GIFT-EVAL-CTX: A BENCHMARK FOR CONTEXTUAL FORECASTING MODEL EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Contextual forecasting seeks to improve time series forecasting by incorporating forward-looking information, such as planned events, external conditions, and expert expectations. However, reliable evaluation remains difficult: existing benchmarks often have limited domain coverage, lack standardized history–future splits, or provide text that is not guaranteed to inform the future. We introduce **GIFT-Eval-CTX**, a benchmark that isolates whether forecasting methods can translate verified future information into accurate numerical trajectories. Built upon GIFT-Eval, GIFT-Eval-CTX contains 10,000 instances across 20 datasets. Each instance provides structured effects derived from the realized future: Macro effects describe horizon-wide behavior, while Micro effects specify localized changes. We construct these annotations with an agentic pipeline combining LLM-based discovery and deterministic code validation. We benchmark TSFMs, frontier LLMs, and tool-using agentic systems. The results show that structured effects reduce aggregate relative MASE by 40.8% for GPT-5.6 Sol and 30.1% for Claude Opus 4.8 relative to history-only prompting. The evaluated MoiraiAgent system achieves the best overall result. We further analyze the contributions of structured effects and agentic systems to performance, and identify opportunities for future research.

1 INTRODUCTION

Time series foundation models (TSFMs) have become strong zero-shot forecasters (Das et al., 2024; Ansari et al., 2025; Liu et al., 2025a; Khwaja et al., 2026), but their standard interfaces are designed primarily for numerical histories and predefined covariates. Consequently, they are not naturally equipped to incorporate open-ended forward-looking information, such as an energy operator expecting unusually high electricity demand during a heatwave, a transportation system facing a planned road closure, or a domain expert specifying how demand may evolve under a future scenario. Contextual forecasting addresses this limitation by integrating textual context with numerical history to guide forecasts (Xie et al., 2024; Liu et al., 2024b; Wang et al., 2025; Guan et al., 2026a). This requires two complementary capabilities: understanding what diverse, domain-specific context implies about the future and translating those implications into accurate numerical trajectories.

Developing and comparing contextual forecasting systems requires reliable benchmarks that pair diverse time series with relevant forward-looking context under standardized evaluation protocols. However, existing contextual forecasting benchmarks fall short of these requirements in several ways (Xu et al., 2025b; Kong et al., 2025; Xu et al., 2025a; Yin et al., 2026; Lee et al., 2026). First, collecting aligned text and time series data is difficult, resulting in limited domain diversity. Second, many datasets lack standardized history–future splits (e.g., providing only several years of financial news and related stock prices). Researchers must construct these splits themselves, reducing comparability across studies. Third, the accompanying text is not guaranteed to contain information about future events or conditions, making it unclear whether the benchmark truly evaluates a model's ability to use forward-looking context.

In this work, rather than treating contextual forecasting as a single end-to-end task, we distinguish two capabilities it requires: *contextual understanding* and *numerical forecasting*. We treat contextual understanding as an upstream capability handled by frontier LLMs, whose knowledge and reasoning continue to improve rapidly, and focus on the evaluation of the second capability: generating accurate numerical forecasts from verified future information. To this end, we introduce **GIFT-Eval-**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

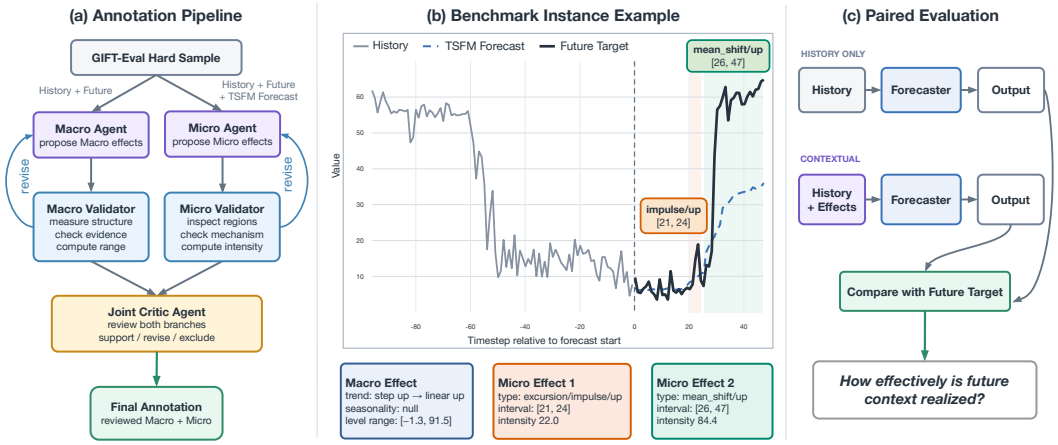


Figure 1: Illustration of agentic annotation, an annotated instance, and paired evaluation.

CTX, a contextual extension of the GIFT-Eval benchmark (Aksu et al., 2024). GIFT-Eval offers broad domain diversity and standardized training–test and history–future splits, addressing the first two limitations above. To ensure that every context is informative about its corresponding future, we derive structured context from the realized future during benchmark construction, analogous to supplying a verified scenario or expert outlook. GIFT-Eval-CTX therefore does not test whether a model can extract or validate future information from raw sources, an upstream task that we believe can be handled by domain experts and increasingly capable frontier LLMs. Instead, it evaluates whether a model can convert provided future information into an accurate numerical trajectory.

To make the context portable across domains and less dependent on the language style of any particular LLM, we represent it as a compact key-value schema called **structured effects**, rather than as free-form text. Structured effects have two compositional levels. **Macro effects** describe horizon-wide trend, seasonality, and value range. **Micro effects** describe localized excursions, mean shifts, and seasonal changes, together with their temporal boundaries and scale. We then propose an agentic labeling pipeline to generate these structured effects. Concretely, we first identify GIFT-Eval test windows on which state-of-the-art TSFMs cannot solve easily, i.e., where future context is needed. Then a Macro Agent and a Micro Agent inspect the numerical evidence independently: the former summarizes global future structure, while the latter identifies localized mechanisms that are not adequately captured by the TSFMs' forecast. Deterministic validators enforce the schema, verify numerical evidence, and compute quantitative fields, after which a Joint Critic Agent reviews all proposed labels and commits a final annotation.

We benchmark three classes of forecasting methods on GIFT-Eval-CTX: TSFMs, frontier LLMs, and agentic systems that augment LLMs with tool use. The results show that LLM forecasts improve substantially when given structured effects: aggregate relative MASE decreases by 40.8% for GPT-5.6 Sol and 30.1% for Claude Opus 4.8 relative to history-only prompting. These improvements demonstrate that the structured effects are informative and actionable. Moreover, MoiraiAgent, which combines a frontier LLM with code-execution tools and a TSFM forecasting service, achieves the best overall performance. We also analyze how Macro and Micro effects contribute across forecast horizons, Micro families, and regions inside and outside labeled Micro intervals. Finally, we examine the sources of MoiraiAgent's gains, showing how its forecasting and code-execution tools improve performance, and identify limitations of code-based forecast editing.

2 Related Work

2.1 Multimodal Time Series Benchmarks

Multimodal time series benchmarks have expanded time series evaluation beyond point prediction. One line of work focuses on understanding and question answering across a diverse collection of tasks (Merrill et al., 2024; Weng et al., 2026). MTBench (Chen et al., 2025) and Time-MQA (Kong et al., 2025) pair numerical sequences with textual context and questions that require temporal or

cross-modal reasoning. MMTS-Bench (Yin et al., 2026) and TSRBench (Yu et al., 2026) broaden the scope to tasks such as pattern perception, feature analysis, and decision making. BEDTime (Sen et al., 2025) instead evaluates the ability to describe and distinguish time series patterns in text.

