706	0.840	0.556	0.544	0.813	1.390	0.815	18.93	1359	91/95	2.7e-16

Table 2: **CoCo outperforms SOTA compute-matched and fine-tuned forecasting approaches on real-world multimodal datasets.** Similar to Table 1, here, we show CoCo's improvements over fine-tuned and compute-matched baselines. Observe how, even after fine-tuning, both unimodal and multimodal baselines struggle to match CoCo's performance.

(detailed in Appendix F.3). For statistical significance, we report the Wilcoxon p-value based on MAE computed across settings. The quantitative comparisons are summarized in Tables 1 and 2 with dataset-level metrics for each baseline category shown in Tables 13-16. For the Context-Is-Key benchmark, we report the Region of Interest Continuous Ranked Probability Score (RCRPS).

CoCo outperforms SOTA TSFMs and LLM-based forecasters consistently. We refer the readers to Tables 1 and 2, and Figure 5. The comparison against the SOTA TSFMs reveals the multimodal gains that can be achieved from textual annotations. Improvements over LLM-based forecasters showcase the ability of CoCo to effectively combine textual context and time series representations. Similarly, significant improvements over multimodal fine-tuning approaches elucidate the right approach to utilize textual annotations to improve SOTA TSFMs.

Synthetic annotations enable generalization to domains without real multimodal data. Table 24 shows that fine-tuning on real FNSPID data achieves strong in-domain performance (MAE 0.891) but degrades on out-of-distribution TimeMMD (MAE 1.528). In contrast, fine-tuning on synthetic annotations (Section 4) yields comparable FNSPID performance (MAE 0.903) and improved TimeMMD performance (MAE 1.365). Finally, fine-tuning on both real and synthetic annotations achieves the best overall results (TimeMMD MAE 1.354) without degrading FNSPID (MAE 0.894). These results show that fine-tuning on our synthetic annotations can enable generalization to domains lacking large-scale multimodal datasets, thereby showcasing **sim-to-real** transfer in multimodal forecasting.

Notably, neither supervised fine-tuning nor reinforcement learning (RL) fine-tuning without text achieves comparable gains, confirming that both the GRPO reward signal and the textual annotations are essential for fine-tuning.

CoCo's gains are strongly dependent on how effectively the LLM utilizes the text. We use the relative MAE improvement in the LLM forecast when the textual context S_t is provided versus withheld, as a measure of text usefulness. Positive values indicate that the text helped the LLM produce a better forecast, while negative values indicate misleading or incorrect text. Figure 7 (left) shows that text usefulness is a strong indicator of CoCo's gains. Figure 7 (right) shows that when text is highly useful, performance peaks at high LLM confidence and degrades as confidence decreases. Conversely, when text is not useful, high LLM confidence is actively harmful, since Chronos2 is misled by confidently wrong covariates. This demonstrates that C_t acts as an effective gating mechanism, allowing Chronos2 to selectively incorporate M_t based on its per-step reliability.

Fine-tuning	M_t	Both $M_t + C_t$
Only θ_{LLM}	0.21	1.64
Only $\theta_{\text{M-TSFM}}$	2.45	2.55
Both θ_{LLM} and $\theta_{\text{M-TSFM}}$	2.6	**3.36**

Table 3: **Using both M_t and C_t consistently outperforms M_t alone across all fine-tuning configurations**. We show the % MAE improvement over SOTA unimodal models. Sequential fine-tuning of both LLM θ_{LLM} and M-TSFM $\theta_{\text{M-TSFM}}$ yields the best % MAE improvement (Table 22).

Both fine-tuning stages and both covariates are crucial for forecasting performance. Tables 3, 22 and 28 show that fine-tuning both LLM and Chronos2, and providing both the covariates, yields the largest improvement (3.36% over the Chronos2 baseline), with each component contributing incrementally. Adding the confidence covariate on top of the forecast covariate consistently improves performance across all fine-tuning configurations, except for the zero-shot setting, where neither the LLM has learned to provide good covariates nor Chronos2 has learned to interpret them.

CoCo generalizes to multiple LLM $\times$ M-TSFM pairs. Tables 25 and 26, and Figure 6 demonstrate that CoCo is not specific to Qwen3-4B or Chronos2. Across five LLM configurations and three M-TSFM architectures, the forecast covariate M_t improves every evaluated LLM $\times$ M-TSFM combination, while the confidence covariate C_t provides further gains with zero-shot Toto and fine-tuned Chronos2. Although Chronos2 is fine-tuned using rollouts pre-generated only by the fine-tuned Qwen3-4B, it generalizes to covariates produced by all evaluated zero-shot LLMs. The fine-tuned Qwen3-4B also outperforms the larger zero-shot LLMs as a standalone forecaster (Figure 13).

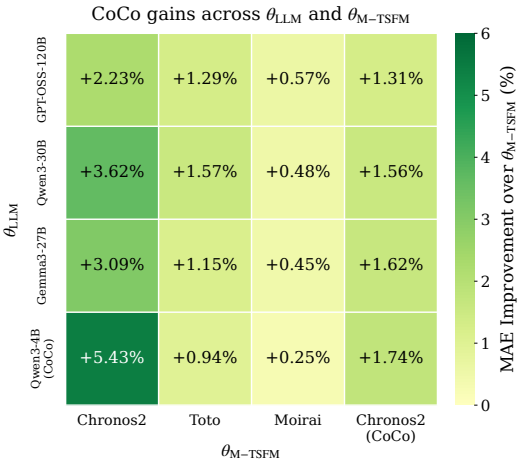


Figure 6: **CoCo generalizes across model choices.** MAE improvement over each numerical-only M-TSFM baseline on Media Cloud (64-16 setting).

CoCo improves text-time series alignment in pretrained LLMs. We evaluate this using the Context-Is-Key benchmark (CIK, Tables 4 and 8), where accurate forecasting requires strong cross-modal alignment between text and time series due to text-induced trends and spikes. The RCRPS metric further penalizes incorrect trends and shifted spikes. The CoCo fine-tuned LLM consistently outperforms its pretrained counterpart, showing that GRPO fine-tuning improves the LLM's ability to combine numerical forecasting with text-based reasoning. Notably, CoCo fine-tuned Qwen3-4B matches the performance of GPT-4o. We refer readers to Appendix D for detailed results.

Limitations. First, we note that CoCo inference is slower than unimodal TSFMs due to the additional LLM inference required for extracting predictive signals from text (Table 29), a capability fundamentally absent in TSFMs. Further, CoCo assumes that textual annotations are generally predictive of future trends. When the text is misleading, the LLM can generate confidently incorrect covariates that Chronos2 cannot fully correct, thereby degrading performance (Table 21, Figure 7 (left)). Finally,

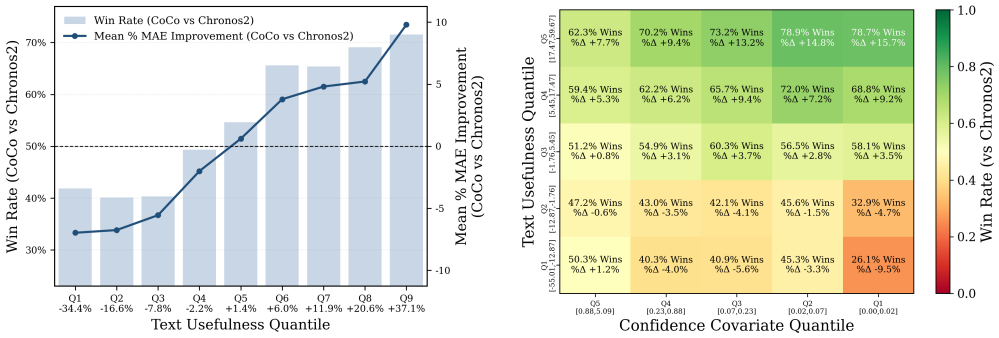


Figure 7: **(Left) CoCo consistently outperforms SOTA unimodal models (Chronos2) when text is useful (Q5–Q9) and underperforms when it is not (Q1–Q3).** Text usefulness is defined as the relative MAE improvement of the LLM forecast when the context S_t is provided versus withheld. **(Right) Furthermore, for the set of windows where text is useful, CoCo significantly outperforms unimodal models when the rollout variance from the LLM (Confidence C_t) is very low.**

CoCo is designed for settings where numerical history and textual context carry complementary predictive signals. In settings where text is the sole driver of forecast outcomes, the TSFM backbone contributes less, and we observe that CoCo's gains over just the LLM are limited (Appendix D).

Additional Experiments. Table 17 shows that CoCo continues to outperform zero-shot multimodal baselines on samples strictly after June 2024, suggesting robustness beyond potential pretraining overlap. To reduce sensitivity to dataset scale and outliers in aggregate MAE, Table 18 reports the geometric mean of MASE across datasets, following GIFT-Eval [57], and yields conclusions consistent with MAE. Table 19 further shows that CoCo's relative performance remains consistent across raw and normalized evaluation metrics. Table 22 demonstrates that the strongest multimodal performance is obtained by fine-tuning both the LLM and M-TSFM and using both M_t and C_t. Finally, across 30 Media Cloud settings, CoCo outperforms GPT-5 in 70% $(21/30)$ of cases and achieves up to an 11% MAE improvement on steel-price forecasting (Table 20). Table 12 shows CoCo's robustness to the stochasticity induced by the LLM rollouts. Table 23 shows that increasing the number of rollouts beyond $N_{\text{rollouts}} = 5$ provides minimal gains. We observe that generating textual annotations from historical numerical values alone produces textual annotations that do not reflect the real forecast trends. On the other hand, future-conditioned textual annotation generation reflects the true forecast trends better (Table 27).

Model	**Mean RCRPS ↓**
Qwen3-4B	0.358
Qwen3-4B + Reasoning	0.189
Qwen3-4B (CoCo)	0.301
Qwen3-4B (CoCo) + Reasoning	0.161
Chronos2 (Unimodal)	0.265
CoCo	0.256
GPT-4o (reference)	*0.191*

Table 4: **LLMs fine-tuned with CoCo have improved text-time series alignment and outperform their pretrained counterparts.** Notably, Qwen3-4B fine-tuned with CoCo, with reasoning, matches the zero-shot results from GPT-4o. We refer to Table 8 for the detailed results.

7 Conclusion and Future Work

In this work, we first proposed a simple yet effective recipe for generating large-scale multimodal time series datasets. Specifically, we focused on text as the alternative modality. Secondly, we proposed **Co**ntext as **Co**variates, a novel multimodal architecture for incorporating orthogonal textual information to refine the forecasts from SOTA TSFMs. CoCo achieves up to 12% MAE improvement over state-of-the-art TSFMs and LLM forecasters. As future work, we aim to add a text reliability assessment step to CoCo to improve the unimodal forecast only when text is useful and not misleading. This can be empirically assessed using the relationship between the text and time series in the context or history. Additionally, the current setup involves fine-tuning the LLM and M-TSFM sequentially. To this end, we aim to explore the possibility of jointly fine-tuning both models.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science Foundation under Grant No. 2148186. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Rui Liu, Jiayou Liang, Haolong Chen, and Yujia Hu. Analyst reports and stock performance: Evidence from the chinese market, 2025. URL https://arxiv.org/abs/2411.08726.

[2] Kevin Taylor and Jerry Ng. Natural language processing and multimodal stock price prediction, 2024. URL https://arxiv.org/abs/2401.01487.

[3] Kamaladdin Fataliyev, Aneesh Chivukula, Mukesh Prasad, and Wei Liu. Stock market analysis with text data: A review, 2021. URL https://arxiv.org/abs/2106.12985.

[4] Kemal Kirtac and Guido Germano. Sentiment trading with large language models. *Finance Research Letters*, 62:105227, April 2024. ISSN 1544-6123. doi: 10.1016/j.frl.2024.105227. URL http://dx.doi.org/10.1016/j.frl.2024.105227.

[5] Ruize Gao, Feng Xiao, Jinpu Li, and Shaoze Cui. Textual semantics and machine learning methods for data product pricing, 2025. URL https://arxiv.org/abs/2511.22185.

[6] Nway Nway Aung, Yuejingxian Mao, Chandra Suwandi Wijaya, Aryel Beck, Miura Koji, and Tajika Yosuke. Demand forecasting: Integrating textual and market data to navigate irregular market disruption. ADMIT '24, page 270–280, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400718120. doi: 10.1145/3701100.3701158. URL https://doi.org/10.1145/3701100.3701158.

[7] 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.

[8] 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, 2025. URL https://arxiv.org/abs/2510.15821.

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

[10] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[11] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-mmd: Multi-domain multimodal dataset for time series analysis, 2025. URL https://arxiv.org/abs/2406.08627.

[12] Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ih2WuBT1Fn.

[13] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. Fnspid: A comprehensive financial news dataset in time series, 2024. URL https://arxiv.org/abs/2402.06698.

[14] Junwoo Park, Hyuck Lee, Dohyun Lee, Daehoon Gwak, and Jaegul Choo. Revisiting llms as zero-shot time-series forecasters: Small noise can break large models, 2025. URL https://arxiv.org/abs/2506.00457.

[15] Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.

[16] Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024.

[17] Kai Kim, Howard Tsai, Rajat Sen, Abhimanyu Das, Zihao Zhou, Abhishek Tanpure, Mathew Luo, and Rose Yu. Multi-modal forecaster: Jointly predicting time series and textual data, 2024. URL https://arxiv.org/abs/2411.06735.

[18] 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, 2024. URL https://arxiv.org/abs/2310.01728.

[19] Yueyang Yao, Jiajun Li, Xingyuan Dai, MengMeng Zhang, Xiaoyan Gong, Fei-Yue Wang, and Yisheng Lv. Context-aware probabilistic modeling with llm for multimodal time series forecasting, 2025. URL https://arxiv.org/abs/2505.10774.

[20] Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. Explainable multi-modal time series prediction with llm-in-the-loop. *arXiv preprint arXiv:2503.01013*, 2025.

[21] Mengpu Liu, Mengying Zhu, Xiuyuan Wang, Guofang Ma, Jianwei Yin, and Xiaolin Zheng. Echo-gl: Earnings calls-driven heterogeneous graph learning for stock movement prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13972–13980, 2024.

[22] Ramit Sawhney, Arnav Wadhwa, Shivam Agarwal, and Rajiv Shah. Fast: Financial news and tweet based time aware network for stock trading. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 2164–2175, 2021.

[23] Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, 2018.

[24] Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1627–1630, 2018.

[25] Zihan Chen, Lei Nico Zheng, Cheng Lu, Jialu Yuan, and Di Zhu. Chatgpt informed graph neural network for stock movement prediction. *arXiv preprint arXiv:2306.03763*, 2023.

[26] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pages 222–235, 2020.

[27] Sandra Śmigiel, Krzysztof Pałczyński, and Damian Ledziński. Ecg signal classification using deep learning techniques based on the ptb-xl dataset. *Entropy*, 23(9):1121, 2021.

[28] Lecheng Zheng, Zhengzhang Chen, Dongjie Wang, Chengyuan Deng, Reon Matsuoka, and Haifeng Chen. Lemma-rca: A large multi-modal multi-domain dataset for root cause analysis, 2025. URL https://arxiv.org/abs/2406.05375.

[29] Wei Chen, Xixuan Hao, Yuankai Wu, and Yuxuan Liang. Terra: A multimodal spatio-temporal dataset spanning the earth. *Advances in Neural Information Processing Systems*, 37:66329–66356, 2024.

[30] Zhijian Xu, Yuxuan Bian, Jianyuan Zhong, Xiangyu Wen, and Qiang Xu. Beyond trend and periodicity: Guide time series forecasting with textual cues.

[31] Xin Zhou, Weiqing Wang, Francisco J. Baldán, Wray Buntine, and Christoph Bergmeir. Motime: A dataset suite for multimodal time series forecasting, 2025. URL https://arxiv.org/abs/2505.15072.

[32] Xusen Guo, Qiming Zhang, Junyue Jiang, Mingxing Peng, Meixin Zhu, and Hao Frank Yang. Towards explainable traffic flow prediction with large language models. *Communications in Transportation Research*, 4:100150, 2024.

[33] Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. *arXiv preprint arXiv:2503.01875*, 2025.

[34] Qianqian Xie, Weiguang Han, Yanzhao Lai, Min Peng, and Jimin Huang. The wall street neophyte: A zero-shot analysis of chatgpt over multimodal stock movement prediction challenges. *arXiv preprint arXiv:2304.05351*, 2023.

[35] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

[36] Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5351–5362, 2024.

[37] Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*, 2023.

[38] 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, pages 23343–23351, 2024.

[39] Lu Han, Yu Liu, Lan Li, Qiwen Deng, Jian Jiang, Yinbo sun, Zhe Yu, Binfeng Wang, Xingyu Lu, Lintao Ma, Han-Jia Ye, and De-Chuan Zhan. UniCA: Unified covariate adaptation for time series foundation model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=I8q4MZb4OP.

[40] Zihao Li, Xiao Lin, Zhining Liu, Jiaru Zou, Ziwei Wu, Lecheng Zheng, Dongqi Fu, Yada Zhu, Hendrik Hamann, Hanghang Tong, et al. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative. In *International Conference on Learning Representations*, volume 2026, pages 24437–24484, 2026.

[41] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, pages 4095–4106, 2024.

[42] Xiaobin Hong, Jiawen Zhang, Wenzhong Li, Sanglu Lu, and Jia Li. Unify and anchor: A context-aware transformer for cross-domain time series forecasting, 2025. URL https://arxiv.org/abs/2503.01157.

[43] 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, 2025. URL https://arxiv.org/abs/2406.01638.

[44] Sameep Chattopadhyay, Pulkit Paliwal, Sai Shankar Narasimhan, Shubhankar Agarwal, and Sandeep P Chinchali. Context matters: Leveraging contextual features for time series forecasting. *arXiv preprint arXiv:2410.12672*, 2024.

[45] Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. Timecap: Learning to contextualize, augment, and predict time series events with large language model agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18082–18090, 2025.

[46] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8415–8426, 2020.

