KAIROSAGENT: Agentic Time Series Forecasting with Fused Semantic Reasoning

Anonymous ACL submission

Abstract

Cross-domain multimodal time series forecasting is a challenging task, requiring models to integrate precise numerical comprehension, cross-domain semantic understanding, and effective multimodal fusion. Existing approaches either build Time Series Foundation Models (TSFMs) from scratch or leverage pretrained Large Language Models (LLMs). However, TSFMs often overlook semantic understanding and lack the ability to perform future-oriented semantic reasoning, and LLMs struggle with numerical comprehension and accurate quantitative forecasting. To overcome these limitations, we propose KAIROSAGENT, a novel agentic framework for multimodal time series forecasting, including an LLM-based reasoner and a TSFM-based forecaster. KAIROSAGENT unifies textual reasoning and numerical forecasting by dynamically invoking analytical tools to enhance the numerical understanding and semantic reasoning capabilities of LLMs. The reasoning results are subsequently fused into the TSFM pipeline, enabling more accurate and reliable future predictions. To further improve the reasoning, we curate a large-scale corpus of high-quality trajectories, alongside a reinforcement learning from forecasting paradigm with multi-turn refinement and turn-level credit assignment. Experiments demonstrate that KAIROSAGENT achieves superior zero-shot forecasting performance while maximizing the utility of pretrained LLMs and TSFMs, presenting a promising direction for efficient and interpretable time series agents.

1 Introduction

Cross-domain multimodal time series forecasting aims to predict future sequences using historical data and domain-specific texts (Wu et al., 2025b). This task requires models to capture precise numerical dynamics and understand semantic information across diverse domains. Existing methods fall into two paradigms. Time Series Reasoning Models (TSRMs) (Guan et al., 2025; Wu et al., 2025a) leverage Large Language Models (LLMs) (Achiam et al., 2023) for strong text interpretation. However, their reliance on standard tokenizers fragments continuous values, often suffering from numerical hallucinations and suboptimal quantitative forecasting (Wu et al., 2026; Ye et al., 2024). Conversely, Time Series Foundation Models (TSFMs) (Ansari et al., 2024; Woo et al., 2024) and their multimodal variants (Wu et al., 2025b) retain the native numerical modality for robust zero-shot forecasting. However, driven primarily by black-box statistical fitting, they lack the semantic understanding necessary to deduce future temporal patterns. Therefore, synergizing the semantic understanding with the numerical precision remains a critical challenge.

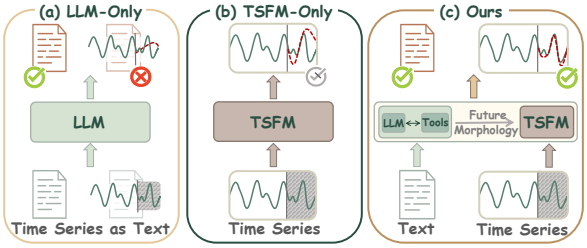

Figure 1: Comparison of forecasting paradigms. Unlike (a) *LLM-Only* models that suffer numerical hallucinations and (b) *TSFM-Only* models that lack interpretability, (c) KAIROSAGENT bridges the modality gap.

To bridge semantic understanding and numerical prediction, recent studies deploy LLMs within agentic frameworks (compared in Table 1). However, they still struggle with complex time series due to three critical limitations: (i) *Inappropriate Numerical Serialization*: Current methods (Jalori et al., 2025) typically feed raw numerical sequences directly to LLMs as text. Given that LLMs are not inherently designed for precise numerical computation, this strategy fails to extract reliable statistical features, limiting their analytical depth and precise forecasting. (ii) *Modality Disconnect*: In

Table 1: Comparison of KAIROSAGENT with existing models and frameworks.

Model / Framework	Primary Tasks	Tool-Calling	Multimodal Fusion	RL Optimization
Time Series Foundation Models	Forecasting	✗	✗ / ✓	✗ / ✓ (Outcome-Level)
Time Series Reasoning Models	Reasoning	✗	✗	✗ / ✓ (Outcome-Level)
TimeART (Wu et al., 2026)	Reasoning	✓	✗	✗
Cast-R1 (Tao et al., 2026)	Forecasting	✓	✗	✓ (Outcome-Level)
KAIROSAGENT (Ours)	**Reasoning & Forecasting**	✓	✓	✓ **(Process-Level)**

existing agentic frameworks (Cao et al., 2025; Tao et al., 2026), the underlying forecasting models rely exclusively on the time series modality, disregarding the semantic insights produced by LLMs. This structural isolation of semantic priors prevents genuine multimodal fusion. (iii) *Optimization Difficulty*: Current training paradigms face challenges in facilitating reliable reasoning. Approaches relying solely on Supervised Fine-Tuning (SFT) (Cui et al., 2025) imitate annotated trajectories, limiting generalization beyond the training distribution. Conversely, Reinforcement Learning (RL) with outcome-only rewards (Tao et al., 2026) suffers from severe reward sparsity, preventing effective credit assignment for intermediate reasoning steps and hindering the model from learning proper exploration and semantic deduction.

To address these limitations, we propose KAIROSAGENT, an agentic multimodal time series forecasting framework comprising an LLM-based reasoner and a TSFM-based forecaster (Figure 1). Specifically, unlike conventional paradigms that naively serialize time series, KAIROSAGENT employs a dynamic tool-calling mechanism to comprehensively profile the statistical features of historical sequences, enabling reliable numerical understanding and deduction of future morphology of time series in text, which is generalizable across different domains. To fully exploit this multimodal information during forecasting, we encode these textual deductions into robust semantic priors and deeply fuse them into the TSFM pipeline. Ultimately, KAIROSAGENT effectively unifies semantic reasoning with numerical forecasting.

To fully facilitate the reasoning and forecasting capabilities of the proposed agentic framework, we further introduce systematic contributions at both the data and training levels. (i) *Data Level*: To ensure KAIROSAGENT genuinely masters time series reasoning rather than relying on rote memorization, we curate the **T**ime-**S**eries **T**ool-**A**ugmented **R**easoning (`T-STAR`) Corpus. Comprising over $40k$ rigorously filtered, high-quality reasoning trajectories, this corpus establishes a robust foundation for reliable multimodal understanding and future deduction across different domains. (ii) *Training Level*: We train KAIROSAGENT via a three-stage pipeline. First, we warm up the tool-calling and morphology-deduction capabilities of the LLM-based reasoner through SFT. Second, we perform multimodal fusion training to enable the time series forecaster to incorporate morphology descriptions into numerical prediction. Finally, we refine the reasoner with RL using turn-level credit assignment, which improves reasoning behavior beyond mere trajectory imitation or sparse outcome rewards. Together, these stages enhance the reliability, interpretability, and numerical accuracy of multimodal time series forecasting across diverse domains.

In summary, the primary contributions of this work are three-fold:

- We propose KAIROSAGENT, an agentic multimodal time series forecasting framework that bridges semantic reasoning and numerical forecasting by deducing future morphology and integrating it into the forecasting pipeline.
- We curate `T-STAR`, a $40k$-trajectory time series reasoning corpus with tool augmentation, and introduce turn-level credit assignment for RL to provide fine-grained supervision beyond sparse outcome-level rewards.
- Extensive experiments show that KAIROSAGENT improves morphology reasoning and achieves strong zero-shot forecasting against competitive TSFMs and full-shot baselines.

2 Related Work

Foundation Models for Time Series Analysis. The application of foundation models in time series analysis has broadly bifurcated into reasoning-centric and forecasting-centric paradigms. Reasoning-centric approaches, such as TSRMs (Kong et al., 2025; Xie et al., 2025; Bazaga et al., 2025), project temporal data into the

textual space. While preserving LLM Question-Answering (QA) capabilities, discarding the native modality inevitably causes numerical hallucinations and degrades forecasting performance. Conversely, forecasting-centric models, including TSFMs (Feng et al., 2025; Liu et al., 2025c; Auer et al., 2026) and multimodal variants (Wu et al., 2025b), retain the native modality for strong zero-shot forecasting. However, driven primarily by black-box statistical fitting, they lack transparent reasoning and interpretability. Rather than compromising between semantic QA and numerical precision, KAIROSAGENT reconciles both paradigms. By using an LLM to deduce future morphology and integrating this semantic prior directly into a TSFM pipeline, our framework achieves precise zero-shot forecasting with robust interpretability.

LLM-based Time Series Agents Recent advances have shifted time series analysis from static prediction to dynamic, agentic workflows, repositioning LLMs as meta-cognitive controllers. For instance, TimeART (Wu et al., 2026) and Cast-R1 (Tao et al., 2026) reformulate forecasting as sequential decision-making, equipping LLMs with tool-calling to extract statistical features. Beyond numerical tools, agents increasingly harness external contexts: Wang et al. (2024) dynamically filter unstructured news and FLAIRR-TS (Jalori et al., 2025) iteratively refines prompts via interacting agents. At the orchestration level, TSOrchestra (Cao et al., 2025) uses LLMs to optimize ensemble weights for multiple foundation models. Despite these advances, existing frameworks face critical structural and optimization bottlenecks. They either force LLMs to output raw quantitative forecasts, yielding suboptimal results, or strictly isolate semantic reasoning from numerical forecasting, preventing genuine multimodal fusion. Furthermore, sparse outcome-only optimization hinders effective credit assignment for intermediate reasoning. To address these limitations, KAIROSAGENT incorporates LLM-derived semantic deductions directly into the TSFM pipeline, achieving deep cross-modal synergy. Additionally, we overcome sparse optimization via turn-level credit assignment to guide the intermediate reasoning process.

3 Methodology

3.1 Problem Formulation

We denote the historical observations within a lookback window of length L as $\mathbf{X} = (x_1, x_2, \ldots, x_L) \in \mathbb{R}^L$, while c represents the optional textual context (e.g., dataset metadata or variable-level information).

Time Series Reasoning. Given $\mathbf{X}$ and textual context c, the goal is to generate a natural language explanation, formulated as $r = \pi_{\text{reason}}(\mathbf{X}, c)$, that analyzes the underlying patterns, trends, and potential future behavior of the time series.

Time Series Forecasting. Based on historical observations $\mathbf{X}$, this task aims to predict the subsequent H values $\hat{\mathbf{Y}} = (\hat{y}_{L+1}, \ldots, \hat{y}_{L+H}) \in \mathbb{R}^H$, where H denotes the forecasting horizon.