A second line of work concentrates more directly on multimodal forecasting. Context is Key (Williams et al., 2025) pairs numerical series with manually constructed, and sometimes intervention-aligned, synthetic textual contexts. Time-MMD (Liu et al., 2024a) aligns time series with real textual information across domains, and MoTime (Zhou et al., 2025) augments forecasting datasets with text, metadata, and images. FinTexTS (Lee et al., 2026) constructs text–time series pairs for financial forecasting through multi-level news matching. FinMultiTime (Xu et al., 2025a) further extends financial multimodality by incorporating financial tables and K-line charts. However, these existing benchmarks often have limited domain coverage and lack standardized history–future splits. Moreover, their text is not guaranteed to provide future-relevant information. GIFT-Eval-CTX complements these benchmarks with broad domain coverage, standardized history–future splits, and structured context guaranteed to describe the realized future. It thereby isolates a specific downstream capability: translating provided future information into an accurate numerical trajectory.

3 GIFT-EVAL-CTX

3.1 TASK DEFINITION

Let $\mathbf{x}_{1:T} \in \mathbb{R}^T$ denote an observed time series history and $\mathbf{y}_{1:H} \in \mathbb{R}^H$ its future target over a horizon of H steps. In contextual forecasting, the model additionally receives structured future context $\mathbf{s}$ describing the global behavior and localized effects within $\mathbf{y}_{1:H}$. Given $(\mathbf{x}_{1:T}, H, \mathbf{s})$, a contextual forecaster F produces $\hat{\mathbf{y}}_{1:H} = F(\mathbf{x}_{1:T}, H, \mathbf{s})$. GIFT-Eval-CTX supports a paired evaluation of the same forecasting model under two information conditions. In the *history-only* condition, the model receives only $(\mathbf{x}_{1:T}, H)$. In the *contextual* condition, it additionally receives $\mathbf{s}$. Comparing the two conditions allows us to assess how effectively a model uses the supplied future context.

3.2 TIME SERIES CORPUS

Data source. We construct GIFT-Eval-CTX from the held-out test regions of GIFT-Eval. Each multivariate series is flattened into univariate targets, and no window is cut from a training or validation region. We consider forecast horizons of at least 24 steps, since shorter futures rarely contain meaningful effects. To limit token costs for long histories, we cap the history length at $10H$, $5H$, and $3H$ for short-, medium-, and long-horizon settings, respectively.

Difficulty filtering. An important design principle of GIFT-Eval-CTX is to focus on test windows that TSFMs cannot solve easily. If a TSFM already forecasts the future accurately from history alone, additional context leaves little room for meaningful improvement. To identify non-trivial windows, we forecast every candidate with top-performing TSFMs on the GIFT-Eval leaderboard, and define the range-normalized error $\mathrm{rMAE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{H^{-1}\sum_{t=1}^{H}|\hat{y}_t - y_t|}{\max_t y_t - \min_t y_t}$. We retain a window only when all evaluated models obtain an rMAE larger than a threshold. Normalizing by the future range makes the threshold comparable across datasets and windows with different scales.

3.3 STRUCTURED EFFECTS SCHEMA

Free-form descriptions confound the information conveyed by a context with its wording, domain vocabulary, and level of detail. We instead represent future context through *structured effects*, a domain-agnostic key-value schema designed to be portable across models. Structured effects decompose the future into a single Macro description and a set of Micro effects.

Macro effects. Macro describes the complete forecast horizon through three fields:

- `trend`: one or two chronological segments describing the horizon-wide baseline movement. Each segment specifies a `direction` that describes the movement of the center line within the segment, a `shape` that describes the curvature of the trend, and a `duration` that gives its fraction of the horizon. Segment durations sum to one.

- `seasonality`: the dominant oscillatory structure of the future, or null when no meaningful oscillation is present. When present, it specifies a `period` in time steps and a `waveform` that describes the cycle shape.
- `level_range`: the min and max future values in history-normalized coordinates.

Micro effects. Each Micro effect occupies an interval $[a, b]$ in the forecast horizon and describes a localized mechanism that is not adequately captured by the TSFMs' forecast. Multiple Micro effects may coexist or overlap when they describe different mechanisms. Micro has the following fields:

- `family`: `excursion` (a temporary deviation followed by recovery), `mean_shift` (a persistent transition from the nearest preceding stable level), or `seasonal_change` (a change in a periodic pattern relative to the nearest preceding stable periodic region).
- `subtype`: `impulse`, `sustained`, or `recurrent` for excursions; null for mean shifts; `amplitude`, `phase`, `period`, `onset`, or `disappearance` for seasonal changes.
- `direction`: the direction relative to the local background or preceding reference regime. Its possible values depend on the family and subtype.
- `start`, `end`: the zero-based inclusive boundaries of the effect within the forecast horizon.
- `intensity`: a value computed deterministically from the raw series to quantify the effect. Its interpretation is family-specific.
- `amplitude_trend`: how the amplitude evolves within the effect: `increase`, `decrease`, `stable`, or `variable`.
- `envelope_change`: how the amplitude envelope evolves within the effect: `expanding`, `contracting`, `stable`, or `variable`.

To make quantitative fields comparable across series, Macro `level_range` and Micro `intensity` values use a shared history-normalized scale. Let $x_{\min}$ and $x_{\max}$ be the minimum and maximum finite history values. A normalized level v is represented as $\ell(v) = 100\frac{v - x_{\min}}{x_{\max} - x_{\min}}$. Values below 0 or above 100 are valid when the future falls outside the historical range. This normalization preserves relative scale while allowing one schema to cover datasets with different units and magnitudes. **Complete schemas and visualization are shown in Appendix B.1, B.2 and B.3.**

3.4 Structured Effects Generation Pipeline

Overview. We propose an agentic pipeline with an inspect–propose–validate–accept/revise workflow. As shown in Figure 1, the pipeline first runs independent Macro and Micro Discovery Agents, each combining LLM reasoning with deterministic code validation. The LLMs identify informative regions, determine effects' mechanism and boundaries, and submit proposals. Deterministic validation then checks the proposed type and type-specific criteria. If a proposal fails validation, the tool returns the numerical evidence so that the Agent can revise its proposal. Finally, a Joint Critic Agent reviews the outputs of both branches and filters out incorrect annotations. Moreover, a random subset of annotated samples is manually reviewed to assess annotation quality.

Macro Discovery Agent. The Macro Agent observes the complete history and future ground truth. It first examines the raw series and calls deterministic tools to measure trend, change-point, and periodic structure. Based on this evidence, it proposes a complete Macro object covering the forecast horizon. A Macro validator checks the schema and segment durations and computes the exact future range from the raw series. Measurements that disagree with the proposed trend or seasonality are returned as evidence for reconsideration. The Agent can then revise an invalid or poorly supported proposal before submitting its final Macro candidate.

Micro Discovery Agent. For each sample, we construct a reference forecast by averaging the best-performing TSFMs' predictions. The Micro Agent observes the history, future ground truth, reference forecast, and forecast residual. It first calls a deterministic overview tool that summarizes where the reference forecast differs substantially from the future ground truth. The Agent then calls inspection tools to read the raw values in each selected region and its surrounding context. From this evidence, it determines whether the region contains an independent effect and, if so, identifies

its mechanism and complete boundaries. It then submits a proposal specifying the family, subtype, direction, interval, and any preceding reference region required by the mechanism.

Each proposal is submitted to a family-specific Micro validator through a tool call. The validator checks the schema, boundaries, and any required component related to the Micro effect. When a proposal fails these checks, the tool returns the relevant numerical evidence so that the Agent can inspect the region again and revise its interpretation. The Agent repeats this process to identify additional effects.

Blind-first Joint Critic Agent. After both discovery branches finish, a Joint Critic Agent reviews their outputs. The Critic first inspects the raw history and future while all outputs of the discovery agents remain hidden. Only after forming an independent interpretation can it view the proposed Macro and Micro effects, their explanations, and the corresponding numerical evidence. This order reduces the risk that the Critic simply adopts the discovery Agents' interpretations.