[47] Mayank Daswani, Mathias M. J. Bellaiche, Marc Wilson, Desislav Ivanov, Mikhail Papkov, Eva Schnider, Jing Tang, Kay Lamerigts, Gabriela Botea, Michael A. Sanchez, Yojan Patel, Shruthi Prabhakara, Shravya Shetty, and Umesh Telang. Plots unlock time-series understanding in multimodal models, 2024. URL https://arxiv.org/abs/2410.02637.

[48] Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series, 2023. URL https://arxiv.org/abs/2303.12799.

[49] Zihao Zhou and Rose Yu. Can llms understand time series anomalies?, 2025. URL https://arxiv.org/abs/2410.05440.

[50] 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. *arXiv preprint arXiv:2502.04395*, 2025.

[51] Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-2.5: Advancing the state of the art in tabular foundation models, 2026. URL https://arxiv.org/abs/2511.08667.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

[53] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[54] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[55] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[56] Hal Roberts, Rahul Bhargava, Linas Valiukas, Dennis Jen, Momin M. Malik, Cindy Sherman Bishop, Emily B. Ndulue, Aashka Dave, Justin Clark, Bruce Etling, Robert Faris, Anushka Shah, Jasmin Rubinovitz, Alexis Hope, Catherine D'Ignazio, Fernando Bermejo, Yochai Benkler, and Ethan Zuckerman. Media cloud: Massive open source collection of global news on the open web. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):1034–1045, May 2021. doi: 10.1609/icwsm.v15i1.18127. URL https://ojs.aaai.org/index.php/ICWSM/article/view/18127.

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

[58] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[59] Vijay Ekambaram, Kushagra Manglik, Sumanta Mukherjee, Surya Shravan Kumar Sajja, Satyam Dwivedi, and Vikas Raykar. Attention based multi-modal new product sales time-series forecasting. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3110–3118, 2020.

[60] Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, sujay sanghavi, and Sandeep P. Chinchali. Time weaver: A conditional time series generation model. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=WpKDeixmFr.

[61] Sai Shankar Narasimhan, Shubhankar Agarwal, Litu Rout, Sanjay Shakkottai, and Sandeep P. Chinchali. Constrained posterior sampling: Time series generation with hard constraints. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=iBLAYaEKdL.

[62] Yunfeng Ge, Jiawei Li, Yiji Zhao, Haomin Wen, Zhao Li, Meikang Qiu, Hongyan Li, Ming Jin, and Shirui Pan. T2s: High-resolution time series generation with text-to-series diffusion models, 2025. URL https://arxiv.org/abs/2505.02417.

[63] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second, 2023. URL https://arxiv.org/abs/2207.01848.

[64] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[65] Prime Intellect. Prime-rl: Agentic rl training at scale, 2025. URL https://github.com/PrimeIntellect-ai/prime-rl.

[66] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

[67] Song Chen. Beijing Multi-Site Air Quality. UCI Machine Learning Repository, 2017. DOI: https://doi.org/10.24432/C5RK5G.

[68] 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, 2025. URL https://arxiv.org/abs/2505.23719.

[69] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024. URL https://arxiv.org/abs/2402.02592.

[70] Ben Cohen, Emaad Khwaja, Kan Wang, Charles Masson, Elise Ramé, Youssef Doubli, and Othmane Abou-Amal. Toto: Time series optimized transformer for observability, 2024. URL https://arxiv.org/abs/2407.07874.

[71] 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, 2025. URL https://arxiv.org/abs/2409.16040.

[72] Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for chaotic dynamics, 2026. URL https://arxiv.org/abs/2505.13755.

[73] Haoxin Liu, Yichen Zhou, Rajat Sen, B. Aditya Prakash, and Abhimanyu Das. Rethinking multimodal time-series forecasting evaluation, 2026. URL https://arxiv.org/abs/2607.06973.

[74] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

[75] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

[76] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models, 2024. URL https://arxiv.org/abs/2402.02370.

[77] Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. $\mathbf{S}^2$ip-llm: Semantic space informed prompt learning with llm for time series forecasting, 2024. URL https://arxiv.org/abs/2403.05798.

[78] Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series, 2024. URL https://arxiv.org/abs/2308.08241.

[79] 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, 2026. URL https://arxiv.org/abs/2603.15452.

[80] Arpad E Elo. The proposed uscf rating system, its development, theory, and applications. *Chess life*, 22(8): 242–247, 1967.

[81] Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. Elo uncovered: Robustness and best practices in language model evaluation, 2023. URL https://arxiv.org/abs/2311.17295.

[82] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024. URL https://arxiv.org/abs/2403.04132.

Appendix Overview

A Extended Related Works **17**

A.1 Multimodal Datasets for Training and Evaluation 17

A.2 Multimodal Forecasting Approaches 17

B Proofs for Theorems in Section 4 **20**

B.1 Assumption Validity Discussion 21

C Synthetic Toy Example Description **22**

D Evaluation on Context is Key (CiK) **23**

E Training **25**

E.1 Training Description 25

E.2 Training Datasets 26

F Evaluation **28**

F.1 Evaluation Datasets 28

F.2 Evaluation Baselines 30

F.3 Evaluation Metrics 31

G CoCo's Robustness to Stochasticity **31**

H Extended Results **32**

I LLM Prompts **44**

I.1 Synthetic Annotation Prompt 44

I.2 LLM Forecast Prompt 44

I.3 Media Cloud Dataset Collection 45

J Broader Societal Impacts **46**

A Extended Related Works

A.1 Multimodal Datasets for Training and Evaluation

Multiple prior works have proposed multimodal time series datasets with primary focus on textual annotations for forecasting. The nature and richness of textual annotations vary widely across domains. In particular, numerous works [21, 22, 16, 23–25] focus on stock price prediction using auxiliary information, such as news articles, Twitter sentiments, and Google search trends. This is primarily due to the abundance of rich and time-varying textual annotations that describe the trends in stock price values, such as the daily opening, closing, high, and low prices. For example, FNSPID [13] contains stock prices and aligned textual annotations for over 4,000 companies from 1999 to 2023. In contrast, textual annotations in the healthcare domain, [26, 27] are typically global and static, in the form of patient medical records for electrocardiogram (ECG) signals. Such datasets have been primarily used for tasks like classification and anomaly detection. Similarly, Zheng et al. [28] have a large collection of performance metrics time series that are logged from complex systems for root-cause analysis. In the weather prediction domain, Chen et al. [29] provide static descriptions of the geography, while Xu et al. [30] focus on time-varying textual annotations that aid in forecasting temperature, rain levels, and other weather metrics. More recently, Zhou et al. [31] proposed Mo-Time, a multimodal dataset with static annotations spanning e-commerce and social media user behavior domains. From these prior works, our key takeaway is that large-scale multimodal datasets are generally domain-specific and localized (annotations are available for a short time frame and are restricted in terms of geographic locations). Additionally, based on the domains, the annotations can be static or time-varying.

TimeMMD [11] and Context-Is-Key [12] provide a collection of multimodal datasets that span healthcare, environment, finance, and retail domains. However, these efforts are restricted in the sizes of their individual datasets, and are mainly suitable for evaluation and not large-scale training. Additionally, curating these datasets is expensive owing to the significant human effort that was required.

A.2 Multimodal Forecasting Approaches

We group the existing works into zero-shot or training-free and dataset-specific training-based approaches. A common methodology for training-free approaches involves repurposing LLMs for time series forecasting, given their ability to process textual inputs efficiently.

A.2.1 Zero-Shot Approaches

Guo et al. [32] perform multimodal traffic flow prediction by converting information from other modalities into textual prompts for an LLM to capture complex patterns in time series and associate them with textual annotations. Kong et al. [33] fine-tuned LLMs (domain- or dataset-agnostic) with Low-Rank Adaptation (LoRA) for forecasting, imputation, and classification, by converting multimodal time series datasets and tasks into a dataset with questions and answers.

Wang et al. [16] used reasoning agents to filter relevant news articles or textual annotations along with domain- or dataset-agnostic fine-tuning and evaluation steps that update LLMs to effectively utilize time series and textual annotations, and evaluate whether all relevant annotations have been retrieved, respectively. The authors show that the proposed approach generalizes for the electricity, finance, and traffic domains.

In the finance domain, Xie et al. [34] evaluated ChatGPT's [10] ability to forecast stock price movements. However, the results indicate that this approach does not perform better than multimodal approaches based on domain-specific training for the same task. Yu et al. [15] focused on price prediction of NASDAQ-100 stocks using historical stock price data, company metadata, and financial news, with LLMs.

Kim et al. [17] proposed Time Text Corpus, a multimodal dataset with around 10,000 timestamps from the weather and the healthcare domains. Additionally, the authors introduced the Hybrid Multi-Modal Forecaster (Hybrid-MMF) which involves a two-stage training setup to jointly predict text and time series. However, the experiments are limited in both context (history, up to 7 timestamps) and forecast lengths.

A.2.2 Dataset-Specific Training-Based Approaches

Li et al. [40] employed a learnable text encoder to project the time-varying textual annotations into the embedding space of learnable time series forecasting models, thereby enabling multimodal training for forecasting.

Li et al. [36] presented UrbanGPT, a hybrid approach that integrates a spatiotemporal dependency encoder with the instruction-tuning paradigm in LLMs with the objective of forecasting traffic flow patterns in New York. This provides the ability for LLMs to understand complex spatial and temporal relationships in multivariate time series datasets.

Liu et al. [37] demonstrate the ability of LLMs to perform classification and prediction for physiological and medical time series using a few-shot tuning approach.

Chattopadhyay et al. [44] introduce ContextFormer, a dataset-specific fine-tuning procedure for unimodal time series forecasting models with textual annotations provided as embedding inputs from open-source LLMs.

Jia et al. [38] fine-tuned a pretrained GPT2 [58] model with time series and textual annotations for large-scale events datasets, curated using the Global Database of Events, Language, and Tone (GDELT) tool. Jia et al. [38] generated relevant time-varying textual annotations from GDELT by filtering the required news articles using cosine similarity with respect to the natural language description of the forecasting target. The quantitative results indicate that the use of highly useful text annotations indeed helps in forecasting events more accurately than unimodal models trained on the same dataset.

Liu et al. [43] employed a cross-modal alignment setup between the time series and text embeddings, for textual annotations that describe the trend and seasonality in the time series. Here, Liu et al. [43] trained a time series embedder from scratch and utilized off-the-shelf LLMs to extract text embeddings. Though the approach outperforms other SOTA trainable approaches, the text annotations mainly provide a summary of the time series and do not provide auxiliary information. Similarly, Ekambaram et al. [59] proposed an attention-based encoder-decoder framework for image-based product sales forecasting, while also including holidays, discounts, etc.

Jiang et al. [20] proposed TimeXL, where a multimodal encoder processes both time series and textual inputs to generate initial forecasts and aided explanations. Then, a prediction LLM refines the forecasts by reasoning over or aligning the time series predictions with the explanations. Consequently, a reflection LLM compares the predicted forecast against the ground truth to identify textual inconsistencies. Overall, TimeXL proposes a closed-loop framework, where the prediction LLM aligns the text and time series predictions, and the refinement LLM iteratively enhances text quality.

TimeLLM [18] learns patch embeddings for time series to align time series representations with the token embedding space of LLMs, so that LLMs can be utilized to forecast or predict the next time series patch. Additionally, TimeLLM deals with static textual annotations, which are broad descriptions of the domain and the geography corresponding to the time series dataset. However, this approach does not deal with complex time-varying annotations that provide auxiliary information to aid forecasting.

TimeCAP [45] utilizes two LLM agents, one summarizes the time series and the other uses aligned representations (learned using a multimodal encoder) of time series and its summary to forecast. Similar to other training-based approaches, TimeCAP performs dataset-specific training.

UniTime [41] and Hong et al. [42] deal with utilizing dataset descriptions (static textual annotations) to better learn time series forecasting for individual domains. Again, these annotations usually describe the dataset domain and the geographic region corresponding to where the dataset was collected, instead of time-varying auxiliary information, as in the case of finance datasets. Hong et al. [42] also generate similar synthetic annotations for time series windows. However, they do not show the forecast values as inputs and primarily summarize the historical values.

Recently, UniCA [39] proposed a unified covariate adaptation framework for TSFMs, where heterogeneous modalities such as text, categorical metadata, and numerical covariates are projected into a shared temporal representation space and fused into pretrained TSFMs using lightweight attention-based adapters. While UniCA enables covariate-aware forecasting across domains, the approach still relies on dataset-specific fine-tuning.

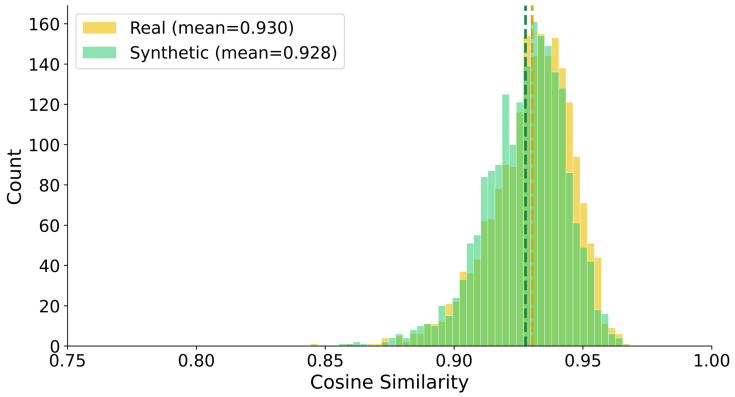


Figure 8: **Real and synthetic annotations contain similar predictive powers to estimate the future.** The significant overlap of cosine similarity distributions between the true and the predicted forecast descriptions for synthetic and real annotations validates our synthetic annotation strategy. For the real cosine similarity distribution, we compare the normalized BERT embeddings for the domain-agnostic forecast predictions from real text with the true domain-agnostic forecast description. Similarly, for the synthetic cosine similarity distribution, we compare the normalized BERT embeddings for the domain-agnostic forecast predictions from synthetic text with the true domain-agnostic forecast description. For this experiment, we consider the Media Cloud datasets for which the real annotations are available, and we generated synthetic annotations for those datasets.

TimeVLM [50] uses pre-trained Vision-Language Models (VLMs), where the language encoders capture contextual information and the vision encoders capture trend and seasonality using time series features represented as images. The approach learns the parameters of the fusion layers that enrich time series representations with the image and the language embeddings for each dataset. Again, the annotations are restricted to the descriptions of the dataset domain. Daswani et al. [47] and Li et al. [48] also suggest similar empirical conclusions regarding viewing time series as images or plots. Daswani et al. [47] benchmarked the performances of the SOTA LLMs, such as GPT-4o [10], in various time series tasks, including classification, detection, and prediction. Similarly, Li et al. [48] used pre-trained vision encoders for time series classification tasks. Zhou and Yu [49] explored anomaly detection using multimodal LLMs with time series inputs in the form of images.

Sawhney et al. [46] proposed an early fusion approach that combines learnable representations for time series and text annotations to predict the stock price movements. Similarly, Chen et al. [25] employed ChatGPT [10] to predict possible stocks that might be affected for the next day, along with the direction of change (positive or negative). This is combined with learnable time series embeddings, obtained from historical time series values, to predict the movements.

Narasimhan et al. [60, 61] and Ge et al. [62] focus on developing dataset-specific models for multimodal time series generation. Here, note that the inputs provided to the model are non-causal, *i.e.,* the inputs describe the values of all the timestamps in the time series to be generated. In contrast, to evaluate forecasting, we require annotations that are causal and contain predictions or speculations about the forecast timestamps.

Although most of the approaches mentioned above reuse SOTA LLMs for forecasting, Park et al. [14] highlighted the key issues that limit the ability of LLMs, such as sensitivity to noise. Additionally, Park et al. [14] show the inferior performance of LLMs in comparison to light-weight linear models (DLinear) on standard time series forecasting benchmarks.

From all these approaches, we find some common trends: (i) Zero-shot / few-shot approaches mainly rely on the predictive power of LLMs; (ii) Most approaches are domain-specific and fine-tune LLMs/VLMs for individual datasets or domains; (iii) Previous works did not utilize powerful TSFMs such as Chronos2 [8], TimesFM2.5 [7], TabPFN-TS [63]. We differentiate our approach from these prior works by seeking to develop a zero-shot solution that corrects/adapts unimodal forecasts from SOTA TSFMs using textual annotations. In addition, we are seeking to develop a framework that can work in multiple domains.

B Proofs for Theorems in Section 4

Under our latent model, we have history, forecast description, and the forecast sampled from $p(X^H, Z, X^F)$. Real annotations are sampled as $S^r \sim p_r(S^r \mid X^H, Z)$, while synthetic annotations are sampled as $S^s \sim p_s(S^s \mid X^H, Z, X^F)$. We define the real and synthetic population risks as

$$R^r_{\text{multi}}(\theta) = \mathbb{E}_{(X^H,Z,X^F)\sim p,\ S^r\sim p_r(\cdot|X^H,Z)}\left[\mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \phi(S^r)), X^F\big)\right], \tag{8}$$

$$R^s_{\text{multi}}(\theta) = \mathbb{E}_{(X^H,Z,X^F)\sim p,\ S^s\sim p_s(\cdot|X^H,Z,X^F)}\left[\mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \phi(S^s)), X^F\big)\right]. \tag{9}$$

Here, note that real annotations are causal and carry information about the forecast trends in domain-specific terms. Therefore, there is no conditioning on X^F in $S^r \sim p_r(S^r \mid X^H, Z)$. However, during training, the synthetic annotations are generated using an LLM which considers the history X^H and the forecast X^F to generate the domain-specific description. Additionally, we note that Z can be extracted from X^F easily by computing certain domain-agnostic statistical features. Therefore, for synthetic annotations, we have $S^s \sim p_s(S^s \mid X^H, Z, X^F)$.