In this work, we unify both tasks into a cohesive *reason-then-forecast* pipeline. The reasoning output r serves as an intermediate semantic prior that bridges qualitative understanding and quantitative prediction. Specifically, the textual context c is processed exclusively by the reasoner to generate r, which then conditions the forecaster:

$$r = \pi_{\text{reason}}(\mathbf{X}, c, \mathcal{T}), \quad \hat{\mathbf{Y}} = f_{\text{forecast}}(\mathbf{X}, r), \tag{1}$$

where $\mathcal{T}$ denotes the set of time series analysis tools available to the agent.

3.2 Overall Framework

KAIROSAGENT is a modular reason-then-forecast framework that separates semantic reasoning from numerical prediction. Given a historical time series $\mathbf{X} = (x_1, \ldots, x_L)$ and textual context c, the system operates in two stages: (1) An LLM-based reasoner π_{reason} interacts with time series analysis tools to produce a *morphology description* r, a natural-language summary of anticipated future patterns; (2) A text-conditioned TSFM-based forecaster f_{forecast} encodes r as a semantic prior and fuses it into the pipeline to generate the final numerical prediction $\hat{\mathbf{Y}}$. This decomposition tasks the LLM with semantic reasoning, while leaving precise numerical generation to the native time series model. Figure 2 illustrates the overall architecture.

Stage I: Tool-Augmented Morphological Reasoning. Given the historical observations $\mathbf{X}$ and textual context c, the LLM-based reasoner π_{reason} interacts with the tool set $\mathcal{T}$ to inspect both global and local temporal structures (e.g., trend, periodicity, volatility, and regime changes; Appendix B.2). This interaction unfolds over *multiple turns*, with each turn comprising either a tool invocation followed by feedback, or the direct output of the final description. Based on the tool outputs and its

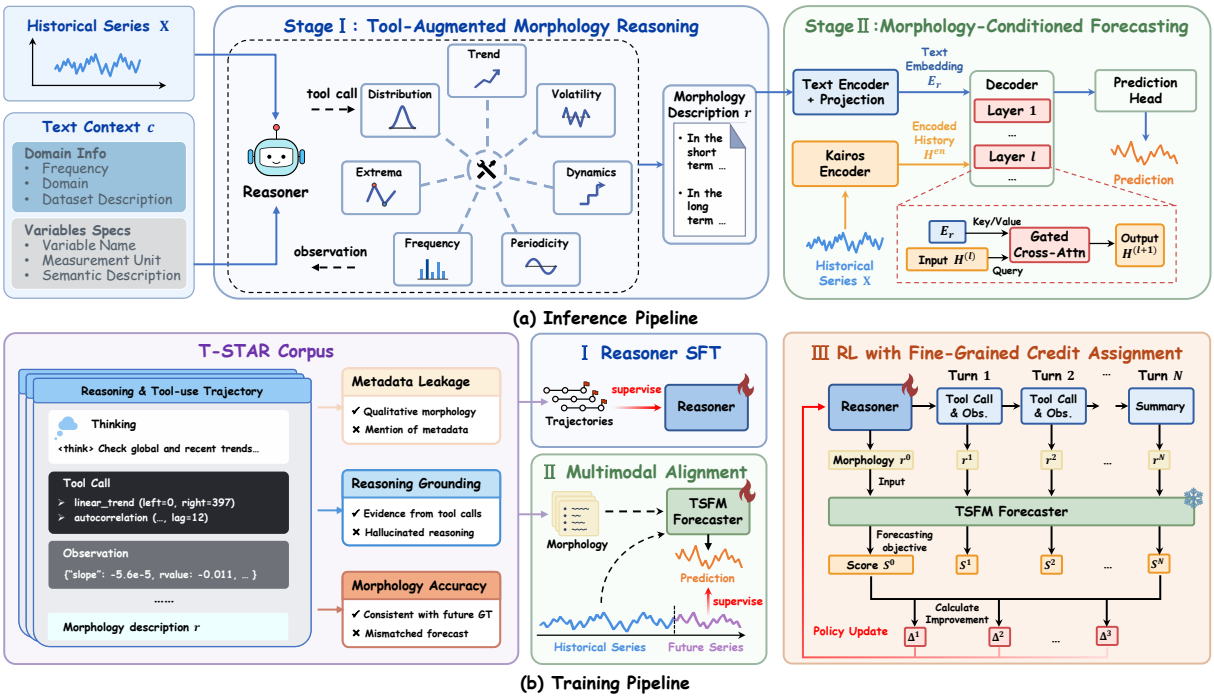

Figure 2: Overview of KAIROSAGENT. **(a)**: *inference pipeline* where the reasoner invokes statistical tools to produce a morphology description r, which is fused into the Kairos decoder to generate numerical forecasts. **(b)**: three-stage *training pipeline* where SFT initializes tool-calling capabilities, multimodal alignment trains the text-conditioned forecaster, and RL with turn-level credit assignment refines the reasoner.

own world knowledge, the reasoner synthesizes a compact morphology description r. Unlike vanilla LLMs that reason solely on serialized numerical inputs, our reasoner actively invokes analytical tools, grounding its reasoning in quantitative evidence. Crucially, r is intentionally semantic rather than numeric: it characterizes the qualitative evolution of the series without committing to specific point values. This design allows the reasoner to contribute high-level pattern reasoning while avoiding the numerical hallucination problem. Example rollouts are provided in Appendix G.

Stage II: Morphology-Conditioned Forecasting. The morphology description r provides a compact semantic prior that complements the numerical history $\mathbf{X}$. To leverage this prior without introducing excessive computational overhead, our numerical forecaster f_{forecast} is built upon Kairos (Feng et al., 2025), a lightweight TSFM designed for zero-shot forecasting across heterogeneous temporal patterns. We retain its encoder-decoder backbone and prediction head, but augment it with a text adapter and a gated cross-modal fusion that integrate the morphology description r into forecasting.

Semantic Prior Encoding. A text encoder (GIST (Solatorio, 2024)) followed by a projection layer (MLP) maps the morphology description into the hidden space of Kairos:

$$\boldsymbol{E}_r = \mathrm{Proj}\big(\mathrm{TextEnc}(r)\big) \in \mathbb{R}^{M \times D_h}, \quad (2)$$

where M is the token length of the encoded description and D_h is the hidden dimension of Kairos.

Gated Cross-Modal Fusion. We introduce a lightweight fusion operator that integrates the semantic prior into the decoder of Kairos. At each decoder layer l, the hidden state $\boldsymbol{H}^{(l)}$ is updated as:

$$\boldsymbol{H}^{(l)} \leftarrow \mathrm{LN}\Big(\boldsymbol{H}^{(l)} + \sigma\big(\boldsymbol{g}^{(l)}\big) \odot \mathrm{CA}\big(\boldsymbol{H}^{(l)}, \boldsymbol{E}_r\big)\Big), \quad (3)$$

where $\mathrm{LN}(\cdot)$ denotes layer normalization, $\mathrm{CA}(\cdot)$ denotes cross-attention with $\boldsymbol{H}^{(l)}$ as queries and $\boldsymbol{E}_r$ as keys/values, $\boldsymbol{g}^{(l)} \in \mathbb{R}^{D_h}$ is a learnable gate, σ is the sigmoid function, and $\odot$ denotes element-wise multiplication. The gate allows the model to adaptively control how much semantic information from the morphology prior is fused at each layer. We additionally decouple the decoder queries into horizon-specific readouts to support multiple prediction lengths in a single forward pass (details in Appendix A.2).

Discussion. Unlike tool-augmented agents (Wu et al., 2026; Tao et al., 2026) that either output nu-

merical forecasts directly or isolate reasoning from prediction, KAIROSAGENT fuses tool-grounded reasoning into the TSFM hidden space as a semantic prior. Furthermore, rather than relying on the static text encodings typical of existing multimodal models (Wu et al., 2025b; Liu et al., 2025a; Chowdhury et al., 2026), our framework dynamically constructs this prior through multi-turn tool interaction, specifically adapting to the statistical characteristics of each input series.

3.3 T-STAR Corpus

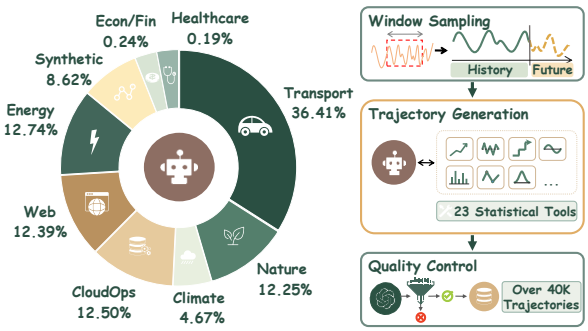

Figure 3: Overview of the T-STAR corpus, illustrating its domain distribution and generation pipeline.

We construct T-STAR, a tool-augmented corpus comprising over $40k$ validated examples drawn from 29 datasets across nine domains (Figure 3). Unlike conventional forecasting datasets that pair a history window with future targets, each T-STAR example records the full reasoning trajectory of a tool-calling agent, including multi-turn tool calls, tool responses, and a natural-language morphology forecast, thereby enabling optimization of both *tool-calling behavior* and *cross-modal fusion*, rather than final-answer imitation alone. The agent is equipped with 23 statistical tools spanning seven categories (Appendix B.2), and 30% of examples are generated with metadata masked to improve robustness. An automatic quality-control pipeline filters invalid trajectories (Appendix B.3).

3.4 Training Strategy

The training of KAIROSAGENT proceeds in three stages: (I) SFT warms up the reasoner with tool-calling capabilities on T-STAR trajectories; (II) multimodal alignment trains the TSFM to leverage morphology descriptions as semantic priors; and (III) RL with turn-level credit assignment further refines the agent beyond imitation.

Stage I: SFT of the Reasoner. We warm up the reasoner via SFT on T-STAR trajectories, formatting each as a multi-turn chat containing system instructions, user queries, tool interactions, and the morphology description. We optimize the reasoner using a standard next-token prediction objective over all assistant-side tokens, including intermediate tool calls. This stage equips the reasoner with a stable imitation policy for tool selection, argument generation, and morphology synthesis.