The Critic assesses the Macro claims and decides whether each Micro candidate should be supported, excluded, or left unresolved. Supporting a Micro candidate requires inspection of its complete interval and any surrounding region on which the effect depends. The Critic can also correct a candidate's type or boundaries, after which the revised candidate is checked again by the validator. The final annotation contains the reviewed Macro and only the supported Micro effects; uncertain candidates are not promoted to the final annotation.

4 Experiments

4.1 Experimental Setup

Dataset. The final GIFT-Eval-CTX benchmark contains 10,000 instances from 20 GIFT-Eval source datasets and 78 dataset–frequency–term configurations. It covers six sampling frequencies and seven distinct horizon lengths between 30 and 900 steps. Short horizons account for 6,559 instances, medium horizons for 2,135, and long horizons for 1,306. Detailed source-level counts are deferred to Appendix A. The benchmark contains 23,476 Micro effects. Excursions are the most common family, with 19,134 effects, followed by 3,117 seasonal changes and 1,225 mean shifts.

Evaluated methods. Our experiments cover three classes of forecasting methods. The TSFM baselines are Chronos-2 (Ansari et al., 2025), Toto-2 (Khwaja et al., 2026), Moirai-2 (Liu et al., 2025a), and TimesFM-3 (Das et al., 2024), each operating in the univariate setting using only the numerical history. GPT-5.6 Sol (OpenAI, 2026) and Claude Opus 4.8 (Anthropic, 2026) serve as frontier-LLM baselines and are tested under paired history-only and contextual conditions. Finally, we evaluate a GPT-5.6 Sol-powered MoiraiAgent (Salesforce, 2026) that receives the structured effects and can use TSFM forecasting (powered by Moirai-2) and code-execution tools.

Metric and aggregation. Following GIFT-Eval, we use mean absolute scaled error (MASE) as the evaluation metric, which normalizes forecast error by the in-sample error of a seasonal-naive predictor. For each of the 78 dataset–term groups, we first average the per-sample MASE of each method and divide it by the corresponding seasonal-naive group mean. We then take the geometric mean across groups to obtain the aggregate relative MASE. Lower values are better, and seasonal naive has a score of 1.0 by construction.

4.2 Main Results

Frontier LLMs still underperform TSFMs in history-only forecasting. Table 1 reports MASE for all evaluated methods. Despite advances in the general knowledge, frontier LLMs still face limitations in history-only time series forecasting, consistent with earlier findings on less capable LLMs (Merrill et al., 2024; Tan et al., 2024). Given history only, both frontier LLMs underperform every native TSFM in terms of overall MASE. Both models outperform seasonal naive at the short horizon, but this advantage disappears as the horizon increases. At the medium and long horizons, GPT-5.6 Sol performs 7.1% and 4.9% worse than seasonal naive, whereas Claude Opus 4.8 degrades more sharply, with errors increasing by 32.5% and 20.6%, respectively. In contrast, all TSFMs consistently outperform seasonal naive across all horizons.

Table 1: MASE on GIFT-Eval-CTX. "Effects" denotes the structured context supplied by the benchmark. The best/second-best result in each column is bold/underlined.

Method	Input	Short	Medium	Long	Overall
Seasonal naive	History	1.000	1.000	1.000	1.000
Chronos-2	History	0.764	0.803	0.848	0.795
Toto-2	History	0.765	0.811	0.851	0.798
Moirai-2	History	0.740	0.798	0.852	0.782
TimesFM-3	History	0.750	0.795	0.823	0.780
Claude Opus 4.8	History	0.967	1.325	1.206	1.109
Claude Opus 4.8	History + effects	0.537	1.059	1.143	0.776
GPT-5.6 Sol	History	0.851	1.071	1.049	0.953
GPT-5.6 Sol	History + effects	<u>0.417</u>	<u>0.728</u>	<u>0.777</u>	<u>0.564</u>
MoiraiAgent	History + effects	**0.398**	**0.623**	**0.604**	**0.497**

Structured effects substantially improve frontier LLM forecasts. Providing structured effects reduces overall MASE by 40.8% for GPT-5.6 Sol and 30.1% for Claude Opus 4.8, enabling both models to achieve performance comparable to or better than TSFMs. These gains indicate that frontier LLMs can translate structured information into more accurate numerical trajectories. For both models, the benefit diminishes as the forecast horizon grows, with the largest improvements occurring at the short horizon and progressively smaller gains thereafter. Structured effects therefore remain useful beyond short horizons, but direct realization becomes increasingly difficult.

MoiraiAgent achieves the best overall performance. MoiraiAgent not only achieves the lowest overall MASE, but also performs best at every horizon. Relative to the best history-only TSFM overall, MoiraiAgent reduces MASE by 36.3%. Its advantage over GPT-5.6 Sol with the same structured effects grows from 4.6% on short horizons to 14.4% on medium horizons and 22.3% on long horizons, suggesting that tool use becomes increasingly valuable as the forecast horizon grows.

4.3 Analysis of Structured Effects

Micro effects drive the largest gains, while Macro effects remain complementary. Table 2 compares history-only forecasting with three structured-effect conditions. Micro effects are the larger individual contributor for both methods, while the additional improvement from combining Macro and Micro effects shows that the two levels provide complementary information.

The horizon breakdown shows that Macro and Micro effects contribute differently. For both GPT-5.6 Sol and MoiraiAgent, the gains from Macro effects diminish as the forecast horizon grows, whereas Micro effects yield substantial improvements at all three horizon lengths. The reason is that Macro effects provide coarse, horizon-wide information. As the horizon grows, many distinct trajectories may satisfy the same global description, making these effects less informative for constructing a long-horizon forecast. Micro effects remain more actionable because they specify localized changes. Nevertheless, Macro effects remain useful in combination with Micro effects: the combined condition outperforms Micro-only forecasting for both methods at every horizon.

Mean shifts yield the largest gains, while seasonal changes remain challenging. To compare the influence of three Micro-effect families, we group samples according to whether they contain at least one effect of excursion, mean shift, or seasonal change. Samples containing effects from multiple families are assigned to each corresponding group. As shown in Table 3, both methods exhibit the same ordering of gains across the Micro families. Mean shifts produce the largest gains, reducing MASE by around two thirds. Excursions yield substantial but smaller reductions of approximately 28–31%, whereas seasonal changes improve performance by only 6–8% overall. The horizon breakdown shows that the gains from mean shifts and excursions are consistent across forecast lengths. Seasonal changes are more difficult: on long horizons, GPT-5.6 Sol deteriorates from 1.070 to 1.118 when Micro effects are provided, whereas MoiraiAgent improves from 0.826 to 0.758. This pattern suggests that persistent level changes are the most readily translated into numerical trajectories, while changes to periodic structure require coordinated adjustments across repeated cycles. The

Table 2: MASE under different structured-effect inputs. MoiraiAgent's history-only performance is the same as Moirai-2.

Method	Input	Short	Medium	Long	Overall
GPT-5.6 Sol	History only	0.851	1.071	1.049	0.953
	History + Macro	0.601	0.992	1.009	0.780
	History + Micro	0.524	0.798	0.813	0.653
	History + Macro + Micro	**0.417**	**0.728**	**0.777**	**0.564**
MoiraiAgent	History only	0.740	0.798	0.852	0.782
	History + Macro	0.605	0.792	0.848	0.711
	History + Micro	0.490	0.632	0.610	0.553
	History + Macro + Micro	**0.398**	**0.623**	**0.604**	**0.497**

Table 3: Comparison of MASE between history-only forecasting and forecasting with Micro effects. Each Micro family contains samples with at least one effect from that family.