For analysis, we also define the risk with the true domain-agnostic forecast description.

$$R^Z_{\text{multi}}(\theta) = \mathbb{E}_{(X^H,Z,X^F)\sim p}\left[\mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \psi(Z)), X^F\big)\right]. \tag{10}$$

Proof of Lemma 4.3. We first introduce $R^Z_{\text{multi}}(\theta)$.

$$|R^r_{\text{multi}}(\theta) - R^s_{\text{multi}}(\theta)| = |R^r_{\text{multi}}(\theta) - R^Z_{\text{multi}}(\theta) + R^Z_{\text{multi}}(\theta) - R^s_{\text{multi}}(\theta)| \tag{11}$$

Now, using the triangle inequality, we have

$$|R^r_{\text{multi}}(\theta) - R^s_{\text{multi}}(\theta)| \le |R^r_{\text{multi}}(\theta) - R^Z_{\text{multi}}(\theta)| + |R^Z_{\text{multi}}(\theta) - R^s_{\text{multi}}(\theta)|. \tag{12}$$

For the first term, we have

$$\begin{aligned}
&|R^r_{\text{multi}}(\theta) - R^Z_{\text{multi}}(\theta)| \\
&= \left|\mathbb{E}_{p,\,p_r}\Big[\mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \phi(S^r)), X^F\big) - \mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \psi(Z)), X^F\big)\Big]\right| \\
&\le \mathbb{E}_{p,\,p_r}\Big[\Big|\mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \phi(S^r)), X^F\big) - \mathcal{M}\big(h^{\text{multi}}_\theta(X^H, \psi(Z)), X^F\big)\Big|\Big] \\
&\le L\,\mathbb{E}_{p,\,p_r}\left[\|\phi(S^r) - \psi(Z)\|\right] &(13)\\
&\le L\varepsilon_r. &(14)
\end{aligned}$$

Here the first inequality is from Jensen's inequality, the second inequality follows from **Assumption 4.2** and the last inequality follows from **Assumption 4.1**.

Similarly, for the second term in Equation 12, we have

$$|R^s_{\text{multi}}(\theta) - R^Z_{\text{multi}}(\theta)| \le L\,\mathbb{E}_{p,\,p_s}\left[\|\phi(S^s) - \psi(Z)\|\right] \le L\varepsilon_s. \tag{15}$$

Combining the two bounds gives

$$|R^r_{\text{multi}}(\theta) - R^s_{\text{multi}}(\theta)| \le L(\varepsilon_r + \varepsilon_s). \tag{16}$$

□

Proof of Theorem 4.4. We decompose $R^r_{\text{multi}}(\hat{\theta}_s) - R^r_{\text{multi}}(\theta^\star_r)$ as

$$\begin{aligned}
R^r_{\text{multi}}(\hat{\theta}_s) - R^r_{\text{multi}}(\theta^\star_r) &= \underbrace{R^r_{\text{multi}}(\hat{\theta}_s) - R^s_{\text{multi}}(\hat{\theta}_s)}_{\text{(I)}} &(17)\\
&+ \underbrace{R^s_{\text{multi}}(\hat{\theta}_s) - R^s_{\text{multi}}(\theta^\star_r)}_{\text{(II)}} &(18)\\
&+ \underbrace{R^s_{\text{multi}}(\theta^\star_r) - R^r_{\text{multi}}(\theta^\star_r)}_{\text{(III)}}. &(19)
\end{aligned}$$

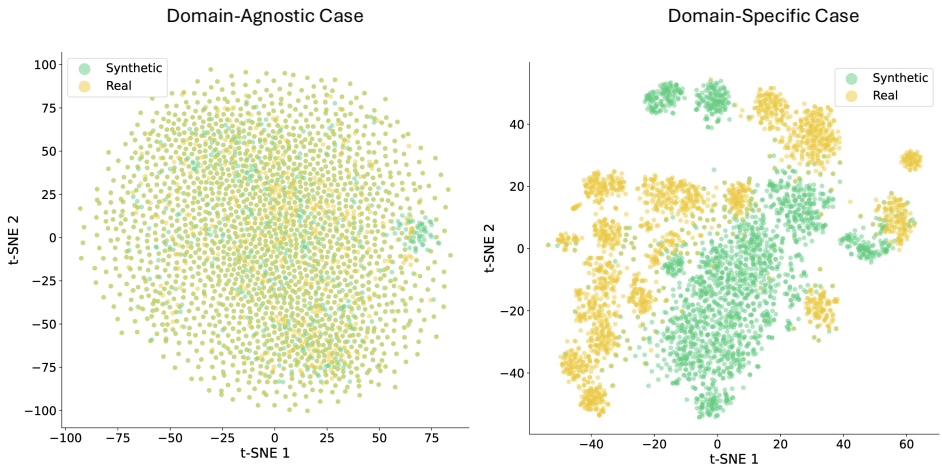


Figure 9: **Real and synthetic annotations contain similar domain-agnostic descriptions, but differ in semantics.** Here, we show the t-SNE plots for concatenated embeddings of real and synthetic annotations. We consider the Media Cloud datasets for which the real annotations are available, and we generated synthetic annotations for those datasets. On the left, we compare the embeddings of real domain-agnostic predictions concatenated with true forecast descriptions against the synthetic counterpart. Similarly, on the right, we compare the embeddings of the real domain-specific prediction concatenated with the true forecast descriptions against the synthetic counterpart. Note that there is near-perfect overlap in the domain-agnostic case and clear separation in the domain-specific case. This conveys that for the same true forecast description (through concatenation), the domain-agnostic predictions overlap indicating that real and synthetic annotations provide similar domain-agnostic prediction. In contrast, the domain-specific predictions vary significantly, indicating that both real and synthetic domain-specific predictions have different semantic constructions.

Term (II) is non-positive since $\hat{\theta}_s$ minimizes R^s_{multi} over Θ. By **Lemma 4.3**, terms (I) and (III) are each bounded above by $L(\varepsilon_r + \varepsilon_s)$. Therefore,

$$R^r_{\text{multi}}(\hat{\theta}_s) \leq R^r_{\text{multi}}(\theta^\star_r) + 2L(\varepsilon_r + \varepsilon_s). \tag{20}$$

□

Observe from Table 5 that changing the text only affects the MAE numbers in the third decimal for the LLM forecast as well as CoCo. To this end, we note that for reasonable forecasters, the smoothness property is generally satisfied in practice. As we explained in Section 4, the key ingredient for sim-to-real transfer is the proximity of real and synthetic annotations to the real future description in some latent space. Given the existence of foundation models that can act as powerful encoders, if the annotations contain speculations of the expected trend in the forecast, it is fair to expect small values for ε_s and ε_r, indicating good proximity in the latent space. Consequently, the bounded gap of $2L(\varepsilon_r + \varepsilon_s)$ tends to zero for very small ε_s and ε_r, indicating that the model trained on synthetic annotations will perform as well as the model trained on real annotations. Additionally, our claim of real and synthetic annotations capturing similar domain-agnostic predictions in Section 4 is visually depicted in Figure 9.

B.1 Assumption Validity Discussion

Assumption 4.1 focuses on the proximity of domain-agnostic forecast descriptions inferred from real and synthetic annotations to the domain-agnostic description of the true future in a common representation space. In Figure 8, we show the strong overlap in cosine similarity distributions between the embeddings of the verbal description of the forecast predictions derived from real and synthetic annotations. Note that the cosine similarity is with respect to the true future descriptions. As mentioned in Section 4, this indicates that synthetic annotations recover the forecast nearly as well as real ones, suggesting $\varepsilon_s \approx \varepsilon_r$. Additionally, the generation process grounds synthetic text in

numerical trajectories and domain descriptions. In our experiments, the embeddings correspond to normalized BERT embeddings, and consequently, the cosine similarity and the Euclidean distance (used in the assumption) are equivalent measures.

To this end, we directly quantify these representation distances on our real-world evaluation datasets using the same unit-normalized BERT embeddings, on a per-domain basis, in Table 7. Across all domains, synthetic annotations exhibit nearly the same distance to the true forecast description as real annotations. Therefore, the significant overlap in the cosine similarity distribution (Figure 8) with a very high mean (~ 0.93) and the similar representation distances between the real and synthetic annotations from the true forecast descriptions (Table 7) convey the same idea that real and synthetic annotations encode domain-agnostic information similar to the true forecast description.

Additionally, we note that the Equations 13 and 15 require only the expected representation distances to be bounded rather than the per-sample bound stated in Assumption 4.1. Therefore, relaxing Assumption 4.1 to reflect the bound on the expected representation distances between the real and synthetic forecast descriptions and the true forecast descriptions would still be sufficient to prove Lemma 4.3 and Theorem 4.4.

Assumption 4.2 focuses on ensuring that small perturbations in the text embedding produce limited changes in forecasting errors. Our empirical evidence in Tables 5, 6, and 21 shows that local perturbations that do not adversely affect the forecast-relevant semantics cause negligible variation in forecasting error. Conversely, for perturbations that alter the forecast relevant semantics (see Tables 6 and 21), we observe a significant change in the forecasting performance. Consequently, we note that this empirical result points only towards local smoothness in the loss space with respect to small perturbations in text annotations for reasonably good forecasters rather than the uniform Lipschitz condition currently stated in Assumption 4.2.

Overall, the theoretical bound on the risk in Theorem 4.4 remains conditional on these assumptions and should be interpreted as explaining when sim-to-real transfer is expected to hold, rather than guaranteeing it for arbitrary cases.

C Synthetic Toy Example Description

We construct a controlled, fully synthetic toy series of length $T = 1100$ daily steps that combines a SARIMA-like latent process with discrete, text-driven shocks. The latent baseline y_t^{base} is generated recursively from a linear trend, weekly seasonality, and an AR(1) autoregressive component centered on a fixed level:

$$y_t^{\text{base}} = \mu + s\,t + \phi\left(y_{t-1}^{\text{base}} - \mu\right) + A\sin\left(\frac{2\pi t}{P}\right) + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{21}$$

with parameters $\mu = 50$ (bias), $s = 0.008$ (linear trend slope), $\phi = 0.6$ (AR coefficient), seasonal amplitude $A = 3.0$ and periodicity $P = 7$, and $\sigma = 1.2$, initialized at $y_0^{\text{base}} = \mu$.

The text component is added through a curated catalog of 30 discrete events, each represented by a tuple $(t_s, t_e, \ell, \text{label}, \text{text})$ with start day t_s, end day t_e, signed sentiment level $\ell \in [-1, 1]$, a short tag, and a natural-language description of the event in an analyst style. The first 25 events are used for training and the last 5 are reserved as held-out test windows. From these events we construct a piecewise-constant sentiment signal Z_t for length T. Every K timestamps before the start and after the end of the event, Z_t linearly increases from 0 to ℓ and linearly decreases from ℓ to 0, respectively. During the event Z_t stays constant at ℓ. We set K to 4. Consequently, the observed series is

$$Y_t = y_t^{\text{base}} + \beta\, Z_t, \tag{22}$$

with text coupling strength $\beta = 8.0$, which makes the text-driven deviations clearly visible against the natural fluctuations of the base signal as described in Equation 21.

We first consider training on real and testing on real (TRTR). By training, we refer to regressing the parameters μ, s, ϕ, A, and β. We use the `least_squares` function from `scipy`. This allows us to obtain the estimate of the underlying model used for data generation. With this model, we simulate the held-out test horizon and compute the MAE for test event timestamps. During training, we do not utilize the true sentiment levels. Instead, we use Sentence Transformer to estimate the sentiment level.

	MAE (↓)	
Dataset	LLM Forecast ($\theta_{\text{LLM}}(X_t^H, S_t)$)	CoCo ($\theta_{\text{M-TSFM}}(X_t^H \| \| M_t \| \| C_t)$)
Raw	0.878	**0.858**
Rephrased	**0.875**	0.859

Table 5: **CoCo is less sensitive to text rephrasing.** We observe that the MAE of the LLM and the MAE of CoCo change marginally with rephrasing as long as the text conveys the same information. This validates **Assumption 4.2** (smoothness assumption) indicating that small perturbations in the context S_t do not significantly affect the forecasting performance.

Perturbation	% Change in the LLM Forecast ($\theta_{\text{LLM}}(X_t^H, S_t)$) MAE (↓)
Rephrased	0.3
Random Word Deletion (5%)	0.8
Less Verbose Annotations	1.1
Sentence Reordering	1.02
Using Annotations From Other Datasets	5.0
Misleading Text	7.9

Table 6: **CoCo exhibits local smoothness with respect to text perturbations.** For simple perturbations that do not affect the forecast trend indicators (first 4 rows), note that the change in the LLM forecast MAE is less than or around 1%, indicating local smoothness towards mild perturbations. However, for perturbations that change the underlying forecast (last two rows), the change is significant.

To showcase the effectiveness of training on synthetic annotations, we use LLMs to annotate the events, using the same approach described in Section 4. We extract sentiment levels from the generated synthetic annotations using the same approach as in TRTR and estimate μ, s, ϕ, A, and β. Consequently, we use these estimated parameters to simulate the test events with sentiments extracted from real text and compute the MAE (TSTR).

The results in Figure 3 show that the TRTR and TSTR MAEs are very similar, thereby establishing the veracity of our **sim-to-real** approach for multimodal forecasting, for this specific toy scenario.

D Evaluation on Context is Key (CiK)

Context is Key (CiK) [12] is a time series forecasting benchmark that pairs numerical data with diverse types of carefully crafted textual context, where crucially every task requires understanding textual context to be solved successfully. We evaluate on CiK to understand the behavior of CoCo's components in a setting that is deliberately adversarial to unimodal numerical forecasting. We first establish where Chronos2 stands on the CiK leaderboard as a unimodal baseline. Chronos2 achieves an average RCRPS of 0.265, outperforming Chronos-Large (0.326), TimeGEN (0.353), Moirai-Large (0.520), all statistical baselines (ARIMA: 0.475, ETS: 0.530), and the majority of LLM-based methods on the leaderboard. This is notable given that CiK is explicitly designed so that numerical history alone is insufficient.

Domain	Mean Distance Between Real and True Forecast Description Representation	Mean Distance Between Synthetic and True Forecast Description Representation	Δ Mean
All domains	0.3715	0.3774	+0.0059
Agriculture	0.3718	0.3735	+0.0017
Energy	0.3744	0.3745	+0.0001
Macro Indicators	0.3720	0.3805	+0.0085
Retail	0.3759	0.3744	-0.0015
Finance	0.3707	0.3775	+0.0068

Table 7: **Our synthetic annotations encode similar information about the true future as the real annotations.** We empirically show this in Figure 8. However, as Assumption 4.1 uses the Euclidean distance measure, we also provide a per-domain breakdown of the l_2 norm between the real and synthetic predicted representations and the representation of the true forecast description. Here, the representation space used is the unit normalized BERT embedding space.

Model	Average RCRPS ↓	Intemporal Information	Historical Information	Future Information	Covariate Information	Causal Information
Qwen3-4b (Base)	0.358 ± 0.031	0.390 ± 0.045	0.247 ± 0.023	0.301 ± 0.042	0.310 ± 0.035	0.700 ± 0.149
Qwen3-4b (CoCo Fine-tuned)	0.301 ± 0.027	0.306 ± 0.040	0.219 ± 0.019	0.276 ± 0.036	0.280 ± 0.033	0.675 ± 0.142
Qwen3-4b (Base) + Reasoning	0.189 ± 0.036	0.159 ± 0.011	0.134 ± 0.011	**0.088 ± 0.012**	0.204 ± 0.047	0.523 ± 0.214
Qwen3-4b (CoCo Fine-tuned) + Reasoning	**0.161 ± 0.013**	**0.154 ± 0.010**	**0.130 ± 0.012**	0.090 ± 0.012	**0.171 ± 0.016**	**0.378 ± 0.064**
Chronos2 (Unimodal)	0.265 ± 0.016	0.254 ± 0.021	0.166 ± 0.017	0.293 ± 0.031	0.239 ± 0.018	0.438 ± 0.066
CoCo (Fine-tuned + Reasoning)	0.256 ± 0.017	0.242 ± 0.023	0.137 ± 0.011	0.281 ± 0.029	0.234 ± 0.019	0.438 ± 0.074
GPT-4o (reference)	*0.191 ± 0.004*	*0.218 ± 0.007*	*0.118 ± 0.001*	*0.121 ± 0.001*	*0.143 ± 0.003*	*0.363 ± 0.011*
Llama-3.1-405B (reference)	*0.143 ± 0.006*	*0.174 ± 0.010*	*0.146 ± 0.001*	*0.075 ± 0.005*	*0.144 ± 0.008*	*0.303 ± 0.037*

Table 8: **RCRPS on the CiK benchmark [12]**. We provide a breakdown of RCRPS by information type, and evaluate the effect of fine-tuning with CoCo on the reasoning capacity of the LLM. **Bold** denotes the best result among our models per column. Reference numbers for GPT-4o and Llama-3.1-405B are from the official CiK leaderboard (`https://github.com/servicenow/context-is-key-forecasting`) and shown in *italics* for context. Lower is better. Our key takeaway from the results is that LLMs fine-tuned with CoCo consistently outperform their base, pre-trained counterpart. Additionally, a Qwen3-4B model that is fine-tuned with CoCo, when prompted for reasoning, can match the performance of powerful closed-source LLMs, such as GPT-4o.

Table 8 reports the average RCRPS (Region of Interest Continuous Ranked Probability Score) on CiK for Qwen3-4b under four configurations: base model without reasoning (as evaluated on CiK), base model with a short chain-of-thought reasoning prompt, CoCo fine-tuned model without reasoning, and CoCo fine-tuned model prompted for reasoning. The reasoning prompt used is:

> *"Before the forecast, provide a reasoning section between <reasoning> and </reasoning> tags."*

Adding a chain-of-thought reasoning prompt reduces RCRPS by 47% for the base model ($0.358 \rightarrow 0.189$) and by 46% for the fine-tuned model ($0.301 \rightarrow 0.161$). This improvement from a single prompt modification substantially exceeds the gain from fine-tuning alone in this setting. This suggests that CiK rewards explicit chain-of-thought reasoning. Fine-tuning provides complementary gains on top of reasoning. While the reasoning prompt dominates, fine-tuning still contributes meaningfully when combined with reasoning ($0.189 \rightarrow 0.161$, a further 15% improvement). The fine-tuned model with reasoning matches GPT-4o, despite being much smaller in parameter count. This confirms that GRPO fine-tuning for covariate generation teaches transferable forecasting capabilities that generalize to a held-out benchmark the model was never trained on.