Stage II: Multimodal Alignment of the Forecaster. With the reasoner trained, we align the TSFM forecaster to leverage morphology descriptions as semantic priors. Each sample pairs the historical sequence and future target with the morphology description from its T-STAR trajectory. The forecaster and text encoder are both warm-started from their respective pretrained weights. During training, we fully fine-tune the forecaster backbone, text encoder, cross-modal fusion modules, and prediction head with the quantile pinball loss:

$$\ell_q(y, \hat{y}_q) = 2\,|(y - \hat{y}_q)\,(\mathbb{I}[y \le \hat{y}_q] - q)|\,, \quad (4)$$

where y is the target value and $\hat{y}_q$ is the predicted quantile at level q. The full forecasting objective $\mathcal{L}_{\mathrm{q}}(\mathbf{X}, r; \mathbf{Y})$ averages this loss over all quantile levels and forecast steps (detailed in Appendix A.3).

Stage III: RL with Fine-Grained Credit Assignment. While SFT establishes a stable imitation policy, it restricts the reasoner to the patterns within the demonstrated training trajectories and cannot directly optimize for the downstream forecasting tasks. To enhance reasoning quality, we refine the reasoner using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), employing the frozen Stage II forecaster as a reward module. This stage aligns the agentic tool-calling and reasoning with the objective of minimizing forecasting error.

Turn-Level Advantage. To provide a denser learning signal than a sparse outcome reward at the end of a reasoning trajectory, we evaluate the forecasting utility at each interaction turn. Given a trajectory with N turns, we first prompt the reasoner to generate an initial morphology description r^0 without tool invocation to serve as a baseline. For each subsequent turn i (where $i = 1, \ldots, N$), the reasoner generates a morphology description r^i based on the current reasoning and tool observations. Each description is scored by the frozen TSFM using the negative forecasting objective defined in Stage II: $S^i = -\mathcal{L}_{\mathrm{q}}(\mathbf{X}, r^i; \mathbf{Y})$. We de-

fine the turn-level reward as the marginal improvement $\Delta^i = S^i - S^{i-1}$, isolating the incremental contribution of turn i. The turn-level return is then computed as the discounted sum of future improvements $R^i = \sum_{j=i}^{N} \gamma^{j-i} \Delta^j$, where $\gamma \in [0, 1]$ is the discount factor. For each prompt, we sample a group of G trajectories and compute the group-normalized advantage $\hat{A}_g^i = (R_g^i - \mu(\{R\}))/\sigma(\{R\})$, where $\mu(\{R\})$ and $\sigma(\{R\})$ are the mean and standard deviation of returns across the group and $g \in [1, G]$.

Turn-Aware Policy Optimization Let s^i denote the set of tokens generated in turn i. We assign the turn-level advantage $\hat{A}^i$ to each token in s^i and optimize the policy π_{reason} via the following objective:

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}_{g,i}\Big[\min\big(\rho_g^t \hat{A}_g^i, \text{clip}(\rho_g^t, 1 \pm \epsilon)\, \hat{A}_g^i\big) - \beta\, D_{\text{KL}}\big(\pi_{\text{reason}} \| \pi_{\text{ref}}\big)\Big], \tag{5}$$

where $\rho_g^t = \frac{\pi_{\text{reason}}(s_g^t | s_g^{<t})}{\pi_{\text{old}}(s_g^t | s_g^{<t})}$ is the importance ratio, ϵ is the clipping threshold, π_{old} is the old policy, π_{ref} is the SFT-initialized reference policy, and β controls the KL regularization strength.

Discussion. Outcome-supervised RL (Shao et al., 2024) distributes a single end-of-trajectory reward uniformly, failing to distinguish informative tool calls from redundant ones in multi-turn settings. By employing a fine-grained, turn-level credit assignment, we effectively attribute forecasting improvements to specific actions. This approach overcomes the limitations of sparse outcome rewards by providing a targeted learning signal for every intermediate reasoning and tool-call step.

4 Experiment

In this section, we evaluate the performance of KAIROSAGENT following four key research questions: **RQ1:** Does the tool-calling mechanism of KAIROSAGENT facilitate the precise analysis of time series data, thereby enabling more accurate reasoning regarding future morphological information? (Section 4.2 and Section 4.4) **RQ2:** Can KAIROSAGENT achieve robust zero-shot forecasting capabilities in time series by inferring future morphological information and leveraging multimodal fusion? (Section 4.3) **RQ3:** Does turn-level fine-grained credit assignment improve reasoning quality over outcome-only RL? (Section 4.4) **RQ4:** Does multimodal fusion effectively contribute to zero-shot time series forecasting? (Section 4.4)

4.1 Experiment Settings

Benchmarks. We evaluate KAIROSAGENT on two established multimodal time series benchmarks. *Time-MMD* (Liu et al., 2024) is a large-scale benchmark spanning nine real-world domains, where each time series is paired with domain-relevant textual context. We adopt Time-MMD for both reasoning evaluation, where an LLM-as-a-judge protocol assesses the deduced future morphology (Section 4.2), and zero-shot forecasting evaluation (Section 4.3). *Time-IMM* (Chang et al., 2026) is a benchmark for irregular multimodal multivariate time series forecasting, comprising nine datasets that each capture a distinct source of real-world temporal irregularity (e.g., event-driven logging, adaptive sampling). We use Time-IMM exclusively for forecasting evaluation to assess model robustness under irregular-sampling conditions. All evaluation datasets are strictly held out from training. We further exclude the *Health* domain from Time-MMD and the *ILINet* dataset from Time-IMM, as both are highly homologous with the *CDC Fluview ILINet* data present in our `T-STAR` training corpus.

Baselines. For *reasoning evaluation*, we include GPT-5.2 (Singh et al., 2025) and DeepSeek-R1 (Guo et al., 2025) as advanced reference models, and evaluate against comparable-scale baselines including Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025). For *forecasting evaluation*, we compare against zero-shot TSFMs, full-shot multimodal models, and full-shot unimodal models (detailed in Appendix C.3).

Metrics. For reasoning evaluation, we report Accuracy as judged by the LLM-as-a-judge protocol. For forecasting evaluation, we adopt Mean Squared Error (MSE) and Mean Absolute Error (MAE) as primary metrics following standard practice (Liu et al., 2024; Chang et al., 2026).

4.2 Zero-Shot Reasoning Evaluation

Finding 1: KAIROSAGENT *(4B) surpasses all comparable-scale baselines on morphology reasoning accuracy, showing that tool-grounded reasoning is more effective than model scale alone.* Table 3 reports the accuracy of predicting future

Table 2: Performance comparison on Time-MMD across diverse domains, with complete results reported in Appendix F. Best results are highlighted in **red bold**, and second-best results are marked with blue underline.

Type	Zero-Shot Models										Full-Shot Multimodal Models						Full-Shot Unimodal Models			
Models	KairosAgent (Ours)		Aurora (2025b)		Sundial (2025c)		Moirai (2024)		ChronosBolt (2024)		T3Time (2026)		TimeCMA (2025a)		CALF (2025b)		PatchTST (2023)		DLinear (2023)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Agriculture	**0.194**	**0.282**	0.282	0.356	0.327	0.366	0.239	0.306	0.218	0.302	0.229	0.303	0.318	0.360	0.241	0.311	0.248	0.308	0.377	0.396
Climate	**0.863**	**0.739**	**0.863**	0.747	0.920	0.765	0.982	0.792	0.948	0.788	1.206	0.894	1.282	0.926	1.199	0.895	1.176	0.891	1.036	0.807
Economy	**0.186**	**0.335**	0.275	0.412	0.216	0.348	0.198	0.345	0.192	0.342	0.239	0.384	0.262	0.412	0.223	0.370	0.223	0.380	0.218	0.370
Energy	**0.217**	**0.330**	0.251	0.370	0.234	0.337	0.261	0.347	0.263	0.355	0.266	0.378	0.351	0.447	0.258	0.373	0.243	0.353	0.233	0.346
Environment	0.378	0.435	**0.276**	**0.379**	0.379	0.443	0.412	0.446	0.427	0.462	0.489	0.507	0.536	0.533	0.537	0.509	0.496	0.513	0.591	0.627
Security	76.658	4.340	72.763	4.085	83.403	4.836	74.249	4.129	73.977	4.117	72.113	4.070	**72.011**	4.113	73.267	**4.040**	76.105	4.445	82.521	4.891
Social Good	**0.769**	**0.376**	0.828	0.506	0.819	0.377	0.868	0.391	0.951	0.388	0.998	0.432	1.092	0.578	0.890	0.416	0.959	0.475	0.891	0.448
Traffic	**0.151**	**0.231**	0.162	0.289	0.228	0.292	0.186	0.263	0.222	0.249	0.289	0.368	0.297	0.412	0.227	0.305	0.209	0.316	0.219	0.315
1st Count	6	6	2	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

Table 3: Morphology reasoning accuracy (%) on Time-MMD. Best and second-best results among open-source models are in **red bold** and blue underline.

Model	Climate	Energy	Traffic
Advanced Models (reference only)			
GPT-5.2	97.80	84.12	34.95
DeepSeek-R1	99.18	79.51	37.76
Comparable-Scale Models			
Llama-3.1-8B-Instruct	52.47	38.72	43.62
DeepSeek-R1-Distill-Qwen-7B	42.86	5.08	14.29
KairosAgent-4B (SFT)	97.80	45.21	40.56
KairosAgent-4B (Outcome RL)	96.70	43.33	38.27
KairosAgent-4B (Turn-Level RL)	**98.08**	**50.47**	**43.88**

morphology (horizon 96) on three Time-MMD domains, judged by GPT-5.2. Among comparable-scale models, KairosAgent-4B with turn-level RL achieves the highest accuracy across all three domains, outperforming Llama-3.1-8B-Instruct and DeepSeek-R1-Distill-Qwen-7B by large margins despite being a smaller model. Notably, on Climate and Traffic, KairosAgent-4B even matches or exceeds the advanced reference models (GPT-5.2 and DeepSeek-R1), demonstrating that tool-augmented reasoning can effectively compensate for the gap in model scale.

4.3 Zero-Shot Forecasting Evaluation

Finding 2: KairosAgent *achieves the best zero-shot forecasting on both Time-MMD and Time-IMM, outperforming zero-shot TSFMs and even full-shot baselines with in-domain supervision.*

Regular Forecasting Evaluation. Table 2 reports the forecasting results on Time-MMD, featuring regularly sampled test series. KairosAgent achieves the lowest MSE and MAE in six of the eight domains, outperforming zero-shot TSFMs. Notably, KairosAgent performs competitively against fully supervised unimodal and multimodal baselines, despite lacking benchmark-specific training. This indicates that morphology priors inferred by the reasoner supply critical global information beyond mere local numerical fitting.