Method	Micro family	Short		Medium		Long		Overall	
		Hist.	+Micro	Hist.	+Micro	Hist.	+Micro	Hist.	+Micro
GPT-5.6 Sol	Excursion	0.826	0.545	1.056	0.783	1.031	0.737	0.930	0.645
	Mean shift	0.810	0.241	0.995	0.448	0.982	0.304	0.890	0.295
	Seasonal change	0.857	0.753	1.104	1.037	1.070	1.118	0.975	0.917
MoiraiAgent	Excursion	0.708	0.505	0.811	0.631	0.862	0.585	0.770	0.555
	Mean shift	0.699	0.236	0.864	0.408	0.900	0.301	0.780	0.284
	Seasonal change	0.742	0.679	0.866	0.795	0.826	0.758	0.796	0.730

contrasting long-horizon results further indicate that tool-supported forecasting is more reliable than direct generation for realizing such distributed changes.

Micro effects improve forecasts both inside and between labeled regions. A successful forecast should follow each local instruction and connect separate effect regions through coherent numerical trajectories. We additionally examine how Micro effects change predictions outside their labeled regions. Table 4 therefore reports performance separately inside and outside the labeled Micro regions. Within the labeled regions, adding Micro effects reduces GPT-5.6 Sol's MASE by 50.6% and MoiraiAgent produces a similar 48.7% reduction, confirming that both methods can effectively use Micro effects to realize the specified local events.

Notably, both methods can propagate local effect information beyond the labeled regions. Outside the labeled regions, the overall MASE improves from 0.890 to 0.804 for GPT-5.6 Sol and from 0.687 to 0.653 for MoiraiAgent. Although smaller than the within-region gains, these improvements show that structured effects can positively influence the trajectory between local events. This also indicates that achieving strong performance on GIFT-Eval-CTX requires both realizing individual effects and connecting them through accurate and coherent numerical dynamics. The left panel of Figure 2 illustrates this behavior. The sample contains an upward sustained excursion followed by an upward mean shift. Supplying these two Micro effects reduces MASE from 0.261 to 0.124 in the unlabeled interval between them, producing a more accurate transition between the two events.

Micro effects can preserve periodic structure even when MASE worsens. The right panel of Figure 2 illustrates a complementary qualitative benefit that is not fully captured by the quantitative MASE metric. For this periodic electricity series, the history-only Moirai-2 forecast progressively attenuates the oscillations toward a nearly flat trajectory, exhibiting a common forecast-collapse behavior of TSFMs (Qiao et al., 2026). Given a seasonal-onset effect with stable amplitude, MoiraiAgent instead preserves meaningful periodic dynamics throughout the forecast horizon, although its sample-level MASE increases from 0.933 to 1.052. Thus, this kind of Micro effects can prevent a periodic forecast from collapsing and preserve meaningful periodic structure even when this benefit is not reflected in the quantitative results.

Table 4: MASE inside and outside labeled Micro regions.

Method	Input	Inside Micro regions				Outside Micro regions			
		Short	Medium	Long	Overall	Short	Medium	Long	Overall
GPT-5.6 Sol	History only	0.943	1.107	1.055	1.011	0.744	1.060	1.062	0.890
	History + Micro	0.416	0.654	0.543	0.500	0.643	0.918	1.091	0.804
MoiraiAgent	History only	0.838	0.893	0.939	0.876	0.610	0.729	0.820	0.687
	History + Micro	0.406	0.538	0.456	0.450	0.596	0.672	0.760	0.653

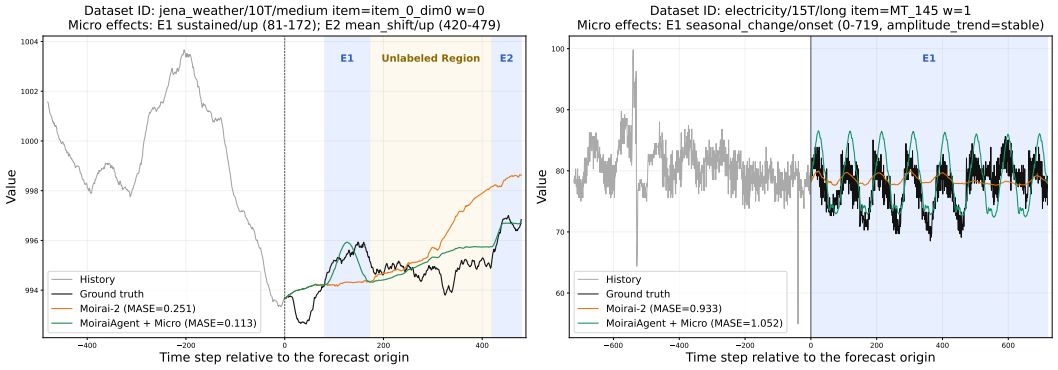


Figure 2: Qualitative examples of the benefits provided by Micro effects. Left: Micro effects improve the unlabeled trajectory between two labeled events. Right: Micro effects preserve meaningful periodic dynamics even when the sample-level MASE does not improve.

4.4 Further Analysis of MoiraiAgent

MoiraiAgent achieves the best performance on GIFT-Eval-CTX. In this section, we investigate the sources of its strong performance and identify potential limitations to guide future improvements.

MoiraiAgent uses forecast tools to ensure exact-length outputs. Beyond accuracy, direct LLM forecasting exhibits a fundamental limitation in following the forecasting task's output-length requirement. As the horizon grows, LLMs may lose track of the required output length during generation, causing many medium- and long-horizon outputs whose lengths do not match the requested horizon. In such cases, we truncate overlong outputs and pad shorter ones to the required horizon. Table 5 shows that this issue is rare on short horizons but common on longer ones. GPT-5.6 Sol can only produce the requested length in 38.8% of medium outputs and 40.9% of long outputs. Claude Opus 4.8 shows even weaker length control, with corresponding rates of only 2.5% and 1.3%.

MoiraiAgent avoids this problem by first calling Moirai-2 to produce an exact-length numerical trajectory. The LLM uses code-execution tools to modify this existing forecast instead of constructing a long, numerically coherent forecast from scratch. Consequently, all MoiraiAgent forecasts have the requested length. This suggests that forecasting tools contribute more than a strong history-only baseline: they also provide an exact-length numerical object that the LLM can directly edit.

MoiraiAgent uses code execution to improve numerical constraint adherence. To evaluate whether a forecast numerically realizes the supplied structured effects, we measure its adherence to the specified level range and Micro-event intensities using a tolerance of five history-normalized points. Table 5 shows that the global level range is generally easier to follow than local event intensities. GPT-5.6 Sol follows the level range on 97.44% of samples, but follows every specified intensity on only 78.62% of samples. Claude Opus 4.8 is substantially less reliable on both constraints.

MoiraiAgent strengthens numerical constraint adherence through code execution and achieves the highest following rates. The code tool allows the Agent to calculate the numerical targets implied by the level range and event intensities, modify the relevant forecast regions, and verify the resulting values against these constraints. Together, these results show how the two tools address comple-

Table 5: Analysis of output-length compliance and numerical constraint adherence. T/P denotes the truncation and padding rates, respectively. The final two columns report sample-level adherence to the specified level range and event intensities using a tolerance of five history-normalized points.