Fundamentally, CiK differs from our main evaluation benchmarks in a key respect: every CiK task is constructed so that textual context is the *primary* source of predictive signal, with numerical history deliberately insufficient. In this regime, the TSFM backbone adds limited value. In contrast, our evaluation datasets contain naturally occurring noisy text where numerical history remains genuinely informative, the complementary setting that CoCo is designed for, and where CoCo consistently outperforms both the LLM and the TSFM in isolation.

E Training

E.1 Training Description

We fine-tune θ_{LLM} and $\theta_{\mathrm{M-TSFM}}$ in two sequential stages, as described in Section 5. Both stages use parameter-efficient LoRA adapters [64] so that the base weights remain frozen at their public checkpoints, and only the adapters are written to disk and exchanged across the training/inference processes.

Window Sampling and Normalization: Each training window is sampled by independently drawing a context length $l \in \{64, 96, 128, 192, 256\}$ and a forecast horizon $h \in \{16, 24, 32, 40\}$ uniformly at random, and then choosing a random valid start index in the source series. This exposes both stages to the full range of (l, h) combinations rather than a single fixed pair. Sampling longer histories is deliberately included even though Chronos2's unimodal accuracy improves rapidly with l. It forces the LLM to remain useful in the regime where the numerical baseline is already strong, and yields a forecast covariate M_t whose marginal benefit (Equation 5) does not collapse as l grows. For every sampled window, the numerical history $X_t^H = [x_{t-l+1}, \ldots, x_t]$ is z-score standardized using per-window statistics μ_t and σ_t (mean and standard deviation) computed over the context portion only, and the ground truth $X_t^F = [x_{t+1}, \ldots, x_{t+h}]$ is normalized with the same (μ_t, σ_t). The LLM is conditioned on raw (unscaled) values in the prompt but is supervised in the normalized space (i.e., the reward MAE in Section 5 is computed after applying (μ_t, σ_t) to the parsed forecast), while Chronos2 operates entirely in the per-window normalized space. The evaluation MAEs reported are in the normalized space, so that they are comparable across datasets. We observe that the relative performance of models does not change significantly for metrics computed in the de-normalized space as well (Table 19).

LLM Fine-Tuning: The base LLM θ_{LLM} is initialized from `Qwen3-4B-Instruct-2507` and adapted with rank-32 LoRA modules ($\alpha = 64$, dropout 0, no bias) attached to all attention and MLP projection matrices (q, k, v, o, gate, up, down) across the 36 transformer blocks, yielding $\sim$61.6M trainable parameters ($\sim 1.5\%$ of the $\sim$ 4B base). We optimize the GRPO objective of Section 5 with $N_{\mathrm{rollouts}} = 8$ rollouts per prompt sampled at temperature 1.0, a prompt batch size of 256, and AdamW at a learning rate of 10^{-6}, for up to 500 optimizer steps. The reward $r_t^{(g)}$ is computed against the *frozen* pretrained Chronos2 (`amazon/chronos-2`) and the resulting GRPO advantage $A_t^{(g)}$ updates only the LoRA adapters of θ_{LLM}. Training prompts are drawn at a 0.5/0.5 ratio from our synthetic corpus and FNSPID real-world annotations with a periodicity stratifier that balances periodic and aperiodic series. We run the training loop with PrimeRL [65], splitting GPUs into 2 trainers and 2 vLLM [66] inference workers (4 H100s).

Textual Context: Although the LLM prompt always includes the full numerical history X_t of length l, the textual annotation S_t is restricted to only the most recent $K_{\mathrm{train}} = 10$ timesteps during fine-tuning. Concretely, only the text fields associated with $\{x_{t-K_{\mathrm{train}}+1}, \ldots, x_t\}$ are inserted into the prompt's annotation block. Older annotations, if present, are discarded. We deliberately use a smaller K_{train} at training time than the $K_{\mathrm{eval}} = 16$ used at evaluation. Textual annotations are dense and verbose, and including all 16 steps in every prompt, in combination with the longest contexts $l = 256$, the $N_{\mathrm{rollouts}} = 8$ on-policy completions per GRPO update, and LoRA activations stored across transformer blocks, pushed the per-rollout sequence length above the 8,192-token budget that fits within the LoRA fine-tuning memory on our 2-trainer setup. Truncating to $K_{\mathrm{train}} = 10$ keeps the per-rollout sequence length comfortably under that budget while preserving the most causally relevant annotations. At inference time, this constraint disappears and we expand the textual window back to $K_{\mathrm{eval}} = 16$, matching the value used by all closed-source LLM baselines for a fair comparison.

Chronos2 Fine-Tuning: Once θ_{LLM} has converged, we pre-generate the covariate set $\{M_t, C_t\}_{t=1}^N$ from Equations 5 and 6 using $N_{\mathrm{rollouts}} = 5$ rollouts per sample for the confidence channel. The forecaster $\theta_{\mathrm{M-TSFM}}$ is initialized from `amazon/chronos-2` ($\sim$ 119.5M parameters, $d_{\text{model}} = 768$, $d_{\text{ff}} = 3072$, 12 encoder blocks) and adapted with rank-8 LoRA modules ($\alpha = 16$, dropout 0, no bias) attached to the self-attention q, k, v, o projections in every encoder block and to the output projection of the patch-embedding head, totaling $\sim$1.21M trainable parameters ($\sim 1.0\%$ of the base). The base weights are kept frozen and only the LoRA adapters are trained by minimizing $\frac{1}{N}\sum_{t=1}^{N} \left\|\theta_{\mathrm{M-TSFM}}(X_t^H \| M_t \| C_t) - X_t^F\right\|^2$ with AdamW for $10,000$ steps, with model selection

by validation MAE on a held-out 5% split. The fine-tuned adapters are merged with the base `amazon/chronos-2` weights only at inference time.

Noise Level for Covariate Construction. The forecast covariate M_t (Eq. 5) and confidence covariate C_t (Eq. 6) include i.i.d. noise $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ in the history portion to allow Chronos2 to interpret it as a noisy next-step predictor. Table 9 shows the effect of σ_ϵ on forecasting performance. Neither removing noise entirely ($\sigma_\epsilon = 0.00$) nor using high noise levels ($\sigma_\epsilon = 1.00$) yields the best performance. The former provides no signal diversity for Chronos2 to calibrate against, while the latter obscures the next-step information in the history portion. A small noise level of $\sigma_\epsilon = 0.05$ achieves the best MAE and is used throughout all experiments for both fine-tuning and inference.

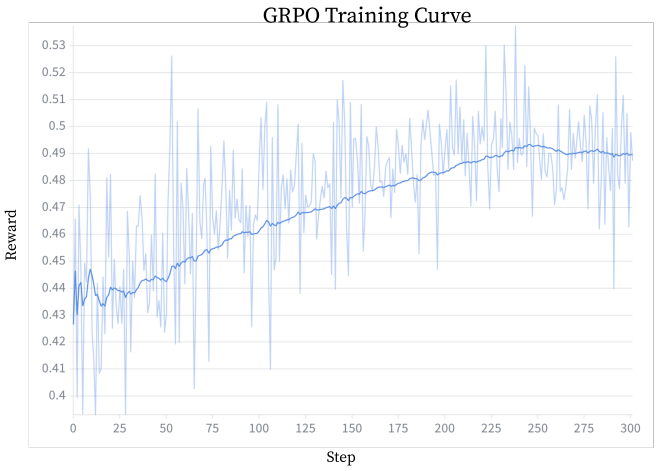


Figure 10: Average GRPO training reward r_t (Eq. 7) over training steps, computed as the mean over N_{rollouts} rollouts per prompt in a batch. The reward increases steadily reflecting that the LLM progressively learns to generate forecast covariates M_t that provide better forecasts ($\theta_{\text{M-TSFM}}(X_t^H || M_t)$) relative to the unimodal baseline ($\theta_{\text{M-TSFM}}(X_t^H)$).

σ_ϵ	CoCo MAE (↓)
0.00	0.810
0.05	**0.801**
0.25	0.802
0.50	0.805
1.00	0.814

Table 9: **Effect of noise standard deviation σ_ϵ on CoCo's forecasting performance.** σ_ϵ controls the noise added to the history portion of the forecast covariate M_t (Eq. 5) and the confidence covariate C_t (Eq. 6). Results are for 16-step forecasts with context length 64 on the Media Cloud dataset, using the base pretrained Chronos2 as the TSFM backbone. **Bold** denotes the best result. $\sigma_\epsilon = 0.05$ is selected and used in all experiments.

E.2 Training Datasets

For training, as described in Section 4, we annotate real-world time series datasets using a large language model (*Qwen3-30B-A3B-Instruct-2507*) with the prompt detailed in Section I.1. Specifically, we provide windows of 30 timestamps to the LLM for annotation in each batch for a dataset. Since we evaluate forecast horizons of up to 16 steps, this procedure yields annotations with varying degrees of predictive information. In particular, the first annotation within a window has access to the full future horizon and may therefore contain stronger predictive signals, whereas annotations closer to the end of the window are increasingly speculative and may carry little to no predictive power.

We focus on the macroeconomic domain, as the relevant datasets are publicly available, abundant, and well-studied. Importantly, macroeconomic time series are known to be influenced by textual

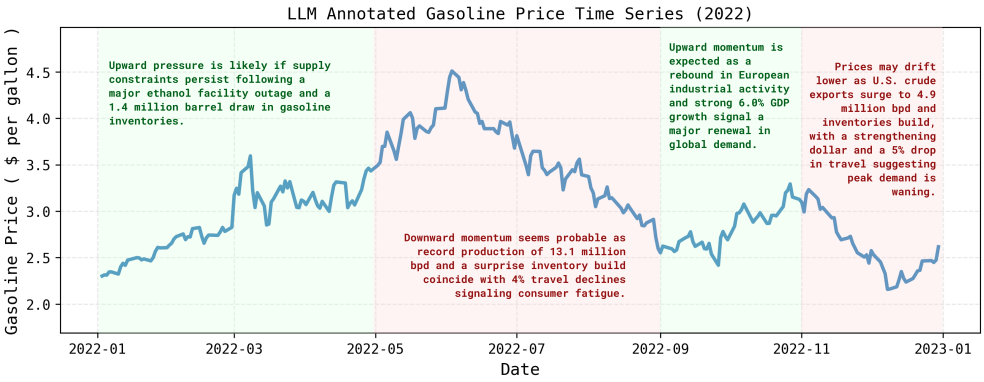


Figure 11: **Qwen3 provides realistic textual annotations.** Here, we show samples of annotations generated by *Qwen3-30b-A3B-Instruct-2507* [9] for gasoline prices in the United States for the year 2022. Note that the text generated is relevant to the domain queried and foreshadows the trend in the numerical data. In the plot, the green portion indicates an increasing trend and the red portion indicates a decreasing trend.

information from news and reports, making them a natural testbed for evaluating multimodal models that incorporate textual context. The primary sources of real-world time series used in our study are:

- **FRED**: A public-domain repository of domestic and international economic indicators maintained by the Federal Reserve Bank of St. Louis. We select around 1200 datasets from the domains included in the FRED-MD collection, which is curated for empirical analysis in economics and finance. These datasets span multiple frequencies and economic domains.
- **Other Sources**: We collect time series from two major European statistical agencies: Eurostat, the statistical office of the European Union; and Statistics Denmark (StatBank), Denmark's national statistical institute. From these sources, we collect time series datasets corresponding to various economic indicators, such as energy utility, unemployment rates, etc. These sources offer datasets at multiple frequencies (daily to monthly).

Source	Frequency	Datasets	Samples
FRED	Daily	95	107,993
	Weekly	91	97,010
	Monthly	807	433,750
	Quarterly	93	18,891
	Annual	52	3,176
EIA Data	Daily	118	252,581
	Monthly	1,596	428,449
DK Data	Monthly	26,766	5,314,577
EU Data	Monthly	7,001	1,471,900
Favorita	Daily	87	148,469
FNSPID (Real)	Daily	142	140,977
Air Quality	Hourly	120	4,135,865
Total		**36,968**	**12,553,638**

Table 10: Summary of the training datasets used. Datasets from FNSPID contain real annotations

A summary of the number of datasets generated from each source, with the frequency and the number of samples annotated, is provided in Table 10. Leveraging the synthetic annotation process, we were able to create a large-scale multimodal training dataset with more than 12 million samples, spanning over 36,968 target variables.

To mitigate memorization and unintended exposure to test sets, we enforce a strict date cutoff of December 2022 when creating the training dataset. While some existing multimodal datasets we use for evaluation contain samples from before this cutoff, the datasets used (summarized in Table 11) also consist of time series from diverse domains, which are substantially out-of-distribution relative to the datasets used during training. To further enforce a strict separation between training and evaluation and to explicitly address the knowledge cutoff issue of LLMs, we additionally curate a

new multimodal evaluation suite consisting exclusively of data from 2024–2025. This dataset enables evaluation under both temporal shift and domain mismatch, providing a test of generalization, and is described in detail in Section F.1.

Apart from the synthetic dataset generated, we also include a subset of the FNSPID dataset during training to evaluate the usefulness of synthetic annotations when compared to real-world text for training (results in Table 24). The top 150 stocks where the text availability is highest are chosen for training. Since FNSPID is also part of the evaluation set, we exclude the stocks in the evaluation set from the training set to prevent leakage during training.

Additionally, we use the Beijing Air Quality dataset [67] as part of real-world multimodal datasets used along with our synthetically annotated multimodal datasets. The dataset contains air quality metrics and environmental variables recorded at 12 different stations. For each metric/variable, we construct textual annotations using the remaining features. The dataset has time series samples with hourly frequency.

F Evaluation

F.1 Evaluation Datasets

A comprehensive summary of the datasets used for evaluation is provided in Table 11. Our study utilizes existing multimodal benchmarks, specifically the TimeMMD benchmark and stock prices from FNSPID. However, these datasets lack samples beyond 2023. Given that our methodology leverages the recent LLMs for forecasting, evaluating solely on legacy data introduces a risk of temporal data leakage, given that they have knowledge cutoffs of up to late 2024. To ensure a rigorous and "fair" evaluation in which the LLM has no prior exposure to test trends, we curated a new multimodal evaluation suite comprising datasets with samples following the 2024 cutoff. We show that CoCo's performance does not degrade when evaluated separately in samples after the cutoff date (Table 17).

TimeMMD We evaluate on eight TimeMMD datasets (spanning Agriculture, Climate, Economy, Energy, Security, Social Good and Traffic). While TimeMMD provides heterogeneous text sources (reports and search queries), we aggregate and summarize all available information for each timestamp to provide a unified context.

FNSPID We include stock price data and timestamped NASDAQ news for a subset of the S&P 500's top 25 companies from different sectors. To handle the high volume of daily articles, we summarize multiple news entries into a single, daily textual summary.

Media Cloud For the curation of the text for the new evaluation datasets, we leveraged the Media Cloud API[3] for news articles with timestamps and used the pipeline shown in Figure 12 to obtain text annotations on a daily basis. The LLM prompts used in the pipeline are provided in I.3. We also provide the Media Cloud queries used to fetch news articles from the API for each dataset.

Target variables were aggregated from three primary sources:

- **Google Finance**[4]: Daily market pricing and equity data were retrieved using Google Finance, via the Google Sheets integration. This source provides broad accessibility to global exchange data. Specific tickers are provided to ensure exact dataset reconstruction.
- **FRED**: Macroeconomic indices and benchmarks were sourced from the FRED database, a public-domain repository for domestic and international economic indicators maintained by the Federal Reserve Bank of St. Louis.
- **Trading Economics**[5]: Commodity prices were obtained from the Trading Economics platform. This source provides curated global economic data. Researchers seeking to replicate this specific subset may require a platform subscription.

Below are the specific tickers and Media Cloud queries used for the curation:

- **S&P500 Index**

[3] https://search.mediacloud.org/search

[4] https://newsinitiative.withgoogle.com/resources/trainings/access-google-finance-data-in-sheets/

[5] https://tradingeconomics.com

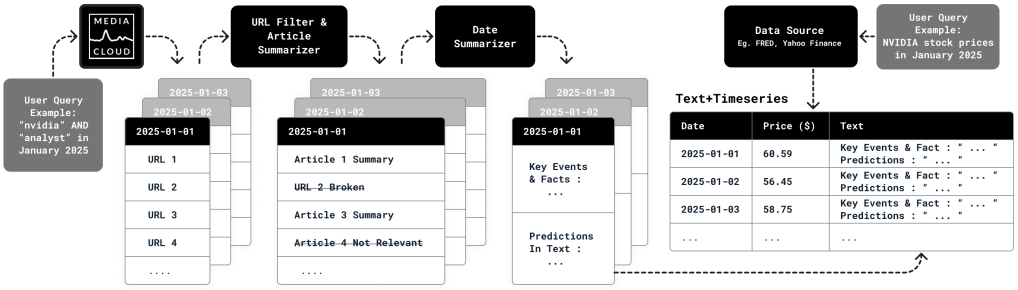


Figure 12: **Our approach to generate textual annotations, for real-world time series datasets, extracted from the news articles that are scraped from the internet.** Let us consider the task of generating textual annotations for the NVIDIA stock price dataset. First, we use the **Media Cloud** API to extract the relevant news articles between two successive timestamps $t-1$ and t. These news articles are summarized into *Key Events and Facts* and *Predictive Information* by an LLM. This summary is treated as the text annotation for the timestamp t.