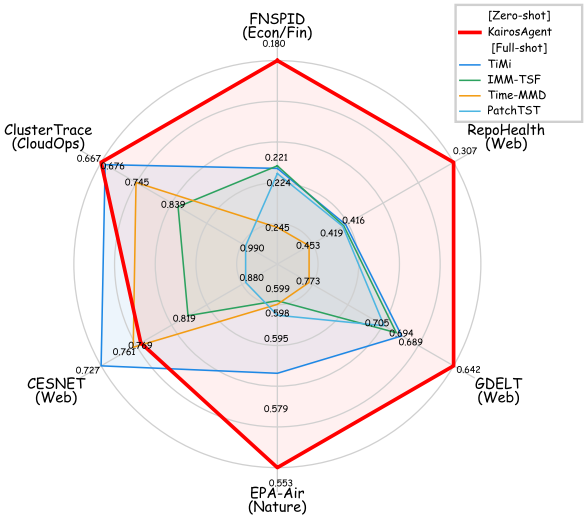

Figure 4: Performance comparison (MAE) on Time-IMM across irregular multimodal time series forecasting tasks, with complete results reported in Appendix F.

Irregular Forecasting Evaluation. Figure 4 evaluates robustness to temporal irregularities on Time-IMM. KairosAgent consistently achieves the lowest MAE in a zero-shot setting, outperforming fully supervised baselines. This confirms that tool-grounded morphology reasoning provides transferable priors for irregular time series forecasting.

4.4 Model Analysis

Finding 3: *Turn-level RL yields significantly better reasoning quality than SFT-only or outcome-only RL, removing the morphology prior leads to substantial forecasting degradation, and the agent learns a data-dependent tool selection policy rather than a fixed calling pattern.*

Ablation on Training Strategies. Table 3 also ablates the training strategies for the reasoner

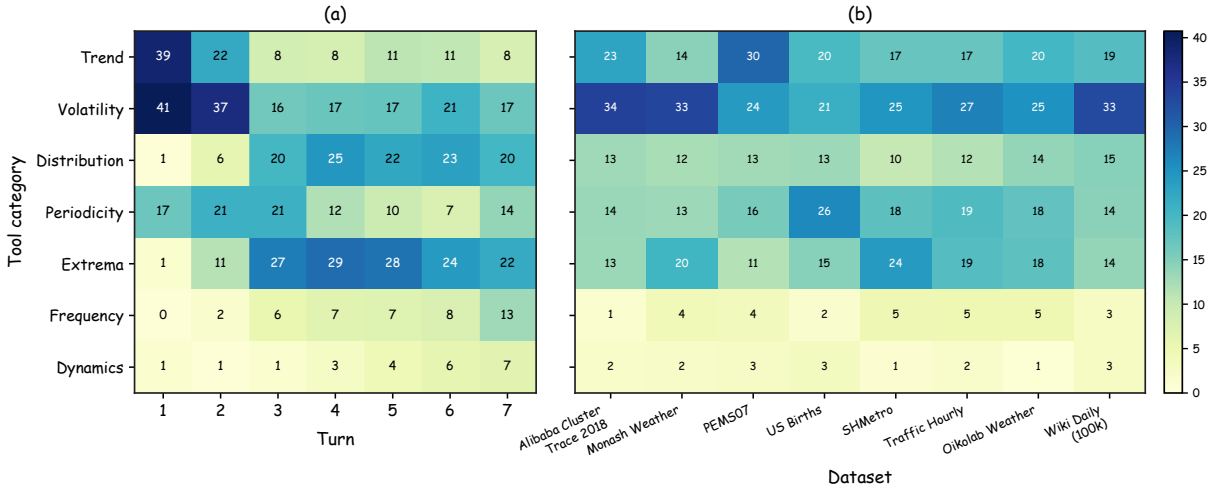

Figure 5: Tool usage in T-STAR reasoning trajectories. (a) Per-turn distribution: Trend and volatility dominate early turns, shifting toward extrema, distribution, and frequency later. (b) Per-dataset distribution: Tool selection adapts to dataset-specific temporal properties. Values denote column-normalized tool call percentages.

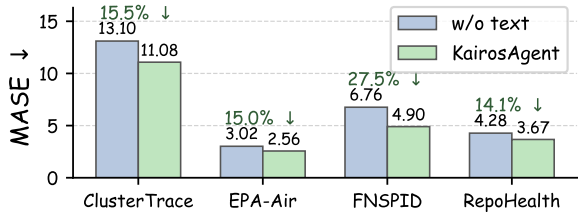

Figure 6: Morphology prior improves Time-IMM forecasting across four datasets (MASE ↓; lower is better). "w/o text" denotes the unimodal TSFM baseline.

across three variants: (1) SFT-only, (2) SFT with outcome-level RL (single trajectory-end reward), and (3) SFT with turn-level RL (our full method). Outcome-level RL slightly degrades performance compared to SFT alone, likely because the sparse reward signal cannot distinguish useful tool calls from redundant ones. Compared to SFT alone, outcome-level RL slightly degrades performance, with sparse terminal rewards struggling to assign accurate credit across long-horizon decision trajectories. In contrast, turn-level RL consistently improves over both baselines across all domains, with the largest gain on Traffic. This validates the necessity of dense, step-wise feedback for effectively optimizing multi-turn reasoning.

Ablation on Input Modalities. We compare KAIROSAGENT against the unimodal TSFM baseline Kairos whose pretrained weights are used to initialize our forecaster. As shown in Figure 6, incorporating the morphology prior consistently reduces MASE across four Time-IMM datasets. This confirms that the semantic prior provides information complementary to what the TSFM extracts from raw numerical history.

Tool Usage Analysis. Figure 5(a) shows that the agent adopts a staged inspection strategy. Initially, trend and volatility tools dominate the first two turns, indicating the agent establishes global evidence prior to specialized queries. Later turns shift toward extrema and distribution tools, refining the forecast via local turning points and value ranges. Meanwhile, frequency and dynamics tools are invoked selectively for periodic or non-stationary patterns. Figure 5(b) reveals dataset-specific adaptations: US Births (Healthcare) and SHMetro (Transport) favor periodicity tools, Alibaba (CloudOps) and Wiki Daily (Web) emphasize volatility, and PEMS07 (Transport) prioritizes trends. Ultimately, instead of a rigid template, KAIROSAGENT learns a dynamic diagnostic policy tailored to the statistical structure of each time series.

5 Conclusion

We present KAIROSAGENT, a framework bridging semantic reasoning and numerical forecasting in time series. By equipping an LLM-based reasoner with statistical analysis tools and fusing the deduced morphology prior into a TSFM latent space, KAIROSAGENT unifies interpretable reasoning with precise zero-shot forecasting. For training, we curated the T-STAR corpus of over $40k$ tool-augmented trajectories and introduced turn-level credit assignment to provide fine-grained optimization signals for intermediate steps. Experiments demonstrate that KAIROSAGENT achieves superior zero-shot performance alongside transparent reasoning traces, highlighting a promising paradigm for multimodal time series agent architectures.

Limitations

Our current implementation uses Kairos as the sole TSFM backbone. While the framework is architecture-agnostic and the gated cross-modal fusion module can be adapted to other TSFMs, we have not yet validated this generality empirically. Additionally, we focus exclusively on reasoning and forecasting tasks. The core idea of fusing semantic priors into a task-specific foundation model is applicable to other time series analysis tasks such as classification and anomaly detection, but exploring these extensions is left for future work.

Ethics Statement

This work introduces T-STAR, a tool-augmented corpus for time series reasoning, built from public or previously released time series datasets. We do not collect new human-subject data or private annotations. Metadata is used only when available from source datasets, and metadata-masked examples are included to reduce reliance on potentially sensitive context. We apply automatic quality checks and exclude benchmark subsets that overlap with training data to reduce leakage.

KAIROSAGENT is a research prototype, not a standalone tool for high-stakes decisions in domains such as healthcare, finance, or infrastructure. Forecasts and reasoning traces may be incorrect and should be used only with expert review, task-specific validation, privacy safeguards, and monitoring for misuse or distribution shift.

All source time-series datasets used to construct T-STAR are publicly available or previously released for research use. Because these datasets originate from heterogeneous repositories and providers, their redistribution and derivative-use terms may differ across sources. We follow the license and terms of use of each original dataset, preserve required attribution, and avoid redistributing raw third-party data when the source terms do not clearly permit it. Upon acceptance, we will release the T-STAR corpus, including generated tool-augmented reasoning trajectories, metadata, preprocessing scripts, and source pointers, subject to the corresponding upstream terms. We will release the KAIROSAGENT codebase under the Apache-2.0 license to facilitate reproducible research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, and 1 others. 2020. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, and 1 others. 2024. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. 2026. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *Advances in Neural Information Processing Systems*, 38:57529–57580.

Adrián Bazaga, Rexhina Blloshmi, Bill Byrne, and Adrià de Gispert. 2025. Learning to reason over time: Timeline self-reflection for improved temporal reasoning in language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28014–28033.

Defu Cao, Michael Gee, Jinbo Liu, Hengxuan Wang, Wei Yang, Rui Wang, and Yan Liu. 2025. Conversational time series foundation models: Towards explainable and effective forecasting. *arXiv preprint arXiv:2512.16022*.

Ching Chang, Jeehyun Hwang, Yidan Shi, Haixin Wang, Wei Wang, Wen-Chih Peng, and Tien-Fu Chen. 2026. Time-imm: A dataset and benchmark for irregular multimodal multivariate time series. *Advances in Neural Information Processing Systems*, 38.

Abdul Monaf Chowdhury, Rabeya Akter, and Safaeid Hossain Arib. 2026. T3time: Tri-modal time series forecasting via adaptive multi-head alignment and residual fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20597–20605.

Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. 2018. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307:72–77.

Zhiqing Cui, Binwu Wang, Qingxiang Liu, Yeqiang Wang, Zhengyang Zhou, Yuxuan Liang, and Yang Wang. 2025. Augur: Modeling covariate causal associations in time series via large language models. *arXiv preprint arXiv:2510.07858*.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Lintao Ma, Xingyu Lu, and Kan Ren. 2025. Kairos: Towards adaptive and generalizable time series foundation models. *arXiv preprint arXiv:2509.25826*.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. 2025. Timeomni-1: Incentivizing complex reasoning with time series in large language models. *arXiv preprint arXiv:2509.24803*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Lu Han, Yu Liu, Qiwen Deng, Jian Jiang, Yinbo Sun, Zhe Yu, Binfeng Wang, Xingyu Lu, Lintao Ma, Han-Jia Ye, and 1 others. 2025. Unica: Adapting time series foundation model to general covariate-aware forecasting. *arXiv preprint arXiv:2506.22039*.