Method	Short (T/P)	Medium (T/P)	Long (T/P)	Follow level range	Follow intensity
Claude Opus 4.8	0.0% / 0.2%	48.2% / 49.3%	33.1% / 65.6%	59.61%	38.29%
GPT-5.6 Sol	0.0% / 0.1%	54.2% / 7.0%	51.2% / 7.9%	97.44%	78.62%
MoiraiAgent	0.0% / 0.0%	0.0% / 0.0%	0.0% / 0.0%	99.97%	89.72%

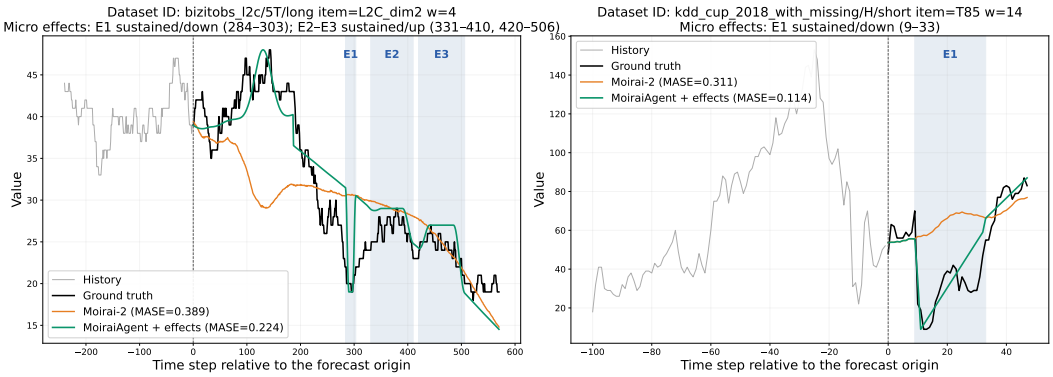


Figure 3: Qualitative limitations of MoiraiAgent. Left: MoiraiAgent replaces local fluctuations with extended plateaus in the BizITObs series. Right: it approximates the recovery from a sustained downward excursion with a nearly linear trajectory in the KDD Cup series. In both examples, sample-level MASE improves despite the loss of fine-grained temporal dynamics.

mentary limitations of direct LLM forecasting: the forecast tool provides an exact-length numerical starting point, while the code tool helps the Agent realize structured effects more precisely.

Limitations of MoiraiAgent. Code execution improves numerical constraint adherence, but it can also simplify the local dynamics of the forecast. MoiraiAgent often realizes an effect through deterministic operations such as level shifts, rescaling, and interpolation. These operations can place an event at the requested location and intensity, yet produce long constant segments or overly smooth linear transitions that do not capture the fine-grained temporal variation of time series.

Figure 3 shows two examples in which code-based editing fails to preserve realistic local dynamics. In the BizITObs series, MoiraiAgent reduces the sample-level MASE from 0.389 to 0.224, but represents the second and third sustained excursions with extended plateaus that suppress their local fluctuations. In the KDD Cup series, the MASE also improves, while the recovery from the sustained downward excursion is approximated by an almost perfectly straight line. Thus, future agentic forecasters should couple numerical constraint enforcement with more flexible mechanisms for preserving temporal variation during forecast editing. Another promising direction is to jointly train multimodal models on time series and structured effects, allowing them to represent and realize effect information natively within the forecasting model.

5 Conclusion

We introduced GIFT-Eval-CTX, a benchmark that isolates the numerical realization of verified future information in time series forecasting. The benchmark represents context as compositional Macro and Micro effects and uses an agentic pipeline with deterministic validation to construct annotations across diverse datasets. Experiments show that structured effects substantially improve frontier LLM forecasts, while MoiraiAgent achieved the best performance by combining LLM reasoning with forecasting and code-execution tools. Our analyses also show that success requires both realizing the specified effects and maintaining coherent numerical dynamics throughout the forecast. These findings motivate two directions: building stronger agentic forecasters, or jointly training models on time series and structured effects for native effect representation and realization.

AI USE STATEMENT

In this work, we used generative AI tools to generate the structured-effect annotations, assist in implementing the code for baseline evaluations, and support the analysis of experimental results. We did not use generative AI to develop theoretical models, formulate mathematical claims, or produce proofs. Additionally, generative AI assisted with manuscript drafting and language polishing. The authors manually inspected annotation quality, executed and tested the code, verified the experimental outcomes, and reviewed the manuscript. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

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

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

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

Anthropic. Introducing claude opus 4.8. `https://www.anthropic.com/news/claude-opus-4-8`, 2026. Published May 28, 2026.

Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*, 2025.

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

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024.

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

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

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, et al. Toto 2.0: Time series forecasting enters the scaling era. *arXiv preprint arXiv:2605.20119*, 2026.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025.

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

Jaehoon Lee, Suhwan Park, Taeyoon Lim, Seunghan Lee, Jun Seo, Dongwan Kang, Hwanil Choi, Minjae Kim, Sungdong Yoo, SoonYoung Lee, et al. Fintexts: Financial text-paired time-series dataset via semantic-based and multi-level pairing. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 9266–9277, 2026.

Jiafeng Lin, Yuxuan Wang, Jialong Wu, Huakun Luo, Zhongyi Pei, and Jianmin Wang. Thoth: Mid-training bridges llms to time series understanding. *arXiv preprint arXiv:2603.01042*, 2026.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025a.

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

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

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In *International Conference on Machine Learning*, 2025b.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3512–3533, 2024.

OpenAI. Previewing gpt-5.6 sol: A next-generation model. `https://openai.com/index/previewing-gpt-5-6-sol/`, 2026. Published June 26, 2026.

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's time: Towards the next generation of time series forecasting benchmarks. *arXiv preprint arXiv:2602.12147*, 2026.

Salesforce. Moiraiagent: An agentic framework for context-aware time-series forecasting. `https://www.salesforce.com/blog/moiraiagent/`, 2026. Published Jan 14, 2026.

Medhasweta Sen, Zachary Gottesman, Jiaxing Qiu, C Bayan Bruss, Nam Nguyen, and Tom Hartvigsen. Bedtime: A unified benchmark for automatically describing time series. *arXiv preprint arXiv:2509.05215*, 2025.

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

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

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

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, et al. Context is key: A benchmark for forecasting with essential textual information. In *International Conference on Machine Learning*, 2025.

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

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*, 2024.

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

Zhijian Xu, Wanxu Cai, Xilin Dai, Zhaorong Deng, and Qiang Xu. Fidel-ts: A high-fidelity multi-modal benchmark for time series forecasting. *arXiv preprint arXiv:2509.24789*, 2025b.

Yao Yin, Zhenyu Xiao, Musheng Li, Yiwen Liu, Sutong Nan, Yiting He, Ruiqi Wang, Zhenwei Zhang, Qingmin Liao, and Yuantao Gu. Mmts-bench: A comprehensive benchmark for time series understanding and reasoning. *arXiv preprint arXiv:2602.08588*, 2026.

Fangxu Yu, Xingang Guo, Lingzhi Yuan, Haoqiang Kang, Hongyu Zhao, Lianhui Qin, Furong Huang, Bin Hu, and Tianyi Zhou. Tsrbench: A comprehensive multi-task multi-modal time series reasoning benchmark for generalist models. *arXiv preprint arXiv:2601.18744*, 2026.

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

A Detailed Benchmark Statistics

Instance distribution. Table 6 summarizes the overall benchmark statistics. GIFT-Eval-CTX contains 10,000 univariate forecasting instances drawn from the held-out test regions of 20 GIFT-Eval source datasets. A source dataset may appear at multiple sampling frequencies, giving 38 distinct dataset–frequency identifiers. Combining these identifiers with the available short-, medium-, and long-horizon settings produces 78 dataset–frequency–term configurations. Table 7 provides a source-level breakdown, where *IDs* denotes the number of dataset–frequency identifiers and *Groups* denotes the number of dataset–frequency–term configurations. Table 8 reports the number of instances at each sampling frequency and forecast horizon.

Table 6: Summary statistics for GIFT-Eval-CTX.

Statistic	Value
Instances	10,000
Source datasets	20
Dataset–frequency identifiers	38
Dataset–frequency–term configurations	78
Sampling frequencies	6
Distinct forecast horizons	7
Horizon range	30–900
Short / medium / long instances	6,559 / 2,135 / 1,306
Micro effects	23,476
Mean Micro effects per instance	2.35

Table 7: Source-level composition of GIFT-Eval-CTX.