  - Source : FRED - SP500[6]
  - Media Cloud Query : *"s&p 500" OR s&p500*
- **Crude Oil Price**
  - Source : FRED - DCOILWTICO[7]
  - Media Cloud Query : *"crude oil" OR wti OR brent*
- **NASDAQ Electricity Benchmark**
  - Source : FRED - NASDAQNQUSB651010[8]
  - Media Cloud Query : *nasdaq AND (electricity OR energy) AND sector*
- **NVIDIA Stock Price**
  - Source : Google Finance, Ticker: *NASDAQ:NVDA*
  - Media Cloud Query : *nvidia AND analyst*
- **SPDR Gold Shares Price**
  - Source : Google Finance, Ticker: *NYSEARCA:GLD*
  - Media Cloud Query : *gold AND analyst*
- **Egg Prices**
  - Source : Trading Economics - Eggs US[9]
  - Media Cloud Query : *egg AND (sales OR price)*
- **Apple Stock Price**
  - Source : Google Finance, Ticker: *NASDAQ:AAPL*
  - Media Cloud Query : *apple AND tech AND analyst*
- **Silver Price**
  - Source : Trading Economics - Silver[10]
  - Media Cloud Query : *silver AND (sales OR price)*
- **Steel Rebar Futures Price**
  - Source : Trading Economics - Steel Rebar Futures[11]
  - Media Cloud Query : *steel AND (NOT amazon) AND (price OR sale OR export OR import)*
- **Google Stock Price**

[6] https://fred.stlouisfed.org/series/SP500
[7] https://fred.stlouisfed.org/series/DCOILWTICO
[8] https://fred.stlouisfed.org/series/NASDAQNQUSB651010
[9] https://tradingeconomics.com/commodity/eggs-us
[10] https://tradingeconomics.com/commodity/silver
[11] https://tradingeconomics.com/commodity/steel

Text Source	Dataset	Target Source	Frequency	Date Range	Total	≥2024-06
Media Cloud	Apple Stock Price	Google Finance	Daily	2024-01 – 2025-12	503	398
	Copper Futures Price	Google Finance	Daily	2025-01 – 2026-01	258	258
	Crude Oil Price	FRED	Daily	2024-01 – 2026-01	500	395
	Egg Price	Trading Economics	Daily	2025-02 – 2025-12	223	223
	Google Stock Price	Google Finance	Daily	2024-01 – 2025-12	502	398
	NASDAQ Electricity	FRED	Daily	2024-01 – 2025-12	523	413
	NVIDIA Stock Price	Google Finance	Daily	2024-01 – 2025-12	502	397
	S&P500 Index	FRED	Daily	2024-01 – 2025-10	463	359
	SPDR Gold Shares Price	Google Finance	Daily	2024-01 – 2025-12	501	396
	Silver Price	Trading Economics	Daily	2024-01 – 2026-01	519	411
	Steel Rebar Futures Price	Trading Economics	Daily	2025-01 – 2026-01	241	241
TimeMMD	Broiler Price (Agriculture)	TimeMMD	Monthly	1980-01 – 2024-04	532	0
	Drought Level (Climate)	TimeMMD	Weekly	2000-01 – 2024-05	1,272	0
	US Trade Balance (Economy)	TimeMMD	Monthly	1987-01 – 2024-03	447	0
	Gasoline Price (Energy)	TimeMMD	Weekly	1993-04 – 2024-04	1,622	0
	Air Quality (Environment)	TimeMMD	Daily	2019-09 – 2023-09	1,480	0
	Disaster Grant (Security)	TimeMMD	Monthly	1998-09 – 2024-05	309	0
	Unemployment Rate (Social Good)	TimeMMD	Monthly	1980-01 – 2024-04	532	0
	Traffic Volume (Traffic)	TimeMMD	Monthly	1979-12 – 2024-03	532	0
FNSPID	Bank of America Stock Price	FNSPID	Daily	2018-06 – 2020-06	499	0
	Boeing Stock Price	FNSPID	Daily	2021-12 – 2023-12	500	0
	Costco Stock Price	FNSPID	Daily	2021-12 – 2023-12	500	0
	Exxon Mobil Stock Price	FNSPID	Daily	2021-12 – 2023-12	500	0
	Goldman Sachs Stock Price	FNSPID	Daily	2021-12 – 2023-12	500	0
	JPMC Stock Price	FNSPID	Daily	2018-09 – 2020-06	437	0
	McD Stock Price	FNSPID	Daily	2018-06 – 2020-06	500	0
	Microsoft Stock Price	FNSPID	Daily	2022-04 – 2023-12	412	0
	Oracle Stock Price	FNSPID	Daily	2021-12 – 2023-12	500	0
	Tesla Stock Price	FNSPID	Daily	2022-05 – 2023-12	410	0
	Walmart Stock Price	FNSPID	Daily	2021-12 – 2023-12	500	0
Total					**16,719**	**3,889**

Table 11: Summary of the multimodal evaluation datasets grouped by source.

  - Source : Google Finance, Ticker: *NASDAQ:GOOG*
  - Media Cloud Query : *(google OR alphabet) AND (analyst)*

- **Copper Futures Price**
  - Source : Google Finance, Ticker: *COMEX:HGW00*
  - Media Cloud Query : *copper AND (NOT amazon) AND (price OR sale OR export OR import)*

F.2 Evaluation Baselines

We organize the 27 baseline configurations according to whether they incorporate textual context and whether their parameters are fine-tuned. This categorization is also used in Figure 5.

1. **Zero-shot unimodal (8).** Chronos2 [8], TimesFM2.5 [7], TiRex [68], Moirai [69], Toto [70], TabPFN2.5 [51], TimeMoE [71], and Panda [72]. These methods forecast directly from the numerical history without textual context or task-specific fine-tuning.

2. **Zero-shot multimodal (7).** GPT-4o, Qwen3-30B, and TimeOmni directly forecast from the numerical history and paired text. Chronos2 MV and Toto MV receive Sentence Transformer embeddings of the text as additional covariates. We also include Chronos2+GPT-4o and TimesFM2.5+GPT-4o mean ensembles [73], which average the corresponding TSFM and LLM forecasts.

3. **Unimodal fine-tuning (6).** Chronos2, PatchTST [74], iTransformer [75], AutoTimes [76], S2IP-LLM [77], and TEST [78] are fine-tuned using only the numerical histories from the synthetic training data.

4. **Multimodal fine-tuning (6).** Chronos2 MV, AutoTimes, VoT [79], TaTS [40], S2IP-LLM, and TEST incorporate both numerical history and textual context during fine-tuning and inference. Where a method was originally unimodal, we adapt its input representation to incorporate text.

Many fine-tuning baselines were originally designed for dataset-specific training. To ensure a controlled comparison, we instead train every fine-tuned configuration using the same synthetic text–time-series pairs used to train CoCo and evaluate it on the same held-out real-world datasets.

F.3 Evaluation Metrics

Normalization: As described in Appendix E.1, we apply z-score normalization during inference, and all reported MAE values are computed in the normalized space.

MASE: Along with the MAE, we also report Mean Absolute Scaled Error (MASE) (Table 18) as a scale-free accuracy metric. For a forecast $[\hat{x}_{t+1}, \hat{x}_{t+2}, \ldots, \hat{x}_{t+h}]$ and ground-truth $[x_{t+1}, x_{t+2}, \ldots, x_{t+h}]$ over horizon length h, we compute

$$\mathrm{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n} |x_t - \hat{x}_t|}{\frac{1}{n-1}\sum_{t=2}^{n} |x_t - x_{t-1}|}.$$

The numerator is the model MAE over the forecast window, while the denominator is a naive lag-1 scaling term computed on the same window. Lower is better; values < 1 indicate performance better than the naive scale baseline. This follows the MASE usage in the **GIFT-Eval** benchmark for comparable, scale-normalized time-series evaluation across heterogeneous datasets. The aggregate MASE over datasets is computed by taking the geometric mean across datasets as per convention.

Wilcoxon Signed-Rank p-value: For each model pair, we form paired per-setting errors and run the two-sided Wilcoxon signed-rank test. The test ranks absolute paired differences and uses their signs, so it reflects both effect size and direction consistency. The null hypothesis is that the median paired difference is zero (no systematic difference). Small p-values indicate stronger evidence of a systematic difference while mixed positive and negative differences typically yield larger p-values.

ELO Ratings: We rank forecasting models across datasets, context lengths, and forecast horizons using an ELO rating system [80], which estimates latent skill levels from relative outcomes. Each evaluation setting induces a multiplayer matchup among all models, converted into relative outcomes based on the MAE (lower values indicate better performance). Model ratings are updated after each matchup using the standard ELO update rule:

$$R_i' = R_i + K \cdot (S_i - E_i), \tag{23}$$

where R_i and R_i' denote the pre- and post-match ratings of model i, S_i is the observed score derived from its rank in the matchup, and E_i is the expected score computed from all participating models' current ratings using a logistic model with deviation parameter $D = 400$. All models are initialized with a base rating of 1500, and $K = 32$ is the update factor controlling sensitivity.

The conventional ELO system is sensitive to the order of matchups and can become unstable as the number of models and evaluation settings increases [81]. To mitigate this, we repeat the rating procedure across 32 random orderings of the matchups and report the mean rating for each model. This approach yields stable ratings at our scale of models and settings. This evaluation protocol is in the same spirit as rating-based leaderboards used for LLMs [82], which employ ELO-style updates to aggregate performance comparisons across many evaluation settings rather than relying on simple averages of ranks or win counts.

G CoCo's Robustness to Stochasticity

We evaluate the stability of CoCo across all 95 dataset–setting combinations under three sources of stochasticity: (1) LLM rollout sampling, (2) noise injection into the historical covariates, and (3) stochasticity from LLM fine-tuning checkpoints.

LLM rollout stochasticity: We generate LLM forecasts using three different random seeds, with five rollouts per seed at temperature 1.0, resulting in 15 rollouts per sample. While individual rollout forecasts exhibit substantial variation, aggregating the rollouts using their median substantially reduces this variability. Across the three runs, the LLM rollout median has a coefficient of variation (CV) of only 0.56%, with the CV remaining below 2% for most dataset–setting combinations. More importantly, this residual variation is effectively absorbed by the TSFM: CoCo's final forecast has a CV of only 0.01%. The corresponding aggregate MAE statistics are shown in Table 12.

Forecast	MAE (mean $\pm$ std)
LLM Rollout ($\times 15$)	0.7787 ± 0.0489
LLM Rollout Median ($\times 3$)	0.7860 ± 0.0044
CoCo ($\times 3$)	0.7194 ± 0.0001

Table 12: **CoCo's robustness to LLM rollout stochasticity.** Overall MAE across all 95 settings, evaluated across three random seeds. Although individual LLM rollouts exhibit substantial variation, taking the median across five rollouts substantially reduces the variability, and the subsequent CoCo forecast is highly stable.

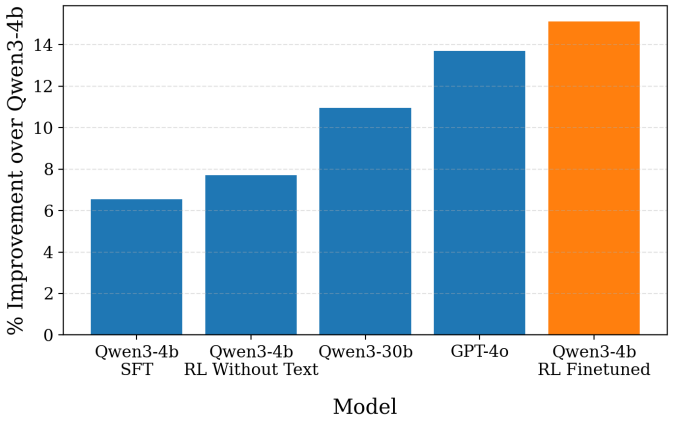


Figure 13: **Comparison of standalone forecasting performance of LLMs.** The plot shows the %-improvement in the MAE over the base Qwen3-4B model across the same 38 settings as Table 22. We observe that the CoCo fine-tuned Qwen3-4B outperforms larger and closed-source LLMs as a standalone forecaster.

Robustness to covariate noise: CoCo additionally injects Gaussian noise with $\sigma = 0.05$ into the historical covariates during Chronos2 inference as a regularization mechanism. To isolate the effect of this stochasticity, we evaluate CoCo using five random seeds while keeping the LLM rollouts fixed. CoCo achieves an overall MAE of 0.7194 ± 0.0005, corresponding to a CV of only 0.06%. Thus, the small amount of noise introduced during inference does not materially affect the final forecast.

Fine-tuning stochasticity: Finally, we evaluate the sensitivity of CoCo to stochasticity during LLM fine-tuning. We train the LoRA adapter for Qwen3-4B using three different training seeds, producing three independent checkpoints. At temperature 0, the resulting LLM forecasts exhibit noticeable variation, with an MAE of 0.818 ± 0.009 (CV = 1.05%). In contrast, the corresponding CoCo forecasts remain highly consistent, achieving an MAE of 0.7194 ± 0.0001 (CV = 0.01%). This demonstrates that CoCo is not sensitive to the particular LLM fine-tuning checkpoint used.

Overall, CoCo remains highly robust across all three sources of stochasticity. We primarily attribute this stability to the use of the median across multiple LLM rollouts, which substantially suppresses variability before the resulting covariate is provided to the TSFM. The TSFM further absorbs the small amount of residual variation, resulting in a highly stable final forecast.

H Extended Results

		Mean MAE (↓)								
Group	**Dataset**	**CoCo**	**TiRex1**	**Chronos2**	**Moirai**	**TimesFM2.5**	**Toto**	**TabPFN2.5**	**TimeMoE**	**PANDA**
Media Cloud	Crude Oil Price	0.700	0.682	0.690	0.687	**0.663**	0.703	0.777	0.770	0.774
	S&P500 Index	**0.594**	0.601	0.641	0.625	0.652	0.667	0.778	0.835	0.816
	Apple Stock	**0.637**	0.666	0.666	0.651	0.676	0.683	0.760	0.797	0.846
	Copper Futures	0.682	0.735	0.727	**0.679**	0.751	0.701	0.781	0.775	0.941
	Egg Price	0.604	0.587	0.583	0.607	0.587	**0.583**	0.655	0.598	0.865
	Electricity Price	**0.576**	0.613	0.605	0.608	0.592	0.625	0.710	0.707	0.751
	SPDR Gold Price	**0.547**	0.576	0.591	0.582	0.642	0.626	0.682	0.821	0.832
	Google Stock	0.606	**0.600**	0.624	0.627	0.636	0.640	0.659	0.809	0.790
	NVIDIA Stock Price	**0.613**	0.626	0.634	0.633	0.655	0.659	0.708	0.762	0.785
	Silver Price	**0.699**	0.725	0.753	0.737	0.758	0.738	0.899	0.926	0.964
	Steel Rebar Futures	0.607	0.626	0.633	0.640	0.642	**0.592**	0.685	0.667	0.748
FNSPID	Boeing Stock	**0.585**	0.587	0.598	0.609	0.611	0.617	0.622	0.689	0.728
	Bank of America Stock	**0.820**	0.826	0.850	0.843	0.831	0.871	0.890	0.941	1.005
	Costco Stock	**0.567**	0.569	0.583	0.604	0.584	0.602	0.672	0.700	0.728
	Goldman Sachs Stock	0.743	0.741	0.743	0.767	0.725	0.767	0.731	**0.719**	0.854
	JPMC Stock	**0.799**	0.804	0.837	0.816	0.849	0.861	0.876	1.040	1.008
	McDonald's Stock	0.684	0.684	0.712	**0.681**	0.736	0.712	0.742	0.806	0.857
	Microsoft Stock	**0.596**	0.610	0.646	0.625	0.645	0.654	0.712	0.801	0.830
	Oracle Stock	0.594	**0.582**	0.600	0.624	0.626	0.648	0.675	0.781	0.807
	Tesla Stock	0.566	0.577	0.593	0.562	**0.559**	0.610	0.587	0.634	0.709
	Walmart Stock	**0.591**	0.616	0.615	0.634	0.633	0.641	0.668	0.716	0.765
	Exxon Mobil Stock	0.690	**0.664**	0.684	0.706	0.730	0.712	0.772	0.731	0.766
TimeMMD	Broiler Price	**0.653**	0.674	0.705	0.682	0.735	0.712	0.809	0.853	0.934
	Drought Level	**0.689**	0.697	0.712	0.718	0.715	0.730	0.740	0.759	0.821
	US Trade Balance	**0.650**	0.705	0.675	0.667	0.655	0.716	0.727	0.813	0.844
	Gasoline Price	**0.671**	0.680	0.697	0.707	0.696	0.723	0.775	0.766	0.855
	Air Quality Index	0.837	**0.822**	0.830	0.849	0.826	0.834	0.849	0.856	0.897
	Disaster Grants	**1.450**	1.577	1.547	1.539	1.580	1.559	1.465	1.522	1.588
	Unemployment Rate	0.550	**0.531**	0.565	0.596	0.555	0.544	0.634	0.669	0.733
	Travel Volume	0.264	0.332	0.270	0.316	**0.252**	0.271	0.292	0.315	0.602
TimeMMD Daily	Air Quality Index	0.974	**0.928**	0.945	0.957	0.933	0.948	0.963	1.070	0.969
TimeMMD Weekly	Drought Level	**1.040**	1.063	1.102	1.062	1.079	1.081	1.111	1.186	1.496
	Gasoline Price	1.041	**1.038**	1.099	1.091	1.093	1.096	1.098	1.189	1.648
TimeMMD Monthly	Broiler Price	**1.652**	1.682	1.681	1.712	1.707	1.774	1.715	1.752	1.671
	US Trade Balance	1.356	1.436	1.390	1.522	1.371	1.343	1.443	**1.335**	1.450
	Disaster Grants	1.610	**1.604**	1.651	1.662	1.627	1.634	1.631	1.642	1.758
	Unemployment Rate	**1.539**	1.588	1.539	1.635	1.601	1.617	1.565	1.599	1.574
	Travel Volume	1.419	1.476	1.485	1.387	1.325	1.372	1.593	**1.254**	1.425
	Overall Wins	**38**	25	0	6	13	2	4	6	1
	Mean	**0.719**	0.735	0.745	0.747	0.749	0.758	0.800	0.839	0.903
	ELO	**1820**	1715	1602	1582	1593	1487	1344	1266	1090
	Average Rank	**2.24**	2.98	4.07	4.31	4.17	5.22	6.52	7.19	8.31
	Head-to-Head Wins	–	60/95	81/95	78/95	72/95	80/95	90/95	88/95	93/95
	Wilcoxon p	–	5.0e-04	2.0e-12	9.2e-13	7.8e-09	8.1e-13	1.4e-16	3.3e-15	4.1e-17

Table 13: **Dataset-level MAE for zero-shot unimodal methods.** Each dataset reports the mean over the 64-16, 128-16, and 256-16 settings. TimeMMD Daily, Weekly, and Monthly retain their native settings as separate blocks. Summary statistics are computed across all 95 settings. CoCo outperforms multiple state-of-the-art TSFMs. Bold and underlined values denote the best and second-best method in each row.