Gunjan Jalori, Preetika Verma, and Sercan Ö Arık. 2025. Flairr-ts–forecasting llm-agents with iterative refinement and retrieval for time series. *arXiv preprint arXiv:2508.19279*.

Jiawei Jiang, Chengkai Han, Wenjun Jiang, Wayne Xin Zhao, and Jingyuan Wang. 2023. Libcity: A unified library towards efficient and comprehensive urban spatial-temporal prediction. *arXiv preprint arXiv:2304.14343*.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. 2025. 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)*, pages 29736–29753.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104.

Jiafeng Lin, Yuxuan Wang, Huakun Luo, Zhongyi Pei, and Jianmin Wang. 2026. Timi: Empower time series transformers with multimodal mixture of experts. *arXiv preprint arXiv:2602.21693*.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. 2025a. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18780–18788.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and 1 others. 2024. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. 2025b. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18915–18923.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2025c. Sundial: A family of highly capable time series foundation models. In *International Conference on Machine Learning*, pages 39295–39317. PMLR.

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

Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. 2020. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Aivin V Solatorio. 2024. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning. *arXiv preprint arXiv:2402.16829*.

Xiaoyu Tao, Mingyue Cheng, Chuang Jiang, Tian Gao, Huanjian Zhang, and Yaguo Liu. 2026. Cast-r1: Learning tool-augmented sequential decision policies for time series forecasting. *arXiv preprint arXiv:2602.13802*.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, and 1 others. 2026. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

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

Zhixian Wang, Qingsong Wen, Chaoli Zhang, Liang Sun, Leandro Von Krannichfeldt, Shirui Pan, and Yi Wang. 2023. Benchmarks and custom package for energy forecasting. *arXiv preprint arXiv:2307.07191*.

Gerald Woo, Chenghao Liu, Akshat Kumar, and Doyen Sahoo. 2023. Pushing the limits of pre-training for time series forecasting in the cloudops domain. *arXiv preprint arXiv:2310.05063*.

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

Wen Wu, Ziyang Zhang, Liwei Liu, Xuenan Xu, Jimin Zhuang, Ke Fan, Qitan Lv, Junlin Liu, Chen Zhang, Zheqi Yuan, and 1 others. 2025a. Scits: Scientific time series understanding and generation with llms. *arXiv preprint arXiv:2510.03255*.

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

Xingjian Wu, Junkai Lu, Zhengyu Li, Xiangfei Qiu, Jilin Hu, Chenjuan Guo, Christian S Jensen, and Bin Yang. 2026. Timeart: Towards agentic time series reasoning via tool-augmentation. *arXiv preprint arXiv:2601.13653*.

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

Wen Ye, Wei Yang, Defu Cao, Yizhou Zhang, Lumingyuan Tang, Jie Cai, and Yan Liu. 2024. Domain-oriented time series inference agents for reasoning and automated analysis. *arXiv preprint arXiv:2410.04047*.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2267–2276.

A Implementation Details

A.1 Model Configurations

A.1.1 Stage I: SFT Configuration

We fine-tune Qwen3.5-4B with full-parameter SFT on 80% of the T-STAR trajectories, with the remaining 20% reserved for Stage III RL training. Training uses 32 NVIDIA A100 GPUs with DeepSpeed ZeRO-3 and FlashAttention-2, completing in approximately 1 day. The maximum sequence length is set to 30k tokens to accommodate multi-turn tool-calling trajectories. Table 4 lists the full hyperparameters.

Table 4: Hyperparameters for Stage I SFT.

Hyperparameter	Value
Base model	Qwen3.5-4B
Fine-tuning type	Full-parameter SFT
Maximum length	30k tokens
Global batch size	256
Training epochs	3
Learning rate	1×10^{-5}
LR scheduler	Cosine
Warmup ratio	0.1
Precision	bf16
Hardware	32× A100 80GB
Training time	∼1 day

A.1.2 Stage II: Multimodal Alignment Configuration

We align the TSFM to leverage morphology descriptions as semantic priors, using the full T-STAR corpus for multimodal alignment. Following UniCA (Han et al., 2025), we adopt GIST-small-Embedding-v0 (Solatorio, 2024) (containing 33.4M parameters) as the text encoder, which provides compact yet informative sentence embeddings suitable for conditioning time series models. Including the text encoder, cross-modal fusion modules, and horizon-specific prediction heads, the modified Kairos-base forecaster comprises a total of 109.1M parameters. The learnable gate $\boldsymbol{g}^{(l)}$ in the cross-modal fusion module is initialized to -2.197, such that $\sigma(-2.197) \approx 0.1$. This ensures that at the beginning of training, only a small amount of semantic information is fused into the decoder, preserving the pretrained forecasting capability of Kairos while allowing the fusion path to be gradually activated during optimization. Training uses single NVIDIA A100 GPUs and completes in approximately 40 minutes. Table 5 lists the full hyperparameters.

Table 5: Hyperparameters for Stage II multimodal alignment.

Hyperparameter	Value
TSFM backbone	The modified Kairos-base
Initialization	Pretrained Kairos
Trainable modules	All
Text encoder	GIST-small-Embedding-v0
Maximum text length	512 tokens
Gate initialization	-2.197 ($\sigma \approx 0.1$)
Global batch size	128
Training steps	10k
Learning rate	1×10^{-4}
Precision	tf32
Hardware	1× A100 80GB
Training time	∼40 min

A.1.3 Stage III: RL Configuration

We refine the reasoner with GRPO on the 20% held-out T-STAR trajectories, using the frozen Stage II forecaster as a reward model. The turn-level credit assignment mechanism provides fine-grained optimization signals for each tool call and reasoning step. Table 6 lists the configuration.

Table 6: Configuration for Stage III RL.

Hyperparameter	Value
Policy initialization	Stage I SFT reasoner
Reward model	Stage II forecaster (frozen)
RL algorithm	GRPO
Group size G	8
Clipping ϵ	0.2
KL coefficient β	0
Discount factor γ	1.0
Learning rate	1×10^{-6}
Training steps	4k
Max turns per trajectory N	8
Precision	bf16
Hardware	8× H800 80GB

A.2 Horizon-Decoupled Forecasting

To support practical deployment across multiple prediction lengths simultaneously, we extend the single-horizon formulation to K target horizons $\{H_1, \ldots, H_K\}$. The Multi-Patch Decoder of Kairos uses a set of J learnable patch queries to attend to the encoded history $\boldsymbol{H}^{\text{en}}$ and predict J future patches in parallel. In KAIROSAGENT, we replace these patch-indexed queries with K horizon-specific queries $\{\mathbf{q}_k\}_{k=1}^{K}$, where each $\mathbf{q}_k$ is a learnable embedding dedicated to horizon H_k. The decoder produces K horizon-specific output

embeddings:

$$\boldsymbol{h}_1^{\mathrm{de}},\ \ldots,\ \boldsymbol{h}_K^{\mathrm{de}}\ =\ \mathrm{Dec}\big([\mathbf{q}_1;\ \ldots;\ \mathbf{q}_K],\ \boldsymbol{H}^{\mathrm{en}},\ \boldsymbol{E}_r\big). \tag{6}$$

All queries share the same encoded history and semantic prior, yet each specialises toward a distinct temporal scope through learned attention patterns. As the gated cross-modal fusion is applied at each decoder layer, $\boldsymbol{h}_k^{\mathrm{de}}$ is already conditioned on the morphology prior. The final predictions are produced by horizon-specific prediction heads:

$$\hat{\mathbf{Y}}_k = \mathrm{Head}_k\big(\boldsymbol{h}_k^{\mathrm{de}}\big), \qquad k = 1, \ldots, K, \tag{7}$$

where $\hat{\mathbf{Y}}_k \in \mathbb{R}^{H_k}$ corresponds to the prediction for the k-th target horizon H_k. By assigning each horizon a dedicated query and prediction head, the model produces all horizons in a single forward pass, avoids the cumulative error inherent in autoregressive multi-step rollouts, and allows each head to specialise its capacity for a particular temporal range while sharing the morphological guidance from $\boldsymbol{E}_r$.

A.3 Loss Function

This section details the full quantile regression objective whose pinball loss is defined in Eq. (4). For a training instance with history $\mathbf{X}$, morphology description r, and future target $\mathbf{Y} = (y_1, \ldots, y_H)$, the forecaster first normalizes the target with instance statistics estimated from the history window. We denote the normalized target value by $\tilde{y}_t$, and the normalized prediction produced by the k-th horizon head for quantile level $q \in \mathcal{Q}$ at step t by $\hat{y}_{k,q,t}$. Invalid or padded future positions are excluded from the loss.

For the k-th prediction head with horizon H_k, the masked horizon loss is

$$\mathcal{L}_k = \frac{1}{B}\sum_{b=1}^{B}\sum_{t=1}^{H_k}\omega_t\,\frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \ell_q(\tilde{y}_{b,t}, \hat{y}_{b,k,q,t}), \tag{8}$$

where B is the batch size and ℓ_q is the pinball loss (Eq. (4)) applied to normalized values. We use log-decay temporal weighting:

$$\omega_t = \frac{1}{H_k}\left(\log H_k - \log t\right), \tag{9}$$

which emphasizes near-term forecast steps while preserving supervision over the full horizon. The final forecasting objective averages over all K horizon-specific heads:

$$\mathcal{L}_{\mathrm{q}}(\mathbf{X}, r; \mathbf{Y}) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_k. \tag{10}$$

B Corpus Curation Details

This appendix provides the full details of the T-STAR corpus construction, including the training dataset composition, the tool suite, the generation pipeline with quality control, and the prompt templates.

B.1 Training Dataset Details

The T-STAR corpus draws from 29 datasets spanning nine domains: Transport, Energy, Climate, CloudOps, Nature, Web, Healthcare, Econ/Fin, and Synthetic. Table 7 provides an overview with the number of validated trajectories retained after quality control. Below we briefly describe each dataset.

Table 7: Training datasets in T-STAR. "# Samples" denotes the number of validated trajectories after quality control.