Source dataset	IDs	Groups	Short	Medium	Long	Total
LOOP_SEATTLE	3	7	1,332	685	306	2,323
M_DENSE	2	4	61	5	4	70
SZ_TAXI	2	4	279	86	73	438
bitbrains_fast_storage	2	4	799	257	112	1,168
bitbrains_rnd	2	4	681	100	97	878
bizitobs_application	1	2	1	1	0	2
bizitobs_l2c	2	5	19	12	12	43
bizitobs_service	1	3	46	18	13	77
covid_deaths	1	1	23	0	0	23
electricity	3	7	1,184	707	502	2,393
ett1	3	7	28	21	20	69
ett2	3	7	31	25	23	79
hierarchical_sales	1	1	97	0	0	97
jena_weather	3	7	56	33	33	122
kdd_cup_2018_with_missing	2	4	572	18	11	601
m4_hourly	1	1	35	0	0	35
restaurant	1	1	130	0	0	130
saugeenday	1	1	1	0	0	1
solar	3	7	641	167	100	908
temperature_rain_with_missing	1	1	543	0	0	543
Total	38	78	6,559	2,135	1,306	10,000

Structured-effect distribution. Each benchmark instance contains one Macro annotation and at least one Micro effect. In total, the benchmark contains 23,476 Micro effects, with a median of two and a maximum of 15 effects per instance. Excursions constitute 19,134 effects, seasonal changes 3,117, and mean shifts 1,225. Table 9 provides the complete subtype breakdown.

Table 8: Distribution of sampling frequency and forecast horizon.

Frequency	Instances	Horizon	Instances
10S	79	30	1,791
5T	2,724	48	4,721
10T	543	60	47
15T	1,597	480	2,116
H	3,266	600	19
D	1,791	720	1,293
		900	13

Table 9: Distribution of Micro families and subtypes.

Family	Subtype	Count
Excursion	Impulse	10,625
	Sustained	6,750
	Recurrent	1,759
Mean shift	–	1,225
Seasonal change	Amplitude	2,391
	Phase	29
	Period	24
	Onset	211
	Disappearance	462

B Complete Structured Effects Schema

B.1 Macro Schema

Macro describes the complete forecast horizon. Table 10 lists its fields and constraints.

Table 10: Complete Macro schema.

Field	Type and allowed values	Definition
`trend`	List of one or two segment objects	Chronological decomposition of the horizon-wide center-line movement. Segment durations are positive and sum to one.
`trend.direction`	`up`, `down`, or `flat`	Direction of the center line within the segment. Flat means there is no meaningful overall rise or fall.
`trend.shape`	`linear`, `curved`, `accelerating_curved`, `decelerating_curved`, `step`, or null	Shape of a non-flat trend segment. The value is null when direction is `flat`.
`trend.duration`	Positive real number in $(0, 1]$	Fraction of the forecast horizon occupied by the segment.
`seasonality`	Object or null	Dominant oscillatory structure of the future. It is null when no meaningful oscillation is present.
`seasonality.period`	Integer greater than one, or null	Dominant period in time steps. A null period denotes an irregular oscillation.
`seasonality.waveform`	`follow_history`, `sinusoidal`, `sawtooth`, or `irregular`	Qualitative shape of the dominant oscillation.
`level_range`	Two finite real values $[\ell_{\min}, \ell_{\max}]$	Minimum and maximum future values in history-normalized coordinates. The field is computed deterministically from the complete future.

B.2 MICRO SCHEMA

A Micro entry describes one interval-specific mechanism. Multiple entries may overlap when they represent different compatible mechanisms. Each entry contains the following fields:

- `family`: the top-level mechanism: `excursion`, `mean_shift`, or `seasonal_change`.
- `subtype`: the specific form of the mechanism. Its valid values depend on the family.
- `direction`: the direction of the change relative to the locally reasonable background.
- `amplitude_trend`: how component or cycle amplitudes evolve within the interval: `increase`, `decrease`, `stable`, `variable`, or null.
- `envelope_change`: how the overall envelope of a bidirectional recurrent pattern evolves: `expanding`, `contracting`, `stable`, `variable`, or null.
- `start` and `end`: zero-based, inclusive boundaries within the future horizon.
- `intensity`: a deterministic, family-specific numerical measurement.

Shared normalization. Let $x_{\min}$ and $x_{\max}$ be the minimum and maximum finite history values, and let $R_x = x_{\max} - x_{\min}$. A normalized level v is represented as $\ell(v) = 100(v - x_{\min})/R_x$, and the inverse transformation is $v = x_{\min} + \ell(v)R_x/100$. Values below 0 or above 100 are valid when the future lies outside the historical range. Macro `level_range` is $[\ell(\min_t y_t), \ell(\max_t y_t)]$. Micro level intensities use the same coordinates.

Excursion. An excursion is a temporary departure from the locally reasonable background followed by a meaningful recovery.

- **Subtype:** `impulse` is one isolated, sharply concentrated peak or trough; `sustained` is one continuous deviation that persists locally before recovering; and `recurrent` represents multiple separated events with quiet intervals between them rather than a stable seasonal cycle.
- **Direction:** `impulse` and `sustained` are `up` or `down`; `recurrent` is `up`, `down`, or `bidirectional`.
- **Amplitude fields:** for a unidirectional recurrent excursion, `amplitude_trend` is `increase`, `decrease`, `stable`, or `variable`, and `envelope_change` is null. For a bidirectional recurrent excursion, `envelope_change` is `expanding`, `contracting`, `stable`, or `variable`, and `amplitude_trend` is null. Both fields are null for `impulse` and `sustained`.
- **Boundaries:** when available, `start` includes the point immediately before the first affected point so that the onset is visible. `end` includes the first clear recovery point when recovery occurs within the horizon.
- **Intensity:** for `impulse` and `sustained`, intensity is the interval maximum for `up`, or the interval minimum for `down`, expressed in history-normalized level coordinates. Thus it is $\ell(\max_{t\in[a,b]} y_t)$ for `up` and $\ell(\min_{t\in[a,b]} y_t)$ for `down`. For `recurrent`, intensity encodes a representative component amplitude $A^\star$ relative to the local background, where $A^\star = \texttt{intensity}\, R_x/100$. It is the largest component amplitude when `amplitude_trend` is `increase` or `decrease`, or when `envelope_change` is `expanding` or `contracting`; otherwise, it is the median component amplitude.

Mean shift. A mean shift is a transition from the nearest preceding stable level into a persistent new level. Its `subtype` is null, and its `direction` is `up` when the settled new level is above the preceding stable level and `down` when it is below. Both `amplitude_trend` and `envelope_change` are null. Intensity records the settled new level as $100(\widetilde{y} - x_{\min})/R_x$, where $\widetilde{y}$ is the median settled level.

Seasonal change. A seasonal change describes a change in a periodic pattern relative to the nearest stable periodic region immediately preceding `start`. There are five subtypes in this family: `amplitude`, `phase`, `period`, `onset`, and `disappearance`. The subtype and its corresponding direction meaning is:

- `amplitude`: `increase` direction means that cycle amplitudes are larger than in the preceding stable region, and `decrease` means that they are smaller.
- `phase`: `advance` direction means that the recurring pattern occurs earlier within each cycle, and `delay` means that it occurs later.
- `period`: `increase` direction means that cycles become longer, and `decrease` means that they become shorter.
- `onset`: a periodic pattern appears after being absent from the preceding stable region; `direction` is null.
- `disappearance`: a periodic pattern present in the preceding stable region disappears; `direction` is null.

For `amplitude` and `onset` subtypes, the field of `amplitude_trend` describes how cycle amplitudes evolve inside the labeled interval. It is required when the interval contains at least two complete cycles and is null when it contains exactly one. For example, `seasonal_change/amplitude/decrease` with `amplitude_trend=increase` means that amplitudes are smaller than in the preceding stable region but increase within the labeled interval. For `phase`, `period`, and `disappearance`, the field of `amplitude_trend` is null, and the field of `envelope_change` is null for every seasonal-change subtype.