		Mean MAE (↓)							
Group	**Dataset**	**CoCo**	**TimesFM2.5+GPT4o**	**Chronos2+GPT4o**	**Chronos2 MV**	**Toto MV**	**GPT-4o**	**Qwen3-30B**	**TimeOmni**
Media Cloud	Crude Oil Price	0.700	**0.625**	0.645	0.707	0.646	0.713	0.842	0.887
	S&P500 Index	**0.594**	0.634	0.649	0.693	0.724	0.723	0.655	0.785
	Apple Stock	**0.637**	0.682	0.692	0.715	0.706	0.770	0.754	0.876
	Copper Futures	**0.682**	0.685	0.685	0.748	0.740	0.725	0.736	0.853
	Egg Price	0.604	0.579	**0.572**	0.594	0.577	0.599	0.672	0.809
	Electricity Price	**0.576**	0.628	0.641	0.631	0.646	0.719	0.668	0.851
	SPDR Gold Price	0.547	0.568	**0.546**	0.660	0.676	0.582	0.650	3.468
	Google Stock	**0.606**	0.619	0.619	0.669	0.668	0.680	0.772	0.726
	NVIDIA Stock Price	**0.613**	0.658	0.658	0.683	0.688	0.758	0.885	0.734
	Silver Price	**0.699**	0.706	0.714	0.830	0.823	0.784	0.781	0.891
	Steel Rebar Futures	0.607	0.603	0.610	0.584	**0.574**	0.695	0.787	1.375
FNSPID	Boeing Stock	**0.585**	0.588	0.591	0.597	0.620	0.620	0.737	0.764
	Bank of America Stock	0.820	**0.802**	0.820	0.860	0.870	0.862	0.889	1.006
	Costco Stock	**0.567**	0.567	0.577	0.621	0.629	0.627	0.632	0.713
	Goldman Sachs Stock	0.743	0.756	0.774	**0.689**	0.733	0.866	0.834	0.879
	JPMC Stock	0.799	**0.763**	0.768	0.865	0.862	0.780	0.840	1.077
	McDonald's Stock	0.684	**0.645**	0.670	0.691	0.745	0.695	0.878	0.875
	Microsoft Stock	**0.596**	0.607	0.615	0.678	0.694	0.632	0.626	0.662
	Oracle Stock	**0.594**	0.635	0.634	0.651	0.658	0.714	0.689	0.744
	Tesla Stock	0.566	0.565	0.583	**0.552**	0.585	0.625	0.635	0.628
	Walmart Stock	**0.591**	0.642	0.638	0.634	0.639	0.717	0.663	0.722
	Exxon Mobil Stock	0.690	0.687	**0.679**	0.718	0.725	0.741	0.795	0.826
TimeMMD	Broiler Price	**0.653**	0.669	0.663	0.786	0.753	0.725	0.779	1.394
	Drought Level	**0.689**	0.699	0.705	0.721	0.716	0.770	0.770	0.841
	US Trade Balance	**0.650**	0.697	0.696	0.703	0.662	0.900	1.153	1.436
	Gasoline Price	0.671	**0.667**	0.671	0.714	0.701	0.710	0.683	0.828
	Air Quality Index	0.837	0.917	0.921	**0.833**	0.833	1.126	1.158	1.146
	Disaster Grants	**1.450**	1.640	1.629	1.552	1.567	1.847	1.764	2.908
	Unemployment Rate	0.550	**0.461**	0.463	0.571	0.546	0.522	0.637	0.786
	Travel Volume	**0.264**	0.434	0.430	0.271	0.271	0.690	0.812	21.457
TimeMMD Daily	Air Quality Index	0.974	1.516	1.535	**0.943**	0.947	2.506	1.910	1.275
TimeMMD Weekly	Drought Level	1.040	**1.035**	1.055	1.112	1.081	1.101	1.100	1.225
	Gasoline Price	1.041	**1.009**	1.021	1.094	1.133	1.042	1.032	1.237
TimeMMD Monthly	Broiler Price	**1.652**	1.670	1.697	1.657	1.772	1.797	2.214	2.170
	US Trade Balance	1.356	1.399	1.423	1.347	**1.316**	1.515	1.700	2.373
	Disaster Grants	**1.610**	1.663	1.674	1.639	1.611	1.767	1.773	1.981
	Unemployment Rate	1.539	**1.400**	1.423	1.536	1.579	1.422	1.610	1.691
	Travel Volume	1.419	1.386	1.466	1.403	**1.344**	1.538	1.712	2.031
	Overall Wins	**41**	25	7	13	9	0	0	0
	Mean	**0.719**	0.742	0.748	0.763	0.766	0.836	0.878	1.756
	ELO	**1795**	1744	1676	1549	1528	1365	1289	1053
	Average Rank	**2.23**	2.64	3.14	4.17	4.34	5.65	6.25	7.58
	Head-to-Head Wins	–	60/95	65/95	74/95	75/95	85/95	94/95	95/95
	Wilcoxon p	–	0.005	6.3e-05	1.8e-10	6.1e-10	8.9e-15	2.8e-17	2.6e-17

Table 14: **Dataset-level MAE for zero-shot multimodal methods.** Each dataset reports the mean over the 64-16, 128-16, and 256-16 settings. TimeMMD Daily, Weekly, and Monthly retain their native settings as separate blocks. Summary statistics are computed across all 95 settings. Mean ensembles are included because they combine zero-shot forecasts. CoCo outperforms state-of-the-art LLM forecasters as well as multivariate TSFMs that directly incorporate Sentence Transformer text embeddings as covariates. Bold and underlined values denote the best and second-best method in each row.

		Mean MAE (↓)						
Group	**Dataset**	**CoCo**	**Chronos2 (FT)**	**AutoTimes**	**PatchTST**	**S2IP-LLM**	**iTransformer**	**TEST**
Media Cloud	Crude Oil Price	0.700	**0.681**	0.691	0.707	0.723	0.763	0.769
	S&P500 Index	**0.594**	0.660	0.652	0.626	0.661	0.740	0.764
	Apple Stock	**0.637**	0.683	0.669	0.683	0.698	0.772	0.827
	Copper Futures	**0.682**	0.729	0.731	0.717	0.776	0.779	0.767
	Egg Price	**0.604**	0.623	0.705	0.698	0.774	0.765	0.805
	Electricity Price	0.576	**0.560**	0.603	0.603	0.614	0.627	0.667
	SPDR Gold Price	**0.547**	0.577	0.606	0.587	0.600	0.633	0.680
	Google Stock	**0.606**	0.643	0.663	0.651	0.685	0.743	0.789
	NVIDIA Stock Price	**0.613**	0.659	0.667	0.651	0.683	0.738	0.759
	Silver Price	**0.699**	0.722	0.762	0.719	0.723	0.764	0.809
	Steel Rebar Futures	0.607	0.602	**0.550**	0.658	0.652	0.679	0.707
FNSPID	Boeing Stock	**0.585**	0.605	0.613	0.618	0.618	0.758	0.716
	Bank of America Stock	**0.820**	0.848	0.855	0.862	0.871	0.938	0.998
	Costco Stock	**0.567**	0.593	0.595	0.576	0.597	0.651	0.629
	Goldman Sachs Stock	0.743	0.737	**0.713**	0.763	0.764	0.819	0.824
	JPMC Stock	**0.799**	0.812	0.827	0.815	0.849	0.891	1.000
	McDonald's Stock	**0.684**	0.709	0.700	0.739	0.718	0.852	0.809
	Microsoft Stock	**0.596**	0.613	0.638	0.616	0.620	0.685	0.761
	Oracle Stock	0.594	**0.589**	0.619	0.622	0.604	0.697	0.690
	Tesla Stock	**0.566**	0.572	0.571	0.598	0.608	0.681	0.722
	Walmart Stock	**0.591**	0.631	0.615	0.634	0.653	0.699	0.762
	Exxon Mobil Stock	0.690	0.736	**0.670**	0.701	0.706	0.752	0.815
TimeMMD	Broiler Price	**0.653**	0.668	0.685	0.692	0.691	0.729	0.800
	Drought Level	**0.689**	0.725	0.709	0.743	0.775	0.818	0.868
	US Trade Balance	0.650	**0.620**	0.813	0.709	0.693	0.678	0.796
	Gasoline Price	**0.671**	0.696	0.734	0.732	0.757	0.807	0.840
	Air Quality Index	0.837	**0.831**	0.869	0.922	0.893	0.930	0.985
	Disaster Grants	**1.450**	1.462	1.525	1.538	1.488	1.537	1.546
	Unemployment Rate	0.550	**0.501**	0.740	0.689	0.647	0.810	0.794
	Travel Volume	0.264	**0.244**	0.478	0.474	0.444	0.466	0.747
TimeMMD Daily	Air Quality Index	0.974	**0.935**	0.971	1.024	0.990	1.058	1.124
TimeMMD Weekly	Drought Level	1.040	1.072	**1.035**	1.061	1.091	1.113	1.239
	Gasoline Price	**1.041**	1.065	1.111	1.089	1.173	1.120	1.315
TimeMMD Monthly	Broiler Price	**1.652**	1.725	1.799	1.685	1.767	1.684	2.144
	US Trade Balance	1.356	1.478	**1.294**	1.300	1.343	1.316	1.589
	Disaster Grants	**1.610**	1.621	1.636	1.616	1.630	1.613	1.980
	Unemployment Rate	1.539	**1.539**	1.619	1.551	1.622	1.542	1.663
	Travel Volume	1.419	1.680	**1.230**	1.329	1.286	1.387	1.665
	Overall Wins	**52**	23	16	1	3	0	0
	Mean	**0.719**	0.738	0.763	0.764	0.773	0.823	0.882
	ELO	**1860**	1697	1595	1549	1472	1249	1078
	Average Rank	**1.69**	2.63	3.40	3.74	4.25	5.71	6.58
	Head-to-Head Wins	–	64/95	78/95	89/95	87/95	92/95	94/95
	Wilcoxon p	–	1.2e-05	3.2e-10	7.2e-15	3.5e-15	6.1e-17	2.7e-17

Table 15: **Dataset-level MAE for unimodal fine-tuning methods.** Each dataset reports the mean over the 64-16, 128-16, and 256-16 settings. TimeMMD Daily, Weekly, and Monthly retain their native settings as separate blocks. Summary statistics are computed across all 95 settings. CoCo outperforms unimodal fine-tuning approaches, indicating that exogenous textual context is essential for improved forecasting and is difficult to recover from numerical history alone. Bold and underlined values denote the best and second-best method in each row.

		Mean MAE (↓)						
Group	**Dataset**	**CoCo**	**Chronos2 MV (FT)**	**VoT**	**TaTS**	**S2IP-LLM**	**AutoTimes**	**TEST**
Media Cloud	Crude Oil Price	0.700	0.680	**0.667**	0.698	0.715	0.699	0.704
	S&P500 Index	**0.594**	0.641	0.679	0.694	0.666	0.682	0.727
	Apple Stock	**0.637**	0.657	0.661	0.702	0.698	0.730	0.699
	Copper Futures	0.682	**0.679**	0.707	0.730	0.755	0.755	0.732
	Egg Price	0.604	0.667	0.709	**0.592**	0.737	0.781	0.749
	Electricity Price	**0.576**	0.587	0.596	0.623	0.601	0.617	0.622
	SPDR Gold Price	**0.547**	0.561	0.642	0.646	0.603	0.661	0.660
	Google Stock	**0.606**	0.622	0.641	0.668	0.711	0.707	0.671
	NVIDIA Stock Price	**0.613**	0.656	0.662	0.672	0.670	0.679	0.682
	Silver Price	**0.699**	0.726	0.775	0.810	0.746	0.828	0.798
	Steel Rebar Futures	0.607	0.608	0.585	**0.567**	0.583	0.567	0.608
FNSPID	Boeing Stock	**0.585**	0.617	0.613	0.631	0.665	0.665	0.704
	Bank of America Stock	0.820	**0.806**	0.856	0.843	0.864	0.915	0.928
	Costco Stock	0.567	**0.566**	0.588	0.613	0.614	0.620	0.670
	Goldman Sachs Stock	0.743	0.741	0.725	**0.684**	0.765	0.727	0.793
	JPMC Stock	0.799	**0.796**	0.813	0.841	0.820	0.892	0.895
	McDonald's Stock	0.684	**0.663**	0.685	0.695	0.703	0.728	0.737
	Microsoft Stock	**0.596**	0.608	0.623	0.669	0.638	0.679	0.683
	Oracle Stock	0.594	**0.592**	0.612	0.634	0.644	0.671	0.669
	Tesla Stock	0.566	0.586	0.565	**0.552**	0.664	0.595	0.676
	Walmart Stock	**0.591**	0.617	0.606	0.627	0.634	0.637	0.670
	Exxon Mobil Stock	0.690	0.708	**0.662**	0.711	0.704	0.686	0.713
TimeMMD	Broiler Price	**0.653**	0.670	0.676	0.777	0.672	0.709	0.696
	Drought Level	**0.689**	0.723	0.710	0.722	0.740	0.773	0.745
	US Trade Balance	0.650	0.637	0.728	**0.630**	0.732	0.918	0.860
	Gasoline Price	**0.671**	0.711	0.732	0.708	0.739	0.754	0.763
	Air Quality Index	**0.837**	0.842	0.909	0.859	0.901	0.867	0.879
	Disaster Grants	**1.450**	1.482	1.480	1.511	1.467	1.450	1.501
	Unemployment Rate	**0.550**	0.586	0.715	0.552	0.727	0.794	0.765
	Travel Volume	0.264	0.303	0.406	**0.262**	0.489	0.663	0.802
TimeMMD Daily	Air Quality Index	0.974	**0.941**	1.013	0.961	0.999	0.960	1.010
TimeMMD Weekly	Drought Level	1.040	**1.027**	1.065	1.071	1.067	1.111	1.104
	Gasoline Price	**1.041**	1.086	1.102	1.066	1.149	1.193	1.177
TimeMMD Monthly	Broiler Price	**1.652**	1.751	1.694	1.703	1.734	1.780	1.710
	US Trade Balance	1.356	**1.293**	1.346	1.350	1.307	1.294	1.436
	Disaster Grants	**1.610**	1.614	1.629	1.641	1.623	1.633	1.707
	Unemployment Rate	1.539	**1.514**	1.586	1.520	1.595	1.621	1.650
	Travel Volume	1.419	1.446	1.275	1.433	1.229	**1.205**	1.325
	Overall Wins	**47**	19	11	13	1	4	0
	Mean	**0.719**	0.734	0.756	0.754	0.775	0.801	0.815
	ELO	**1761**	1642	1572	1499	1438	1325	1263
	Average Rank	**2.07**	2.80	3.45	3.91	4.55	5.38	5.84
	Head-to-Head Wins	–	68/95	71/95	70/95	85/95	83/95	91/95
	Wilcoxon p	–	1.8e-05	1.5e-08	1.3e-08	1.3e-13	2.3e-13	2.7e-16

Table 16: **Dataset-level MAE for multimodal fine-tuning methods.** Each dataset reports the mean over the 64-16, 128-16, and 256-16 settings. TimeMMD Daily, Weekly, and Monthly retain their native settings as separate blocks. Summary statistics are computed across all 95 settings. The CoCo fine-tuning approach for incorporating text into forecasting outperforms other multimodal fine-tuning methods that jointly incorporate numerical history and text. Bold and underlined values denote the best and second-best method in each row.

Group	Setting	Dataset	N	Mean MAE (↓) CoCo	TimesFM2.5+GPT4o	Chronos2+GPT4o	Chronos2 MV	Toto MV	GPT-4o	Qwen3-30B	TimeOmni
Media Cloud	64-16	Crude Oil Price	316	0.802	**0.752**	0.758	0.817	0.760	0.877	0.924	1.082
		S&P500 Index	280	**0.759**	0.797	0.805	0.904	0.929	0.883	0.859	0.978
		Apple Stock	319	**0.903**	0.977	0.977	1.043	1.014	1.089	1.059	1.467
		Copper Futures	179	0.775	0.784	**0.769**	0.893	0.914	0.824	0.824	1.004
		Egg Price	144	0.701	0.651	0.669	0.673	**0.644**	0.714	0.778	0.902
		Electricity Price	334	**0.838**	0.887	0.907	0.846	0.873	1.029	0.967	1.294
		SPDR Gold Price	317	**0.883**	0.918	0.887	1.047	1.105	0.926	1.020	3.396
		Google Stock	319	**0.716**	0.770	0.768	0.867	0.847	0.853	0.843	0.851
		NVIDIA Stock Price	318	**0.848**	0.914	0.934	0.934	0.937	1.104	1.416	1.001
		Silver Price	332	0.903	**0.902**	0.912	1.089	1.112	0.971	0.946	1.137
		Steel Rebar Futures	162	**0.732**	0.779	0.777	0.746	0.739	0.907	0.942	1.400
Media Cloud	128-16	Crude Oil Price	252	0.692	**0.618**	0.641	0.678	0.629	0.709	0.850	0.856
		S&P500 Index	216	**0.573**	0.609	0.641	0.632	0.646	0.727	0.663	0.824
		Apple Stock	255	**0.695**	0.716	0.743	0.736	0.725	0.834	0.876	0.897
		Copper Futures	115	0.589	0.586	0.601	0.604	**0.565**	0.626	0.649	0.702
		Egg Price	80	0.506	0.506	**0.474**	0.516	0.510	0.484	0.566	0.716
		Electricity Price	270	**0.597**	0.668	0.704	0.653	0.670	0.810	0.712	0.884
		SPDR Gold Price	253	0.512	0.530	**0.503**	0.633	0.639	0.519	0.590	0.775
		Google Stock	255	0.526	**0.526**	0.528	0.559	0.572	0.572	0.650	0.659
		NVIDIA Stock Price	254	**0.548**	0.580	0.578	0.580	0.594	0.684	0.782	0.649
		Silver Price	268	**0.667**	0.676	0.676	0.805	0.810	0.761	0.765	0.864
		Steel Rebar Futures	98	0.483	0.427	0.442	0.423	**0.408**	0.483	0.631	1.349
Media Cloud	256-16	Crude Oil Price	124	0.453	**0.341**	0.368	0.525	0.416	0.396	0.533	0.532
		S&P500 Index	88	0.240	0.265	0.262	0.422	0.420	0.262	**0.239**	0.370
		Apple Stock	127	**0.405**	0.440	0.443	0.495	0.507	0.478	0.437	0.509
		Electricity Price	142	**0.467**	0.501	0.502	0.548	0.530	0.556	0.573	0.575
		SPDR Gold Price	125	**0.286**	0.312	0.287	0.309	0.335	0.308	0.346	1.085
		Google Stock	127	0.547	0.540	0.544	0.676	0.686	0.538	**0.509**	0.650
		NVIDIA Stock Price	126	**0.352**	0.373	0.366	0.483	0.441	0.402	0.483	0.459
		Silver Price	140	0.608	0.604	**0.597**	0.809	0.754	0.610	0.604	0.746
		Overall Wins		**16**	5	4	0	3	0	2	0
		Mean		**0.620**	0.632	0.635	0.698	0.691	0.698	0.734	0.954
		ELO		**1727**	1667	1652	1449	1475	1436	1393	1200
		Average Rank		**2.17**	2.77	2.90	5.03	4.80	5.17	5.67	7.50
		Head-to-Head Wins		–	20/30	20/30	27/30	24/30	27/30	27/30	30/30
		Wilcoxon p		–	0.084	0.058	2.3e-06	9.9e-05	2.8e-06	3.5e-08	1.9e-09

Table 17: **Dataset-level MAE for samples strictly after June 2024.** We evaluate CoCo against multimodal baselines in a regime where the LLMs have limited or no prior exposure to the test period. The gains are consistent with those in Table 14, confirming that the improvements stem from the ability of the model to extract predictive signals from text rather than from memorized knowledge. Bold and underlined values denote the best and second-best method in each row.