Dataset	Domain	Frequency	# Samples
Los-Loop	Transport	5T	2,679
PEMS03	Transport	5T	139
PEMS04	Transport	5T	209
PEMS07	Transport	5T	134
PEMS08	Transport	5T	169
PEMS-BAY	Transport	5T	311
SHMetro	Transport	15T	2,696
Taxi (30 Min.)	Transport	30T	1,879
Mexico City Bikes	Transport	H	3,894
Traffic Hourly	Transport	H	3,254
Covid19 Energy	Energy	H	283
Elecdemand	Energy	30T	131
ELF	Energy	H	189
ERCOT Load	Energy	H	869
GEF12	Energy	H	520
Electricity Hourly	Energy	H	3,366
Oikolab Weather	Climate	H	383
Monash Weather	Climate	D	1,586
Alibaba Cluster Trace 2018	CloudOps	5T	2,311
Azure VM Traces 2017	CloudOps	5T	2,962
WeatherBench	Nature	D	2,834
China Air Quality	Nature	H	1,928
Sunspot	Nature	D	92
Extended Web Traffic	Web	D	2,994
Wiki Daily (100k)	Web	D	2,235
CDC FluView ILINet	Healthcare	W	45
US Births	Healthcare	D	35
Exchange Rate	Econ/Fin	D	101
Synthetic	Synthetic	–	3,636
Total			**41,864**

B.1.1 Transport

Los-Loop (Jiang et al., 2023). 5-minute traffic speed time series from 207 loop detectors on Los

Angeles County highways, derived from Los Angeles loop detector data.

PEMS03/04/07/08 (Jiang et al., 2023). 5-minute traffic flow and multivariate time series for road sensors from the California PeMS benchmark. PEMS03 contains 358 sensors (2018-09-01 to 2018-11-30), PEMS04 contains 307 sensors with flow/occupancy/speed (2018-01-01 to 2018-02-28), PEMS07 contains 883 sensors (2017-05-01 to 2017-08-06), and PEMS08 contains 170 sensors with flow/occupancy/speed (2016-07-01 to 2016-08-31).

PEMS-BAY (Jiang et al., 2023). 5-minute traffic speed time series for 325 road sensors in the California Bay Area, spanning 2017-01-01 to 2017-06-30.

SHMetro (Jiang et al., 2023). 15-minute metro passenger flow time series for 288 stations in the Shanghai metro system from 2016-07-01 to 2016-09-30, with inbound and outbound dimensions.

Taxi (30 Min.) (Alexandrov et al., 2020). 30-minute taxi ride count time series for New York City locations, sampled from January 2015 and January 2016.

Mexico City Bikes (Ansari et al., 2024). Hourly bike-sharing trip count time series for Mexico City ECOBICI stations, spanning 2011-02-01 to 2022-10-01.

Traffic Hourly (Godahewa et al., 2021). Hourly road occupancy rate time series for 862 traffic sensors from the Monash Forecasting Repository, spanning 2015-01-01 to 2016-12-31.

B.1.2 Energy

Covid19 Energy (Wang et al., 2023). A single hourly aggregated-level electricity demand time series for a metropolitan electric utility area from 2017-03-18 to 2020-11-06, covering the COVID-19 period.

Elecdemand (Godahewa et al., 2021). A single half-hourly operational electricity demand time series for Victoria, Australia in 2014.

ELF (Wang et al., 2023). A single hourly electricity demand time series for Panama from 2018-01-01 to 2020-06-27.

ERCOT Load (Ansari et al., 2024). Eight time series representing the hourly energy load in eight weather zones within the Texas ERCOT grid from 2004 to 2021.

GEF12 (Wang et al., 2023). Twenty time series representing the hourly electricity load for a US utility from 2004-01-01 to 2008-06-30.

Electricity Hourly (Godahewa et al., 2021). Hourly electricity load time series for 321 Portuguese clients from the Monash Forecasting Repository, spanning 2012-01-01 to 2015-01-01.

B.1.3 Climate

Oikolab Weather (Woo et al., 2024). Hourly climate data from eight variables nearby Monash University, Clayton, Victoria, Australia, spanning 2010-01-01 to 2021-05-31.

Monash Weather (Godahewa et al., 2021). Daily weather time series measured at weather stations in Australia, including rain, minimum temperature, maximum temperature, and solar radiation.

B.1.4 CloudOps

Alibaba Cluster Trace 2018 (Woo et al., 2023). Multivariate 5-minute time series from Alibaba's 2018 production cluster trace, covering container-level CPU and memory utilization for approximately 4,000 machines over 8 days.

Azure VM Traces 2017 (Woo et al., 2023). 5-minute CPU utilization time series for Azure virtual machines from the 2017 Azure VM traces in one geographic region. Per-sample start times range from 2016-11-15 to 2016-12-13.

B.1.5 Nature

WeatherBench (Rasp et al., 2020). Daily time series derived from ERA5 reanalysis on a global grid from 1979-01-01 to 2018-12-31. Each univariate series corresponds to one meteorological variable at one latitude/longitude point.

China Air Quality (Zheng et al., 2015). Hourly air pollution time series for monitoring stations in China, spanning approximately 2014-05-01 to 2015-04-30.

Sunspot (Godahewa et al., 2021). Daily total sunspot number with missing observations retained, beginning on 1818-01-08.

B.1.6 Web

Extended Web Traffic (Godahewa et al., 2021). Daily web traffic time series for many Wikipedia pages, preserving missing values and spanning 2015-07-01 to 2022-06-30.

Wiki Daily (100k) (Ansari et al., 2024). Daily Wikipedia page view count time series for 100,000 pages from 2015-07-01 to 2022-12-31.

B.1.7 Healthcare

CDC FluView ILINet (Ansari et al., 2024). Weekly influenza-like illness surveillance time series from the CDC FluView system for national, HHS regional, census regional, and state-level reporting areas, spanning 1997-10-12 to 2023-10-22.

US Births (Godahewa et al., 2021). Daily birth counts in the United States, starting at 1969-01-01 and spanning 7,275 observations.

B.1.8 Econ/Fin

Exchange Rate (Lai et al., 2018). Eight daily foreign exchange rate time series relative to the US dollar, covering 1990-01-01 to 2019-01-30.

B.1.9 Synthetic

Synthetic. 10,000 procedurally generated synthetic time series following the synthesis procedure of Kairos (Feng et al., 2025), covering diverse morphological patterns. No metadata is provided (always treated as unavailable).

B.2 Details of Tools

The agent is equipped with 23 statistical analysis tools adapted from `tsfresh` (Christ et al., 2018). All tools operate on a user-specified window $[\texttt{left}, \texttt{right})$ of the history, allowing the agent to inspect both global and local temporal structures. Table 8 lists each tool grouped by category.

B.3 Pipeline

The corpus generation pipeline consists of three stages: sample windowing, trajectory generation, and quality control.

Sample Windowing. We scan the raw datasets to collect all logical sequences and generate candidate windows. Each window specifies a history segment of length L=2,048, a short-term future of H_s=96, and a long-term future of H_l=720. The detailed windowing procedure is presented in Algorithm 1. To improve robustness to incomplete real-world metadata, 30% of samples per dataset are generated with all metadata fields replaced by `unavailable`, forcing the model to learn time-series-only forecasting without relying on textual context.

Algorithm 1 Sample Windowing Strategy

Require: Dataset sequences $\{S_1, \ldots, S_N\}$; history length $L = 2048$; horizons $H_s = 96$, $H_l = 720$; default stride $d = 512$; budget cap $B_{\max} = 5000$; minimum budget $B_{\min} = 500$; fallback strides $\mathcal{D} = \{256, 128, 64\}$
Ensure: Set of candidate windows $\mathcal{W}$

$\mathcal{W} \leftarrow \emptyset$
for each sequence S_i with length n_i **do**
  if $n_i < H_s + H_l$ **then**
    skip S_i {Too short}
  else if $n_i < L + H_l$ **then**
    Extract one window using maximum available history {Max-length sample}
  else
    Generate sliding windows with stride d; let s_i = count
  end if
end for
$n_{\text{total}} \leftarrow \sum_i s_i$
if $n_{\text{total}} > B_{\max}$ **then**
  Allocate per-series budget $b_i \propto \sqrt{s_i}$ s.t. $\sum_i b_i = B_{\max}$
  Uniformly subsample b_i windows from each series S_i
else if $n_{\text{total}} < B_{\min}$ **then**
  for $d' \in \mathcal{D}$ **do**
    Re-generate windows with stride d'
    if count $\geq B_{\min}$ **then**
      break
    end if
  end for
end if
return $\mathcal{W}$

Trajectory Generation. For each manifest entry, the history values and metadata (when available) are assembled into a prompt and submitted to Kimi-K2.5 (Team et al., 2026), which serves as the tool-calling agent. The agent interacts with the tool suite in a multi-turn loop (up to 8 turns, 3 tool calls per turn), after which it produces a two-paragraph morphology forecast covering the short-term and long-term horizons.

Quality Control. Each generated trajectory undergoes three automatic quality checks executed sequentially by GPT-5.2 (Singh et al., 2025):

Table 8: The 23 time series analysis tools available to KAIROSAGENT, grouped by category.