Intensity is defined separately for each seasonal-change subtype:

- For `amplitude` or `onset`, intensity encodes a representative peak-to-trough cycle amplitude $A^\star$, with $A^\star = \texttt{intensity}\, R_x/100$. If the interval contains one complete cycle, it is that cycle's amplitude. With multiple complete cycles, it is the maximum amplitude when `amplitude_trend` is `increase` or `decrease`, and the median when it is `stable` or `variable`.
- For `phase`, intensity is $100\Delta/P_{\mathrm{ref}}$, where Δ is the shortest circular phase shift and P_{ref} is the preceding period. The direction specifies whether the waveform advances or is delayed.
- For `period`, intensity is $100|P_{\mathrm{new}} - P_{\mathrm{ref}}|/P_{\mathrm{ref}}$. The direction specifies whether the period increases or decreases.
- For `disappearance`, intensity is null.

B.3 Micro Effect Examples

Figure 4 visualizes representative examples from the Micro-effect taxonomy. Panels (a)–(d) show the three excursion subtypes: impulse, sustained, and recurrent. Panel (c) contains upward recurrent excursions whose amplitudes increase, while panel (d) contains bidirectional recurrent excursions with a variable envelope. Panels (e) and (f) show upward and downward mean shifts. Panels (g)–(l) cover the seasonal-change subtypes. Panels (g) and (h) distinguish change relative to the preceding periodic region from amplitude evolution within the labeled interval: panel (g) has increased and stable amplitudes, whereas panel (h) has decreased amplitudes that subsequently increase. Panels (i)–(l) show a phase delay, period increase, seasonal onset, and seasonal disappearance, respectively.

C Benchmark Construction

C.1 Time Series Corpus Selection

We begin with GIFT-Eval test windows whose forecast horizons fall between 24 and 1,000 steps. Multivariate collections are expanded into independent univariate targets. No candidate is cut from a training or validation region. For a horizon of length H, the most recent $10H$, $5H$, and $3H$ historical observations are retained for short-, medium-, and long-horizon configurations, respectively. A

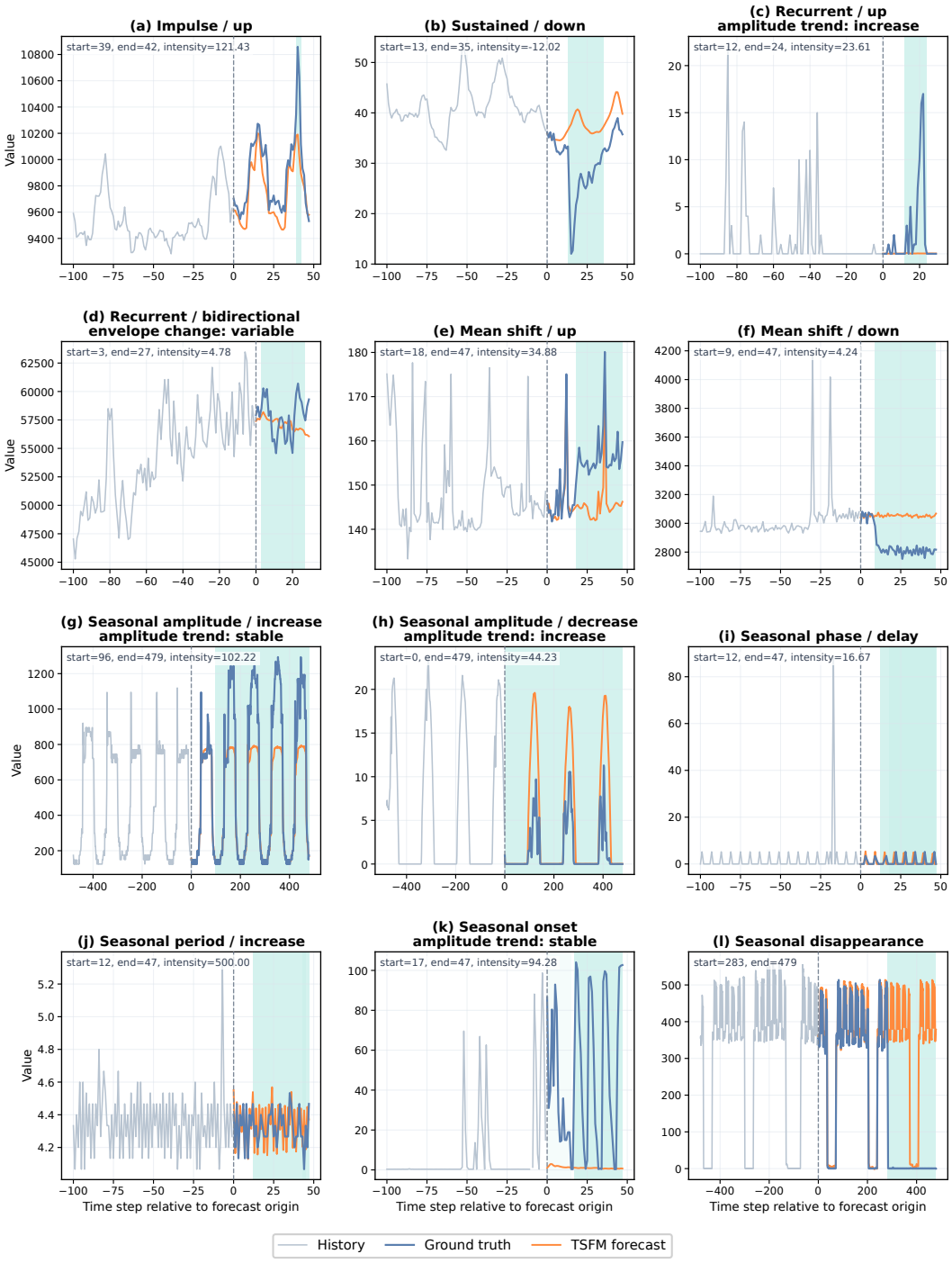


Figure 4: Representative examples of the Micro-effect taxonomy. Shaded regions indicate the inclusive intervals described by the effects.

candidate is discarded if its future contains a missing value, more than 20% of its history is missing, or the finite history or future is effectively constant, defined by a standard deviation below 10^{-8}.

Each candidate is forecast independently by top-performing models on GIFT-Eval. We use the median prediction from each model and compute $\mathrm{rMAE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{H^{-1}\sum_{t=1}^{H}|\hat{y}_t - y_t|}{\max_t y_t - \min_t y_t}$. A candidate is retained only if all models obtain $\mathrm{rMAE} \geq 0.10$. Thus, each model's mean absolute error must be

at least 10% of the future value range. This deliberately removes windows that are too easy for the TSFMs. After retaining a window, we skip enough subsequent windows from the same univariate target to ensure that the complete history–future spans of retained instances do not overlap. This prevents similar windows from entering the subsequent annotation process. To prevent a few large sources from dominating the benchmark, medium- and long-horizon configurations are capped at approximately 600 and 300 retained windows, respectively. The largest short-horizon configurations receive source-specific caps, and capped sources are sampled with a fixed seed of 42.

C.2 Structured Effects Generation

Macro generation. The Macro Agent receives the complete history and future ground truth. It can inspect deterministic measurements of trend, change points, and periodic structure before proposing a complete Macro object. The Macro validator enforces the schema, checks that trend durations sum to one, and replaces the proposed `level_range` with the value computed directly from the future. Trend and seasonality measurements are returned as evidence for reconsideration when they disagree with a proposal.