	Geometric Mean MASE (↓)														
	64–16 Setting			128–16 Setting			256–16 Setting			Native					
Model	**Media Cloud**	**FNSPID**	**TimeMMD**	**Media Cloud**	**FNSPID**	**TimeMMD**	**Media Cloud**	**FNSPID**	**TimeMMD**	**TimeMMD***	**Overall MASE**	**Avg. Rank**	**ELO**	**CoCo Wins**	**Wilcoxon p**
CoCo															
CoCo	**3.064**	3.136	1.605	**3.009**	**3.056**	1.533	**2.817**	2.961	1.530	1.915	**2.455**	**5.21**	**1845**	–	–
Zero-shot Unimodal															
TiRex1	3.136	3.175	1.621	3.075	3.064	1.609	2.857	**2.950**	1.662	1.936	2.506	6.62	1782	59/95	9.4e-04
Chronos2	3.209	3.306	1.625	3.163	3.173	1.607	2.935	3.035	1.617	1.975	2.556	9.56	1668	79/95	4.9e-12
TimesFM2.5	3.230	3.251	1.598	3.184	3.225	1.561	2.994	3.131	1.606	1.900	2.553	9.60	1672	65/95	1.1e-08
Moirai	3.209	3.204	1.693	3.092	3.231	1.631	2.955	3.104	1.638	2.002	2.571	10.15	1655	79/95	7.5e-12
Toto	3.330	3.362	1.661	3.172	3.355	1.627	3.050	3.246	1.632	1.951	2.625	13.60	1542	83/95	1.3e-13
TabPFN2.5	3.462	3.410	1.776	3.559	3.379	1.688	4.153	3.724	1.844	2.002	2.853	18.35	1381	91/95	3.9e-16
TimeMoE	3.949	3.753	1.920	3.548	3.609	1.832	4.361	3.894	1.804	2.050	3.019	20.96	1281	88/95	1.3e-15
PANDA	4.311	4.016	2.311	4.378	4.130	2.162	3.701	3.732	2.045	2.260	3.295	24.03	1136	93/95	4.6e-17
Zero-shot Multimodal															
TimesFM2.5+GPT4o	3.143	3.150	1.787	3.076	3.175	1.739	2.933	3.087	1.642	2.070	2.583	8.80	1700	67/95	3.4e-05
Chronos2+GPT4o	3.171	3.222	1.808	3.096	3.199	1.742	2.951	3.115	1.638	2.143	2.611	10.45	1642	76/95	7.2e-08
Chronos2 MV	3.365	3.270	1.678	3.275	3.196	1.610	3.312	3.167	1.594	1.931	2.618	12.22	1583	70/95	7.3e-09
Toto MV	3.389	3.365	1.647	3.248	3.378	1.592	3.222	3.278	1.568	1.903	2.634	13.11	1550	72/95	1.0e-09
GPT-4o	3.562	3.503	2.283	3.457	3.521	2.173	3.309	3.454	1.996	2.445	2.987	19.79	1332	91/95	2.2e-16
Qwen3-30B	3.663	3.488	2.391	3.686	3.635	2.382	3.529	3.682	2.359	2.485	3.152	21.17	1276	94/95	2.7e-17
TimeOmni	4.932	3.987	3.156	5.190	3.783	4.252	4.533	3.831	3.366	2.616	3.952	25.41	1063	95/95	2.6e-17
Unimodal Fine-tuning															
Chronos2	3.281	3.242	**1.583**	3.118	3.195	**1.498**	2.894	3.105	**1.481**	2.008	2.522	8.96	1689	64/95	2.2e-07
AutoTimes	3.270	3.158	1.902	3.180	3.158	1.906	3.285	3.178	1.972	1.946	2.700	11.55	1602	79/95	1.4e-10
PatchTST	3.337	3.283	1.942	3.252	3.235	1.883	3.037	3.333	1.964	1.926	2.729	14.01	1529	88/95	4.5e-16
S2IP-LLM	3.493	3.342	1.910	3.387	3.227	1.805	3.144	3.370	1.963	2.011	2.769	16.07	1455	90/95	9.0e-16
iTransformer	3.509	3.481	1.999	4.039	4.056	2.085	3.522	3.920	2.055	1.987	3.054	21.45	1257	92/95	4.5e-17
TEST	3.978	3.805	2.301	3.916	3.829	2.221	3.889	4.273	2.519	2.428	3.322	24.94	1091	95/95	2.6e-17
Multimodal Fine-tuning															
Chronos2 MV	3.224	3.142	1.662	3.074	3.136	1.605	2.917	3.103	1.580	**1.852**	2.521	7.47	1742	59/95	2.7e-05
TaTS	3.330	3.233	1.629	3.180	3.192	1.573	3.261	3.168	1.566	1.921	2.582	10.61	1635	68/95	2.5e-07
VoT	3.190	**3.061**	1.872	3.250	3.157	1.834	3.515	3.345	1.869	1.967	2.698	11.57	1604	71/95	3.9e-09
S2IP-LLM	3.316	3.147	1.913	3.329	3.259	1.872	3.400	3.768	1.989	1.945	2.789	14.36	1513	83/95	3.1e-14
AutoTimes	3.429	3.271	2.109	3.460	3.305	2.096	3.587	3.440	2.053	1.979	2.875	16.95	1425	83/95	4.6e-14
TEST	3.334	3.364	2.129	3.531	3.596	2.111	3.693	3.772	2.099	2.064	2.968	19.04	1351	91/95	2.9e-16

Table 18: **MASE comparison across all 28 evaluated configurations and 95 settings.** Each setting/group column reports the geometric mean over its constituent datasets; TimeMMD* pools the Daily, Weekly, and Monthly native-frequency benchmarks. The relative performance of CoCo under MASE is consistent with its performance under MAE in Tables 1 and 2. Bold and underlined values denote the best and second-best models.

	Average Rank (↓)				
Model	**MAE**	**MSE**	**MAPE**	**MAE Unscaled**	**MASE**
CoCo					
CoCo	**5.24**	**5.98**	**6.39**	**5.81**	**5.21**
Zero-shot Unimodal					
TiRex1	<u>7.26</u>	<u>8.63</u>	<u>6.92</u>	<u>6.97</u>	<u>6.62</u>
Chronos2	10.18	11.16	9.81	10.11	9.56
Moirai	10.57	11.15	10.43	10.41	10.15
TimesFM2.5	10.78	12.09	9.35	9.78	9.60
Toto	13.36	13.97	12.82	13.54	13.60
TabPFN2.5	18.27	18.71	17.39	17.85	18.35
TimeMoE	20.82	19.95	20.46	20.32	20.96
PANDA	24.21	23.53	23.76	23.99	24.03
Zero-shot Multimodal					
TimesFM2.5+GPT4o	8.35	8.97	8.49	8.76	8.80
Chronos2+GPT4o	9.65	10.16	10.58	10.37	10.45
Chronos2 MV	13.42	14.45	11.09	11.41	12.22
Toto MV	14.04	14.92	11.58	12.60	13.11
GPT-4o	18.23	17.69	19.34	18.94	19.79
Qwen3-30B	20.96	21.16	22.02	21.45	21.17
TimeOmni	25.67	27.04	25.55	25.62	25.41
Unimodal Fine-tuning					
Chronos2	9.18	10.57	9.80	9.55	8.96
AutoTimes	11.75	11.17	11.56	11.41	11.55
PatchTST	13.06	12.61	15.11	14.96	14.01
S2IP-LLM	14.98	13.61	16.93	16.57	16.07
iTransformer	20.49	18.01	21.84	21.58	21.45
TEST	24.14	21.93	24.97	24.98	24.94
Multimodal Fine-tuning					
Chronos2 MV	8.02	9.40	8.25	7.72	7.47
VoT	11.06	9.65	10.85	10.91	11.57
TaTS	11.79	13.08	9.37	9.95	10.61
S2IP-LLM	14.18	12.99	14.99	14.53	14.36
AutoTimes	17.40	16.51	16.97	16.83	16.95
TEST	18.93	16.93	19.39	19.12	19.04

Table 19: **CoCo performs consistently across raw and normalized evaluation metrics.** Average ranks are computed over the same 95 settings, similar to Tables 1, 2, and 18. Bold and underlined values denote the best and second-best method for each metric.

				Mean MAE (↓)								
Group	**Setting**	**Dataset**	**N**	**CoCo**	**TimesFM2.5+GPT4o**	**Chronos2+GPT4o**	**Chronos2 MV**	**Toto MV**	**GPT-4o**	**Qwen3-30B**	**TimeOmni**	**GPT-5**
Media Cloud	64-16	Crude Oil Price	421	0.877	0.817	0.831	0.853	**0.816**	0.930	1.027	1.160	0.859
		S&P500 Index	384	0.797	0.827	0.843	0.944	0.998	0.896	0.855	0.979	**0.766**
		Apple Stock	424	0.872	0.948	0.945	1.008	1.003	1.043	0.961	1.325	**0.854**
		Copper Futures	179	0.775	0.784	**0.769**	0.893	0.914	0.824	0.824	1.004	0.787
		Egg Price	144	0.701	0.651	0.669	0.673	**0.644**	0.714	0.778	0.902	0.729
		Electricity Price	444	**0.798**	0.854	0.858	0.831	0.872	0.948	0.907	1.189	0.829
		SPDR Gold Price	422	0.849	0.866	0.844	1.010	1.066	0.895	0.976	4.565	**0.824**
		Google Stock	423	**0.730**	0.757	0.748	0.881	0.858	0.822	0.869	0.876	0.757
		NVIDIA Stock Price	423	**0.883**	0.939	0.950	0.966	0.984	1.084	1.330	1.021	0.948
		Silver Price	440	**0.886**	0.895	0.907	1.003	1.028	0.982	0.966	1.131	0.888
		Steel Rebar Futures	162	**0.732**	0.779	0.777	0.746	0.739	0.907	0.942	1.400	0.828
Media Cloud	128-16	Crude Oil Price	357	0.714	**0.634**	0.657	0.727	0.658	0.721	0.900	0.904	0.710
		S&P500 Index	320	**0.562**	0.607	0.617	0.641	0.656	0.700	0.650	0.811	0.574
		Apple Stock	360	**0.610**	0.639	0.657	0.657	0.642	0.737	0.757	0.768	0.630
		Copper Futures	115	0.589	0.586	0.601	0.604	**0.565**	0.626	0.649	0.702	0.602
		Egg Price	80	0.506	0.506	**0.474**	0.516	0.510	0.484	0.566	0.716	0.520
		Electricity Price	380	**0.531**	0.585	0.609	0.606	0.613	0.686	0.631	0.789	0.579
		SPDR Gold Price	358	**0.486**	0.510	0.492	0.617	0.610	0.526	0.585	3.884	0.488
		Google Stock	359	**0.546**	0.566	0.575	0.549	0.573	0.643	0.678	0.685	0.568
		NVIDIA Stock Price	359	**0.541**	0.588	0.590	0.578	0.595	0.698	0.764	0.647	0.580
		Silver Price	376	**0.665**	0.680	0.689	0.762	0.771	0.776	0.791	0.837	0.695
		Steel Rebar Futures	98	0.483	0.427	0.442	0.423	**0.408**	0.483	0.631	1.349	0.482
Media Cloud	256-16	Crude Oil Price	229	0.509	**0.423**	0.448	0.540	0.463	0.488	0.599	0.598	0.483
		S&P500 Index	192	**0.422**	0.468	0.488	0.493	0.519	0.574	0.459	0.564	0.435
		Apple Stock	232	**0.428**	0.459	0.472	0.480	0.473	0.531	0.543	0.535	0.440
		Electricity Price	252	**0.399**	0.445	0.456	0.456	0.454	0.523	0.467	0.575	0.445
		SPDR Gold Price	230	0.305	0.330	0.303	0.354	0.353	0.326	0.389	1.955	**0.293**
		Google Stock	231	0.540	0.534	0.534	0.578	0.574	0.573	0.768	0.618	**0.519**
		NVIDIA Stock Price	231	**0.416**	0.446	0.435	0.503	0.485	0.494	0.561	0.533	0.457
		Silver Price	248	0.547	**0.544**	0.547	0.723	0.670	0.592	0.584	0.704	0.561
		Overall Wins		**16**	3	2	0	4	0	0	0	5
		Mean		**0.623**	0.636	0.641	0.687	0.684	0.708	0.747	1.124	0.638
		ELO		**1745**	1677	1633	1440	1487	1381	1311	1173	1654
		Average Rank		**2.27**	2.97	3.47	5.67	5.17	6.43	7.23	8.60	3.20
		Head-to-Head Wins		–	22/30	20/30	27/30	24/30	28/30	30/30	30/30	21/30
		Wilcoxon p		–	0.067	0.028	3.2e-06	3.4e-04	1.0e-07	1.9e-09	1.9e-09	0.015

Table 20: **Dataset-level MAE for 16-step forecasts across context lengths 64, 128, and 256 on the Media Cloud subset, including GPT-5 as an additional LLM baseline.** Media Cloud, comprising datasets from 2024–2025, is the only subset where GPT-5 has limited prior exposure to test trends. GPT-5 performs competitively with strong TSFMs, yet CoCo outperforms GPT-5 in 70% (21/30) of settings, demonstrating that a fine-tuned 4B LLM providing forecast-space covariates to Chronos2 surpasses a closed-source model used as a standalone forecaster. Bold and underlined values denote the best and second-best method in each row.

MAE (↓)		
CoCo Fine-tuned Chronos2	CoCo (With Misleading Text)	**CoCo (With Truthful Text)**
2.929	3.025	**2.802**

Table 21: **CoCo's performance degrades when text is adversarial.** We evaluate on a synthetically generated dataset of 1000 samples where the same numerical series is paired with two kinds of annotations: *truthful* text describing the correct future trend, and *misleading* text deliberately describing the opposite trend. CoCo improves over the fine-tuned Chronos2 baseline with truthful text but degrades with misleading text, consistent with the gating analysis in Figure 7 (right), where confidently wrong LLM forecasts are the most harmful regime.

	LLM Forecast $\theta_{\text{LLM}}(X_t^H, S_t)$		Chronos2 Forecast $\theta_{\text{M-TSFM}}(X_t^H)$		CoCo $\theta_{\text{M-TSFM}}(X_t^H \| \| M_t)$		CoCo $\theta_{\text{M-TSFM}}(X_t^H \| \| M_t \| \| C_t)$	
LLM	MAE (↓)	Rank (↓)	MAE (↓)	Rank (↓)	MAE (↓)	Rank (↓)	MAE (↓)	Rank (↓)
Zeroshot	1.270	8.842	0.980	-	1.028	2.579	1.110	4.676
Qwen3-4b Fine-tuned	1.078	3.816	0.980	-	0.978	2.632	0.964	2.684
Chronos2 Fine-tuned	1.270	8.842	0.959	2.342	0.956	2.459	0.955	2.459
Both Fine-tuned	1.078	3.816	0.959	2.342	0.954	2.263	**0.947**	**2.184**

Table 22: **Effect of fine-tuning and the usefulness of the covariates.** Extended version of Table 3 showing MAE and average rank over the 30 datasets for the 64-16 setting and the 8 TimeMMD datasets with their native settings (38 settings in total) for different fine-tuning settings and covariate inputs. '-' indicates that the rank is not valid since Chronos2 is one of the baselines.

	CoCo Forecast $\theta_{\mathrm{M-TSFM}}(X_t^H \| \| M_t \| \| C_t)$	
N_{rollouts}	MAE (↓)	Rank (↓)
5	**0.845**	**1.833**
9	**0.845**	1.867
15	0.846	1.900

Table 23: **Effect of N_{rollouts}, the number of rollouts used to compute confidence covariate on performance.** The MAE and the average rank are shown over the 30 datasets for 16-step forecasts with a context length of 64.