Category	Tool	Description
Trend	`linear_trend`	Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one. This feature assumes the signal to be uniformly sampled. It will not use the time stamps to fit the model. This tool automatically returns the five attributes "pvalue", "rvalue", "intercept", "slope", and "stderr" from scipy.stats.linregress.
Volatility	`standard_deviation`	Returns the standard deviation of the selected time series window.
	`mean_abs_change`	Average over first differences. Returns the mean over the absolute differences between subsequent time series values: $\frac{1}{n-1}\sum_{i=1}^{n-1}\lvert x_{i+1}-x_i\rvert$.
	`absolute_sum_of_changes`	Returns the sum over the absolute value of consecutive changes: $\sum_{i=1}^{n-1}\lvert x_{i+1}-x_i\rvert$.
	`ratio_beyond_r_sigma`	Ratio of values in the selected time series window that are more than $r\cdot\sigma$ away from the mean, where σ is the standard deviation.
Distribution	`quantile`	Calculates the q quantile of the selected time series window. This is the value greater than $q\%$ of the ordered values.
	`change_quantiles`	First fixes a corridor given by the quantiles q_l and q_h of the distribution of the selected time series window. Then calculates the average consecutive change inside this corridor.
Periodicity	`autocorrelation`	Calculates the autocorrelation of the specified lag for the selected time series window, according to the formula $\frac{1}{(n-l)\sigma^2}\sum_{t=1}^{n-l}(X_t-\mu)(X_{t+l}-\mu)$ where n is the window length, σ^2 its variance, μ its mean, and l denotes the lag.
	`agg_autocorrelation`	Descriptive statistics on the autocorrelation of the time series. Calculates the value of an aggregation function f_{agg} (e.g. the variance or the mean) over the autocorrelation $R(l)$ for different lags. Returns $f_{\text{agg}}(R(1),\ldots,R(m))$ for $m=\max(n,\texttt{maxlag})$.
Extrema	`number_peaks`	Calculates the number of peaks of at least support n in the selected time series window. A peak of support n is defined as a value bigger than its n neighbours to the left and to the right.
	`number_cwt_peaks`	Number of different peaks in the selected time series window. The series is smoothed by a Ricker wavelet for widths ranging from 1 to n. Returns the number of peaks that occur at enough width scales and with sufficiently high SNR.
	`first_location_of_minimum`	Returns the first location of the minimal value of the selected time series window. The position is calculated relatively to the length of the window.
	`last_location_of_minimum`	Returns the last location of the minimal value of the selected time series window. The position is calculated relatively to the length of the window.
	`first_location_of_maximum`	Returns the first location of the maximum value of the selected time series window. The position is calculated relatively to the length of the window.
	`last_location_of_maximum`	Returns the relative last location of the maximum value of the selected time series window.
	`longest_strike_below_mean`	Returns the length of the longest consecutive subsequence in the selected time series window that is smaller than the mean.
	`longest_strike_above_mean`	Returns the length of the longest consecutive subsequence in the selected time series window that is bigger than the mean.
	`mean_n_absolute_max`	Calculates the arithmetic mean of the n absolute maximum values of the time series.
Frequency	`fft_coefficient`	Calculates the Fourier coefficients of the one-dimensional discrete Fourier Transform for real input by FFT. Can return the real part, imaginary part, absolute value, or angle in degrees.
	`spkt_welch_density`	Estimates the cross power spectral density of the time series at different frequencies. The time series is first shifted from the time domain to the frequency domain. Returns the power spectrum of the different frequencies.
	`fourier_entropy`	Calculate the binned entropy of the power spectral density of the time series (using the Welch method).
Dynamics	`augmented_dickey_fuller`	The Augmented Dickey-Fuller test is a hypothesis test which checks whether a unit root is present in a time series sample. Returns the value of the respective test statistic.
	`cid_ce`	An estimate for time series complexity. Calculates $\sqrt{\sum_{i=1}^{n-1}(x_i-x_{i-1})^2}$. A more complex time series has more peaks, valleys, etc.

1. **Metadata Leak Check.** A judge verifies that the final forecast does not contain exact numbers, timestamps, dataset names, variable names, units, or domain labels. Violating samples are regenerated with a stricter prompt constraint.

2. **Reasoning Usage Check.** A judge assesses whether the reasoning trace meaningfully incorporates tool outputs and metadata context, rather than producing a generic answer.

3. **Forecast Accuracy Check.** A judge evaluates whether the predicted morphology is broadly consistent with the held-out future values in terms of trend, periodicity, volatility, regime changes, and turning points.

Samples failing any check are retried up to 3 times. Samples that remain invalid after all retries are discarded. This pipeline ensures that retained trajectories exhibit correct tool usage, grounded reasoning, and morphologically accurate forecasts.

B.4 Prompt Templates

We provide the key prompt templates used during corpus generation.

System Prompt. The system prompt instructs the agent to act as an expert time-series forecasting analyst with domain knowledge. It specifies: (i) a *tool-usage policy* requiring at least one broad-window inspection and one targeted local window near the end of the history before forecasting; (ii) *metadata handling rules* that treat `unavailable` fields as absent with zero semantic weight; (iii) *interpretation principles* grounding claims on visible morphological evidence (trend, periodicity, cycle shape, phases, events, extreme placement, transition sharpness, intermittency, roughness, volatility, and amplitude change); and (iv) *output format requirements* (exactly two paragraphs beginning with "In the short term," and "In the long term," under 300 words). The prompt also communicates the turn budget (`max_assistant_turns`) and parallel tool-call limit (`max_parallel_calls`) so the agent can plan its tool usage accordingly.

User Prompt. Two user prompt variants are used depending on metadata availability. When metadata is available, the prompt provides the dataset name, domain, frequency, dataset description, variable name, variable description, unit, timestamp ranges for history and both future windows, and

Prompt: System

You are an expert time-series forecasting analyst with strong domain knowledge. When metadata is available, use it cautiously. When metadata is unavailable, operate as a domain-agnostic morphology forecaster.

You have access to external tools for local window analysis. Use tools first to inspect shape details, then produce one final paragraph.

Tool-usage policy: Prefer evidence from tools over unaided guessing. For every tool call, always include `left` and `right` (`right` is exclusive). Before forecasting, inspect at least one broad window that covers most or all of the history. Also inspect at least one targeted local window near the end of the history, because the short-term forecast should be grounded in the most recent regime. If needed, inspect additional windows around suspected turning points, repeating segments, or regime boundaries. Use additional tool calls only when they materially reduce uncertainty. After enough evidence is collected, stop calling tools and write the final answer. Do not mention tools, tool names, or tool outputs in the final answer.

Metadata rules: Some samples may contain missing or intentionally masked metadata. The token `unavailable` means the information is absent and carries zero semantic weight. Treat `unavailable` as missing information, not as a weak hint. Do not infer domain, units, variable identity, calendar semantics, or real-world causes from fields marked as `unavailable`. If frequency or timestamps are `unavailable`, reason only in relative positions such as early, middle, late, recent, and broader history. If metadata is unavailable, rely only on the numeric history and tool evidence. Do not mention missing or unavailable metadata in the final answer.

Interpretation principles: Use history only; do not assume unsupported external events. When reliable metadata is available, use domain knowledge only to interpret plausible persistence, recurrence, seasonality, or regime evolution that is consistent with the observed history. When metadata is unavailable or masked, do not guess hidden semantics; rely on morphology and tool evidence only. Base claims on visible evidence such as trend, periodicity, cycle shape, regimes, events, extreme placement, transition sharpness, intermittency, roughness, volatility, and amplitude change. If evidence is weak or conflicting, hedge explicitly instead of overclaiming.

Output requirements: Final answer must contain exactly two paragraphs. The first paragraph must begin with "In the short term," The second paragraph must begin with "In the long term," Each paragraph should describe predicted morphology only. Do not output chain-of-thought, hidden reasoning, or process narration. Avoid exact numbers, timestamps, dataset names, units, domain knowledge and causal stories. Be specific but conservative. Use 3–5 sentences per paragraph. Keep the full answer under 300 words.

Use tools when needed before writing the final answer. You can output at most `{max_assistant_turns}` assistant replies in total, and in each reply at most `{max_parallel_calls}` tool calls will be executed.

the comma-separated history values. When metadata is masked, all metadata fields are replaced with `unavailable`, and only the history length, forecast horizons, and raw history values are provided.

Prompt: User (Metadata Available)

Dataset: {dataset}
Domain: {domain}
Frequency: {freq}
Dataset description: {dataset_description}

Variable name: {var_name}
Variable description: {var_desc}
Unit: {unit}
Avoid **exact numbers, timestamps, dataset names, units, domain knowledge and causal stories** in final morphology output, only use them in your reasoning.

History window:
- start_time: {ht0}
- end_time: {ht1}
- length: {history_length}

Future window (short term):
- start_time: {ft0_s}
- end_time: {ft1_s}
- horizon: {horizon_short}

Future window (long term):
- start_time: {ft0_l}
- end_time: {ft1_l}
- horizon: {horizon_long}

History values (comma-separated, earliest to latest):
{history_values_text}

Prompt: User (Metadata Unavailable)

Metadata availability: unavailable

Avoid **exact numbers, timestamps** in final morphology output.

History window:
- start_time: unavailable
- end_time: unavailable
- length: {history_length}

Future window (short term):
- start_time: unavailable
- end_time: unavailable
- horizon: {horizon_short}

Future window (long term):
- start_time: unavailable
- end_time: unavailable
- horizon: {horizon_long}

History values (comma-separated, earliest to latest):
{history_values_text}

Judge Prompts. Each quality-control judge receives a structured prompt and returns a JSON verdict with a binary pass/fail decision and brief evidence (up to 60 words).

Prompt: Metadata Leak Judge

You are a strict judge for time-series morphology forecasts.

Determine whether the forecast text contains forbidden metadata-like content. Forbidden content includes exact numbers, timestamps, dataset names, variable names, units, domain labels.

Return JSON only:
{"pass": true/false, "evidence": "<=60 words"}

Forecast text: {forecast_text}
Reference metadata / context: {metadata_context}

Prompt: Reasoning Usage Judge

You are a strict judge for tool-using time-series reasoning.

Determine whether the model's reasoning meaningfully uses the provided metadata context and tool outputs, instead of ignoring them and producing a generic answer. If metadata is unavailable, judge whether the reasoning meaningfully uses the numeric history and tool outputs.

Return JSON only:
{"pass": true/false, "evidence": "<=60 words"}

Metadata context: {metadata_context}
Serialized conversation: {messages_text}

C Evaluation Details

C.1 Benchmarks

Time-MMD. Time-MMD (Liu et al., 2024) is a multi-domain multimodal time series benchmark designed for forecasting with aligned numerical and textual series. It covers nine primary domains: agriculture, climate, economy, energy, environment, health, security, social good, and traffic. Each domain provides a target numerical series together with temporally associated text collected from reports and web search results. Following the official setting as specified by the benchmark, datasets are split chronologically into 70% training, 10% validation, and 20% testing. Trainable full-shot baselines are fit on the training

Prompt: Forecast Accuracy Judge

You are a strict judge for time-series morphology forecast accuracy.

Determine whether the forecast text is broadly consistent with the actual future morphology shown in the short-term and long-term future values. Focus on morphology only: trend, periodicity, roughness, volatility, regime change, turning points, and overall structure.

Return JSON only:
`{"pass": true/false, "evidence": "<=60 words"}`

Forecast text: `{forecast_text}`
Short-term future values: `{future_short_text}`
Long-term future values: `{future_long_text}`

split and selected on the validation split, whereas KAIROSAGENT is evaluated zero-shot on the test split without using benchmark training labels. Details on how each model consumes textual context are provided in Appendix C.3.1. The official forecasting horizons are frequency-dependent, as summarized in Table 9. We do not evaluate the Health domain as it is highly homologous with the CDC FluView ILINet data included in the T-STAR training corpus; excluding it avoids train-test contamination and gives a cleaner zero-shot evaluation. *We also note that the Economy domain in Time-MMD has its original temporal order reversed (most recent observations first). We correct this ordering before evaluation; methods that do not account for this may report misleadingly low error on this domain.*

Table 9: Official Time-MMD datasets and forecasting horizons. †Excluded from evaluation due to overlap with training data.