Micro generation. For each retained instance, we define a TSFM reference forecast as the pointwise average of the top-performing TSFMs' predictions. The Micro Agent receives the history, future, reference forecast, and forecast residual relative to the ground truth. It first calls an overview tool to identify regions where the reference forecast differs substantially from the future. These candidate regions guide the Agent's subsequent inspection. For each candidate region, the Agent calls an inspection tool to retrieve the exact raw values and surrounding context. Based on this evidence, the Agent determines whether the region contains a distinct mechanism and, if so, proposes its family, subtype, direction, and complete interval. Mean shifts and seasonal changes additionally use the nearest stable region immediately preceding the label. Reference coordinates may lie in history, cross the history–future boundary, or lie in the future. Seasonal comparisons use complete cycles rather than isolated peaks.

Deterministic Micro validation. Each proposal is checked against the raw time series by a family-specific deterministic validator. The validator verifies that the proposed mechanism and boundaries are supported by numerical evidence. It also requires the labeled region to coincide with a substantial error in the TSFM reference forecast. Proposals that lack sufficient evidence are rejected. For accepted proposals, the validator computes the family-specific intensity defined in Appendix B.2; otherwise, it returns numerical evidence that allows the Agent to inspect the region again and revise its proposal.

Blind-first joint review. After both discovery branches finish, the Joint Critic Agent first inspects the raw history and future while all proposals remain hidden. It then views the Macro and Micro candidates, their rationales, and validator evidence. For each Micro candidate, the Critic may support, exclude, or revise the proposal; revised candidates pass through the same Micro validator as above. Only supported Micro effects enter the final annotation. This ordering reduces anchoring on discovery outputs and ensures that the final decision is based on an independent reading of the raw series. We additionally assess annotation quality by manually reviewing a random subset of annotated samples.

D Experimental Details

D.1 Model Configurations

Table 11 summarizes the method configurations. All TSFMs operate in the univariate setting and return their median point forecast. The frontier LLMs are accessed through API calls, allowing up to 16,000 output tokens and using medium reasoning effort. MoiraiAgent combines the API-accessed GPT-5.6 Sol with an agent harness built around the public Moirai-2 checkpoint. Concretely, MoiraiAgent reads the numerical history from a session-local CSV file. It first calls the Moirai-2 forecasting tool to obtain an exact-length numerical trajectory, then uses the supplied effects and code-execution tools to inspect and modify that trajectory. The final forecast must be written as a finite, exact-length JSON array. Each run permits at most ten Agent steps and 16,000 output tokens.

Table 11: Model configurations used in the experiments.

Method	Model or checkpoint	Forecasting interface
Seasonal naive	–	Repeats the last observed seasonal cycle
Chronos-2	`amazon/chronos-2`	Direct median forecast
Toto-2	`Datadog/Toto-2.0-313m`	Direct median forecast
Moirai-2	`Salesforce/moirai-2.0-R-small`	Direct median forecast
TimesFM-3	`google/timesfm-3.0-pytorch`	Direct median forecast
GPT-5.6 Sol	GPT-5.6 Sol	Direct numerical generation
Claude Opus 4.8	Claude Opus 4.8	Direct numerical generation
MoiraiAgent	GPT-5.6 Sol + Moirai-2	Forecast and code-execution tools

D.2 LLM PROMPTS

We use separate system prompts for the history-only and history-with-effects conditions. The prompt construction is given below.

History only. The system message is:

> You are a time series forecasting expert. You will be given historical values of a target variable and the forecast horizon. Analyze the history to identify its trend, seasonality, level, and noise, then generate exact numerical forecast values for the specified horizon. The forecast length must equal the specified horizon. Return a JSON object with exactly one key, {`"forecast": [<H floats>]`}, and no other text.

The user message is instantiated as:

> Forecast horizon: {`H`} steps
> Historical values: $[x_1, \ldots, x_T]$
>
> Respond with only the JSON object {`"forecast": [...]`}. The forecast array must contain exactly {`H`} numerical values—no more and no fewer. Count and verify the values before finishing. No analysis or prose.

We repeat the exact-length requirement at the end of the user message because initial experiments showed that direct LLMs frequently failed to return the requested number of forecast values.

History + effects. The system message is:

> You are a time series forecasting expert. You will be given historical values of a target variable, the forecast horizon, and structured effects describing how the future horizon will behave. The structured effects contain two complementary parts: Macro describes the overall structure of the complete future horizon, while Micro describes interval-specific mechanisms. Treat the structured effects as reliable guidance and use them together with the historical values to produce the future numerical values.
> *[Macro schema and definitions]*
> *[Micro schema and definitions]*
>
> Study the history to understand the scale, recent level, and the shape and amplitude of any oscillation so that the forecast connects naturally to the history. Produce a forecast that makes full use of both parts of the structured effects: Macro describes the overall behavior of the future horizon, and Micro describes interval-specific mechanisms. The forecast length must equal the specified horizon. Return a JSON object with exactly one key, {`"forecast": [<H floats>]`}, with no analysis, prose, Markdown, or code fences.

The corresponding user message is:

> Forecast horizon: {`H`} steps
> Historical values: $[x_1, \ldots, x_T]$
>
> Structured effects (describing future behavior):
> {`"macro": {<Macro object>}, "micro": [<Micro objects>]`}

> For this sample, the Macro `level_range` corresponds to the inclusive raw-value range `[{L}, {U}]` over the forecast horizon. Every forecast value must remain within this range.
>
> Respond with only the JSON object `{"forecast": [...]}`. The forecast array must contain exactly `{H}` numerical values—no more and no fewer. Count and verify the values before finishing. No analysis or prose.

We additionally provide `level_range` in raw-value units because direct LLMs sometimes miscalculate the conversion from normalized coordinates, resulting in disproportionately large forecast errors.

D.3 Metrics and Aggregation

Let m be the seasonal period associated with the sampling frequency. For a sample with history $\mathbf{x}_{1:T}$, future $\mathbf{y}_{1:H}$, and forecast $\hat{\mathbf{y}}_{1:H}$, we compute

$$\mathrm{MASE} = \frac{H^{-1}\sum_{t=1}^{H}|y_t - \hat{y}_t|}{(T-m)^{-1}\sum_{t=m+1}^{T}|x_t - x_{t-m}|}.$$

If $m \geq T$, the denominator falls back to the one-step naive error. Seasonal periods follow the GluonTS frequency mapping. The seasonal-naive forecast repeats the last complete seasonal cycle, or the historical mean when the history is shorter than one cycle.

Aggregation follows GIFT-Eval. For each of the 78 dataset–frequency–term groups, we first take the arithmetic mean of per-sample MASE for each method. We divide this group mean by the corresponding seasonal-naive group mean, then take the geometric mean of the resulting ratios.

E Additional Related Work

E.1 Time Series Foundation Models and Multimodal Models

Time series foundation models have achieved strong zero-shot forecasting performance (Woo et al., 2024; Liu et al., 2025b; Ansari et al., 2024), but their standard interfaces operate primarily on numerical values and cannot directly incorporate multimodal context. Multimodal time series methods address this limitation through several modeling strategies. HORAI uses large-scale multimodal pretraining to incorporate endogenous and exogenous information (Chen et al., 2026). ChatTS, ChatTime, and OpenTSLM adapt LLMs to represent and reason over time series alongside language (Xie et al., 2024; Wang et al., 2025; Langer et al., 2025). More recent systems, including Thoth, TimeOmni-1, and TimeOmni-VL, extend this direction toward broader time series reasoning and unified understanding and generation (Lin et al., 2026; Guan et al., 2026a;b).

Many of these LLM-based approaches build on small open-weight LLMs to keep training feasible. These base models generally offer less world knowledge and weaker reasoning and agentic capabilities than frontier LLMs. Systems built on them may therefore be insufficient for real-world contextual forecasting, where relevant information can be broad, heterogeneous, domain-specific, and semantically complex. GIFT-Eval-CTX instead evaluates frontier LLMs and agentic systems built around them.