	LLM Forecast ($\theta_{\mathrm{LLM}}(X_t^H, S_t)$)			
Qwen3-4b Fine-tuning	Overall MAE (↓)	TimeMMD MAE (↓)	FNSPID MAE (↓)	Rank (↓)
Supervised Fine-tuning	1.187	1.635	1.050	8.289
CoCo Fine-tuning without Text	1.172	1.543	0.954	4.395
CoCo Fine-tuning with Synthetic Text	1.087	1.365	0.903	3.974
CoCo Fine-tuning with Real Text	1.140	1.528	**0.891**	**3.605**
CoCo Fine-tuning with Real and Synthetic Text	**1.078**	**1.354**	0.894	3.737

Table 24: **Effect of using text and the kind of text used in fine-tuning the LLM.** The results are for the `Qwen3-4B-Instruct-2507` model. Real text here denotes FNSPID data and synthetic text refers to the dataset described in Appendix E.2. Though the model trained completely with FNSPID data has a marginally better rank (the wins are concentrated in the FNSPID subset, which was seen during training), we observe that it fails in the TimeMMD dataset, which is out of distribution and hence does not generalize. On the other hand, models trained on completely synthetic text or a mixture of both provide improved performance in TimeMMD as well. The MAE and average ranks reported are over the same 38 settings as Table 22.

M-TSFM	LLM	Mean MAE (↓)			
		$\theta_{\mathrm{LLM}}(X_t^H, S_t)$	$\theta_{\mathrm{M-TSFM}}(X_t^H)$	$\theta_{\mathrm{M-TSFM}}(X_t^H \| M_t)$	$\theta_{\mathrm{M-TSFM}}(X_t^H \| M_t \| C_t)$
Chronos2	GPT-OSS-120B	0.885	0.847	**0.828**	0.842
	Qwen3-30B	0.949	0.847	**0.816**	0.851
	Gemma3-27B	0.871	0.847	**0.821**	0.827
	Qwen3-4B	1.019	0.847	**0.824**	0.895
	Qwen3-4B (FT)	0.862	0.847	0.818	**0.801**
Toto	GPT-OSS-120B	0.885	0.865	0.864	**0.854**
	Qwen3-30B	0.949	0.865	0.859	**0.851**
	Gemma3-27B	0.871	0.865	0.859	**0.855**
	Qwen3-4B	1.019	0.865	0.863	**0.858**
	Qwen3-4B (FT)	0.862	0.865	0.862	**0.857**
Moirai	GPT-OSS-120B	0.885	0.843	**0.838**	0.844
	Qwen3-30B	0.949	0.843	0.841	**0.839**
	Gemma3-27B	0.871	0.843	**0.839**	0.843
	Qwen3-4B	1.019	0.843	**0.839**	0.841
	Qwen3-4B (FT)	0.862	0.843	**0.841**	0.845
Chronos2 (LoRA FT)	GPT-OSS-120B	0.885	0.824	0.816	**0.813**
	Qwen3-30B	0.949	0.824	0.814	**0.811**
	Gemma3-27B	0.871	0.824	0.814	**0.810**
	Qwen3-4B	1.019	0.824	0.815	**0.815**
	Qwen3-4B (FT)	0.862	0.824	0.815	**0.809**

Table 25: **CoCo generalizes across LLM and M-TSFM choices.** Mean MAE is reported on the Media Cloud datasets for the 64-16 context–horizon setting. Across five LLM configurations and three M-TSFM backbones, the forecast covariate M_t improves over the corresponding M-TSFM baseline in every evaluated combination. The confidence covariate C_t provides further gains with the zero-shot Toto backbone and after fine-tuning Chronos2. Importantly, Chronos2 is fine-tuned using rollouts pre-generated only by the fine-tuned Qwen3-4B, yet it generalizes to covariates from every evaluated zero-shot LLM. These results demonstrate that CoCo is not reliant on properties specific to Qwen3-4B or Chronos2. **Bold** denotes the best result in each row.

				Mean MAE (↓)			
Group	**Setting**	**Dataset**	**N**	TSFM Toto	TSFM MV Toto	CoCo (With Toto) Only Forecast Covariate	**CoCo (With Toto) Both Covariates**
Suite	64-16	Crude Oil Price	421	0.900	**0.816**	0.899	0.899
		S&P500 Index	384	**0.926**	0.998	0.959	0.948
		Apple Stock	424	0.959	1.003	0.951	**0.941**
		Copper Futures	179	0.827	0.914	**0.814**	0.815
		Egg Price	144	0.666	**0.644**	0.738	0.694
		Electricity Price	444	0.847	0.872	0.840	**0.838**
		SPDR Gold Price	422	0.987	1.066	0.968	**0.953**
		Google Stock	423	0.794	0.858	0.780	**0.775**
		NVIDIA Stock Price	423	0.935	0.984	0.926	**0.914**
		Silver Price	440	**0.924**	1.028	0.925	0.925
		Steel Rebar Futures	162	0.747	0.739	0.740	**0.739**
FNSPID	64-16	Boeing Stock	421	0.834	0.822	0.824	**0.813**
		Bank of America Stock	420	1.103	**1.084**	1.111	1.101
		Costco Stock	421	0.842	0.875	0.843	**0.829**
		Goldman Sachs Stock	421	0.968	**0.935**	0.959	0.951
		JPMC Stock	358	1.131	**1.117**	1.134	1.132
		McDonald's Stock	421	0.957	1.005	0.946	**0.938**
		Microsoft Stock	333	0.947	0.987	0.943	**0.938**
		Oracle Stock	421	0.913	0.926	0.906	**0.894**
		Tesla Stock	331	0.848	0.829	0.830	**0.817**
		Walmart Stock	421	0.987	0.991	0.970	**0.958**
		Exxon Mobil Stock	421	0.972	1.008	0.956	**0.952**
TimeMMD	64-16	Broiler Price	453	0.984	1.063	0.978	**0.974**
		Drought Level	1193	0.909	0.920	0.907	**0.904**
		US Trade Balance	368	0.882	**0.837**	0.901	0.899
		Gasoline Price	1543	0.926	0.937	**0.907**	0.912
		Air Quality Index	1401	0.886	**0.882**	0.884	0.884
		Disaster Grants	230	1.139	1.144	**1.136**	1.140
		Unemployment Rate	453	**0.659**	0.660	0.670	0.680
		Travel Volume	453	**0.304**	0.308	0.319	0.316
		Overall Wins		4	7	3	**16**
		Mean		0.890	0.908	0.889	**0.882**
		ELO		1458	1415	1486	**1640**
		Average Rank		2.767	2.967	2.567	**1.700**
		Head-to-Head Wins		22/30	21/30	26/30	-
		Wilcoxon p		1.28e-02	9.3e-03	4.97e-05	-

Table 26: **Results with Toto as the TSFM backbone, using covariates generated by the fine-tuned Qwen3-4b LLM without any Toto-specific fine-tuning.** CoCo (with Toto) outperforms Toto in 73% $(22/30)$ of datasets and outperforms sentence transformer augmented Toto (TSFM MV) in 70% $(21/30)$ of the datasets, demonstrating that the covariate design transfers across TSFM backbones and is not specific to Chronos2. **Bold** and underlined indicate the best and second best per dataset respectively.

Annotations	Direction	Magnitude Tertile	Argmax Bucket
Media Cloud (Real)	0.533	0.344	0.301
FNSPID (Real)	0.544	0.340	0.279
Synthetic (LLM generated with ground truth future values)	0.555	0.316	0.272
Synthetic (LLM generated without ground truth future values)	0.491	0.376	0.346

Table 27: **Synthetic textual annotations that depend on numeric history alone provide incorrect information about the true future.** In this table, we show the accuracy of linear probes for predicting the domain-agnostic forecast statistics, such as the direction, magnitude tertile, and the argmax location/bucket. Note that the linear probes take an LLM's embeddings of the context as input. In our training, we utilize synthetic annotations/context that is synthetically generated (Section 4). Here, we ablate the synthetic annotation generation strategy. We show two approaches: one where the numeric future is used to generate the annotation (row 3) and one where the numeric future is not used (row 4). We compare the accuracies of predicting domain-agnostic forecast statistics from these two variants of synthetic annotations with the accuracies of predicting the same from real-annotations. Prefix-only (without the numeric future, row 4) annotations are generated using only domain information and the observed history, without access to forecast values. Compared with real and future-conditioned annotations, they yield lower direction accuracy but substantially higher magnitude-tertile and argmax-bucket accuracies, indicating an out-of-distribution probe profile relative to the real text. This mismatch suggests that prefix-only generation may weaken sim-to-real transfer and motivates our randomized-context design (Section 4) as a middle ground between limiting future leakage and preserving real-world text statistics. However, one key limitation of this experiment is that the probes are linear, and more non-linear approaches could extract better forecast statistics from prefix-only annotations as well.

Fine-tuned Components	Mean MAE (↓)	Avg. Rank among Baselines (↓)	ELO among Baselines (↑)	Avg. Rank among Variants (↓)	ELO among Variants (↑)
M-TSFM & LLM	**0.719**	**5.24**	**1829**	**1.87**	**1535**
LLM	0.732	6.44	1785	1.93	1508
M-TSFM	0.724	7.16	1758	2.19	1457

Table 28: **Ablation of fine-tuned components in CoCo across all 95 settings.** We compare variants with only the LLM fine-tuned, only the M-TSFM fine-tuned, and both components fine-tuned, evaluating each against all the 27 methods as well as against the other CoCo variants. Fine-tuning both components jointly (in a sequential manner as described in Section 5) achieves the best mean MAE, average rank, and ELO in both comparisons. Fine-tuning either component individually also outperforms the zero-shot baselines, demonstrating that both stages contribute independently. Notably, the advantage of jointly fine-tuning both components is more pronounced when evaluated against the full baseline pool than in the direct three-way comparison.

Model	Latency (ms/sample)
Chronos2	$\mathbf{11.43 \pm 7.18}$
Toto	31.15 ± 39.67
TimesFM 2.5	37.08 ± 52.43
Chronos MV	75.77 ± 54.26
TabPFN 2.5	270.17 ± 28.62
CoCo	280.07 ± 21.96
Toto MV	312.14 ± 2.47

Table 29: **Inference latency (ms/sample) for CoCo and all TSFM baselines, measured on a single H100 GPU.** Latency is computed by processing 128 samples (16-step forecasts with a context length of 64) in batches of 16 and averaging over batches. For CoCo, the fine-tuned Qwen3-4b-Instruct-2507 is served via a vLLM inference server on the same GPU, with $k = 5$ rollouts generated per sample for the confidence covariate C_t. Despite requiring LLM inference with multiple rollouts, CoCo achieves 280 ms/sample, comparable to TabPFN 2.5. The latency overhead is a direct consequence of incorporating an LLM, which we view as necessary for extracting predictive signals from text, a capability that unimodal TSFMs fundamentally lack.

I LLM Prompts

I.1 Synthetic Annotation Prompt

The following prompt is used to generate the large-scale training dataset in Section E.2.

```
You are given a time series for the economic indicator: { Series Title
/ Information }. The data consists of dates (t) and corresponding
values in appropriate units. Your task is to annotate each data point
with a text column that serves as an analytical briefing note. For
each data point (t, y_t), follow this structured thinking process to
generate the text:

1.  Observation: State the current value and its immediate change. Is
it an increase or decrease? Is the pace of change accelerating,
decelerating, or stable compared to the prior few periods?

2.  Contextualization: Explain this movement by citing 2-3 plausible,
interconnected macroeconomic drivers. Make sure to use factors that
align and are relevant to the time period / date of the data point.

3.  Outlook: Provide a speculative, forward-looking statement. This
outlook must NOT reveal future data. Instead, frame it in terms of
risks, potential, and key indicators to monitor for confirmation. Use
cautious but insightful language.

You don't need to have explicit sections in your response. Just the
text.

Key Directives:
-   Report-Like Tone: The text must be formal, objective, and
analytical.
-   Predictive Value: The combination of context and outlook should
provide a rich narrative basis for forecasting without leaking future
values.
-   Dynamic Narrative: Do not use repetitive sentence structures or
explanations. Each annotation should feel like a unique daily or
monthly report.
-   Conciseness: Keep each annotation under 150 words.
-   Output Format: Your entire output must be in CSV format with three
columns:`t`, `y_t`, `text`. Do not add headers, markdown, or any text
outside of the CSV rows.

Begin generating the CSV output now.

Data:
{Data as rows containing timestamp and value.}
```

I.2 LLM Forecast Prompt

The following prompt is used to obtain forecasts from LLMs, given the numerical history and textual annotations. It is used to both evaluate LLMs for forecasting performance and to obtain the covariates required for CoCo.

```
You are a time series forecasting expert. Analyze the historical data
with text annotations to predict future values.

FULL TIME SERIES ({l: context length} timesteps):
[
{Data as rows containing timestamps and values}
]
```

```
DETAILED CONTEXT (most recent 16 timesteps with text annotations):
[
{Textual context as rows containing timestamps and annotations}
]

TASK:
Analyze the full time series pattern and the text annotations for:
1.  FACTUAL PATTERNS: What trends, events, or behaviors are described
in the recent history?
2.  PREDICTIVE SIGNALS: Are there any forward-looking statements,
expectations, or indicators of future change?

Based on this analysis, provide a forecast for the next {h: prediction
horizon} steps. FORECAST TARGET TIMESTAMPS (next steps, in order):
[
{Timestamps to forecast}
]

IMPORTANT: Your forecast numbers MUST correspond to these timestamps
in the same order. Output ONLY {h: prediction horizon} comma-separated
numbers representing your forecast. Do not include any explanations or
other text.

Forecast:
```

I.3 Media Cloud Dataset Collection

I.3.1 Prompt used to filter irrelevant articles and summarize relevant articles

```
First, determine if the following article is related to {topic} in any
way.

If the article is NOT related to {topic}, respond with exactly: NA

If the article IS related to {topic}, provide a concise summary (3-4
sentences) focusing on:
1. Key facts and events
2. Forward-looking statements or predictions
Only use the provided article to create the summary.

Title: {article title}

Content:
{article content}

Response:
```

I.3.2 Prompt used to generate summaries from collection of articles

```
You are analyzing news articles from {date} to provide a comprehensive
daily summary for financial forecasting.

Below are summaries of {N} articles from this date:

{Article 1 Summary: ....}
{Article 2 Summary: ....}
...
{Article N Summary: ....}
```

```
Based on these article summaries, provide a comprehensive daily
summary with TWO sections:

SECTION 1 - KEY EVENTS AND FACTS:
Synthesize the main events, developments, and factual information from
the day. Focus on market movements, economic data, corporate
announcements, and significant news. (3-5 sentences)

SECTION 2 - FORWARD-LOOKING SIGNALS:
Identify and synthesize any forward-looking information, predictions,
market expectations, or signals that could indicate future trends.
Include analyst forecasts, anticipated policy changes, and emerging
patterns. (3-5 sentences)

Please provide your response in exactly this format:

KEY EVENTS AND FACTS:
[Your synthesis here]

FORWARD-LOOKING SIGNALS:
[Your synthesis here]
```

J Broader Societal Impacts

Our main contribution in this work is **Co**ntext as **Co**variates (CoCo), which is a novel forecasting pipeline that combines the strengths of LLMs and TSFMs to perform accurate multimodal time series forecasting. We also propose a novel approach to generate synthetic multimodal annotations, thereby enabling **sim-to-real** transfer in domains that lack multimodal time series datasets. We note that improved forecasting using exogenous text has a lot of positive impacts in the time series domain, as it can enable accurate decision making based on precise forecasts. Additionally, CoCo unlocks the ability to perform accurate scenario simulation by providing speculative text annotations for the future. As such, the societal consequences of our work are limited to those arising from improved quality of generated synthetic data. Apart from that, one of the limitations (Section 6) of CoCo is that it is sensitive to incorrect or misleading text. In other words, if the text is deliberately misleading or false, CoCo can generate erroneous forecasts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We prove the effectiveness of CoCo to interpret text accurately and improve multimodal forecasting in Section 6. Additionally, we also showcase the effectiveness of training on synthetic multimodal data that is generated using the approach outlined in Section 4 (see Table 24).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 6, we list three key limitations of our approach pertaining to inference latency, lack of evaluating the reliability of text, and limitations in benchmarks where numerical history has the least significance for forecasting.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We validate Assumptions 4.1 and 4.2 through real-world experiments. Lemma 4.3 and Theorem 4.4 are proved in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All training and inference hyperparameters are clearly discussed in Appendix E.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided the code, data, and instructions to reproduce the results in Section 6, as part of our supplementary material. Additionally, we have publicly released the code and data, with the links provided in the manuscript.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendices E.1 and F.1 contain the exact training and testing related hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In all the tables, we report the Wilcoxon p-value for statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.1 describes the required compute resources for the LLM fine-tuning experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our research conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work mainly focuses on improving time series forecasting with multimodal context. As explained in the limitations (see Section 6), we note that our approach utilizes the textual context without guardrails on how much the text should be trusted. We provided a detailed discussion regarding the same in Appendix J.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We believe our work does not suffer from high-risk misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models used in this work are properly credited via citations in the main paper. Specifically: (1) Qwen3-4b [9] (`Qwen/Qwen3-4B-Instruct-2507`) is released under Apache 2.0; (2) Chronos2 [8] (`amazon/chronos-2`, version `2.1.0`) is released under Apache 2.0; (3) TimeMMD [11] is released under CC BY 4.0; (4) FNSPID [13] is released under CC BY-NC 4.0, and is used exclusively for research purposes consistent with its non-commercial license. Media Cloud data is accessed via the public Media Cloud API [56] in accordance with its terms of service. All model checkpoints are accessed from their official HuggingFace repositories.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our paper introduces three new assets: (1) a synthetic multimodal training dataset, (2) a curated real-world evaluation dataset, and (3) CoCo fine-tuned TSFM and LLM checkpoints. We provide detailed documentation for all assets to ensure reproducibility. Appendix E.2 and Appendix I describe the synthetic data generation pipeline, along with the prompts used. Appendix F.1 provides details on the numerical time series and textual annotations used to curate the real-world evaluation dataset. Appendix E.1 documents the fine-tuning procedures, hyperparameters, and training configurations for both the TSFM and LLM components. We additionally provide the fine-tuned checkpoints and inference pipeline as supplementary material, and have open-sourced the curated datasets (check the main manuscript for the links).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our work utilizes LLMs to understand and interpret the effect of text on time series forecasting. Our approach involves fine-tuning LLMs for multimodal forecasting. We have described the fine-tuning procedure in Section 5 and described all the prompts that we used in the Appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.