Dataset	Freq.	Horizons
Agriculture	Monthly	$\{6, 8, 10, 12\}$
Climate	Monthly	$\{6, 8, 10, 12\}$
Economy	Monthly	$\{6, 8, 10, 12\}$
Energy	Weekly	$\{12, 24, 36, 48\}$
Environment	Daily	$\{48, 96, 192, 336\}$
Health†	Weekly	$\{12, 24, 36, 48\}$
Security	Monthly	$\{6, 8, 10, 12\}$
Social Good	Monthly	$\{6, 8, 10, 12\}$
Traffic	Monthly	$\{6, 8, 10, 12\}$

Time-IMM. Time-IMM (Chang et al., 2026) is an irregular multimodal multivariate time series benchmark. Unlike regular-grid benchmarks, it explicitly preserves irregular sampling, asynchronous numerical and textual timestamps, and missing observations. The benchmark contains nine datasets, each corresponding to a distinct real-world cause of irregularity: GDELT, RepoHealth, MIMIC, FNSPID, ClusterTrace, StudentLife, ILINet, CESNET, and EPA-Air. These datasets are organized into trigger-based, constraint-based, and artifact-based irregularity categories, covering event-driven logging, adaptive sampling, human-initiated observations, operational-window sampling, resource-aware collection, human scheduling, unplanned missingness, scheduling jitter, and multi-source asynchrony. Table 10 summarizes the domain, irregularity type, and official query horizon of each dataset. Following the official setting, forecasting windows are constructed within each entity, so the history and future segments of a sample never cross entity boundaries. Each window is divided into a past context segment and a future query segment, and the train/validation/test partitions are chronological 60%/20%/20% splits. Context and query window sizes are dataset-specific, reflecting the native timestamp distribution and sampling pattern of each dataset. Baseline results are directly adopted from TiMi (Lin et al., 2026). For KAIROSAGENT, we use the same official test split in the zero-shot setting and provide the model with the historical observations and available textual context before the forecast cutoff. Following TiMi (Lin et al., 2026), we exclude MIMIC and StudentLife to ensure a consistent comparison protocol. We additionally exclude ILINet as it is highly homologous with the CDC FluView ILINet data in T-STAR.

Table 10: Official Time-IMM datasets, irregularity categories, and query horizons. †Excluded following TiMi to ensure a consistent comparison protocol. ‡Excluded due to overlap with training data.

Dataset	Domain	Irregularity	Horizon
GDELT	Web	Event-based logging	14 days
RepoHealth	Web	Adaptive/reactive logging	31 days
MIMIC†	Healthcare	Human-initiated observations	24 hours
FNSPID	Econ/Fin	Operational-window sampling	31 days
ClusterTrace	CloudOps	Resource-aware collection	12 hours
StudentLife†	Healthcare	Human scheduling/availability	31 days
ILINet‡	Healthcare	Unplanned missing data/gaps	4 weeks
CESNET	Web	Scheduling jitter/delay	7 days
EPA-Air	Nature	Multi-source asynchrony	7 days

C.2 Metrics

Reasoning Metric. For morphology reasoning, we report judge accuracy. Given a generated morphology description and the held-out future values, GPT-5.2 (Singh et al., 2025) determines whether

the description is broadly consistent with the future morphology in terms of trend, periodicity, volatility, regime changes, turning points, and overall structure. Accuracy is computed as the fraction of test samples judged as correct:

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\big[\mathrm{Judge}(r^i, \mathbf{Y}_i) = 1\big], \qquad (11)$$

where N is the number of evaluated samples, r^i is the generated morphology description, and $\mathbf{Y}_i$ denotes the corresponding future window.

Forecasting Metrics. For numerical forecasting, we report Mean Squared Error (MSE) and Mean Absolute Error (MAE) as primary metrics on Time-MMD, and additionally report Mean Absolute Scaled Error (MASE) for the Time-IMM modality ablation where cross-dataset scale normalization is needed. Let Ω denote all valid forecast positions included in evaluation: for regular Time-MMD series, Ω contains all target positions in the evaluated horizon; for irregular Time-IMM series, invalid or unobserved target positions are excluded, and metrics are computed only on finite future observations at the official query timestamps. For a sample with target $\mathbf{Y}_i = (y_{i,1}, \ldots, y_{i,H})$ and prediction $\hat{\mathbf{Y}}_i = (\hat{y}_{i,1}, \ldots, \hat{y}_{i,H})$:

$$\mathrm{MSE} = \frac{1}{|\Omega|}\sum_{(i,t)\in\Omega}\left(y_{i,t} - \hat{y}_{i,t}\right)^2, \qquad (12)$$

$$\mathrm{MAE} = \frac{1}{|\Omega|}\sum_{(i,t)\in\Omega}\left|y_{i,t} - \hat{y}_{i,t}\right|. \qquad (13)$$

MASE normalizes MAE by the in-sample seasonal naive error computed from the historical context:

$$\mathrm{MASE} = \frac{\mathrm{MAE}}{\frac{1}{L-s}\sum_{t=s+1}^{L}\left|x_t - x_{t-s}\right|}, \qquad (14)$$

where $\mathbf{X} = (x_1, \ldots, x_L)$ is the history window and s is the seasonal period. For datasets without a benchmark-specific seasonality, we set $s = 1$. All reported forecasting metrics are averaged over test samples within each benchmark/domain and lower values indicate better performance.

C.3 Baselines

For forecasting evaluation, we consider three categories of models: (i) *Zero-shot TSFMs*: Aurora (Wu et al., 2025b), Sundial (Liu et al., 2025c), Moirai (Woo et al., 2024), and ChronosBolt (Ansari et al., 2024); (ii) *Full-shot multimodal models*: T3Time (Chowdhury et al., 2026), TimeCMA (Liu et al., 2025a), CALF (Liu et al., 2025b), TiMi (Lin et al., 2026), IMM-TSF (Chang et al., 2026), and Time-MMD (Liu et al., 2024), which leverage both textual and numerical modalities but require task-specific supervised training; (iii) *Full-shot unimodal models*: PatchTST (Nie et al., 2023) and DLinear (Zeng et al., 2023), which operate solely on numerical time series with full supervision. Below we describe the implementation details for each baseline to ensure a fair comparison.

C.3.1 Textual Input Configuration

Full-shot Multimodal Models. We follow the original textual-input construction protocols of each model rather than supplying benchmark metadata directly. T3Time (Chowdhury et al., 2026) and TimeCMA (Liu et al., 2025a) construct template-based prompts from the numerical history. CALF (Liu et al., 2025b) uses embedding-level textual tokens extracted from the pretrained LLM vocabulary space.

Zero-shot Multimodal Foundation Model. For Aurora (Wu et al., 2025b), we use the textual prompts provided in its original paper.

Zero-shot TSFMs and Unimodal Models. Sundial (Liu et al., 2025c), Moirai (Woo et al., 2024), ChronosBolt (Ansari et al., 2024), PatchTST (Nie et al., 2023), and DLinear (Zeng et al., 2023), do not accept textual input.

KairosAgent. The reasoner receives textual context on both Time-MMD and Time-IMM. We annotate the evaluation datasets with metadata following the same format used during training.

C.3.2 Model Configuration

Zero-shot TSFMs. We use the official pretrained checkpoints on the evaluation benchmarks, including ChronosBolt-base (205M) (Ansari et al., 2024), Moirai-large (311M) (Woo et al., 2024), and Sundial-base (128M) (Liu et al., 2025c).

Zero-shot Multimodal Foundation Model. For Aurora (Wu et al., 2025b), we also use the official pretrained checkpoint, Aurora (210.8M). For each dataset, we follow the recommended context length configuration provided by its official codebase.

Full-shot Models. For full-shot baselines, including T3Time (Chowdhury et al., 2026), TimeCMA (Liu et al., 2025a), CALF (Liu et al.,

2025b), PatchTST (Nie et al., 2023), and DLinear (Zeng et al., 2023), we follow the hyperparameter configurations recommended in the original papers for best reported performance. For the baselines in Time-IMM, the results are directly adopted from TiMi (Lin et al., 2026).

D Stability Across Random Seeds

To assess the sensitivity of KAIROSAGENT to training randomness, we repeat the Time-IMM evaluation using three independently trained checkpoints with different random seeds. We report the mean and standard deviation of MSE and MAE across the three runs. As shown in Table 11, KAIROSAGENT exhibits consistently low variance across all datasets, with the macro-average MSE of 0.689 ± 0.001 and MAE of 0.520 ± 0.000. The largest variation occurs on ClusterTrace, where sparse and irregular sampling amplifies seed-level sensitivity.

Table 11: Stability of KAIROSAGENT on Time-IMM across three random seeds (mean $\pm$ std). Lower is better.

Dataset	MSE	MAE
GDELT	$1.076 \pm .001$	$0.642 \pm .000$
RepoHealth	$0.428 \pm .002$	$0.305 \pm .000$
FNSPID	$0.111 \pm .000$	$0.180 \pm .000$
ClusterTrace	$0.891 \pm .012$	$0.672 \pm .004$
CESNET	$1.035 \pm .007$	$0.768 \pm .002$
EPA-Air	$0.591 \pm .004$	$0.556 \pm .003$
Macro Avg.	**$0.689 \pm .001$**	**$0.520 \pm .000$**

E Inference Efficiency

We evaluate the inference efficiency of the Stage II forecaster to quantify the additional overhead introduced by the multimodal fusion modules. This experiment measures only the TSFM forward pass and does not include the Stage I agent reasoning process. All models are evaluated with an input length of 2048 and a prediction horizon of 96 on a single NVIDIA TITAN RTX GPU. Table 12 reports parameter count, peak GPU memory, and wall-clock latency. Kairos-Base† denotes our unimodal backbone with horizon-decoupled heads (Appendix A.2) but without text conditioning; our forecaster adds the text encoder, projection layer, and gated cross-modal fusion on top of this backbone.

Despite introducing text-conditioned components, our forecaster remains substantially more lightweight than most zero-shot baselines, using fewer parameters and less GPU memory than Moirai-Large, Sundial-Base, ChronosBolt-Base, and Aurora. Compared to its unimodal backbone Kairos-Base†, the multimodal fusion adds only 40.6M parameters and 0.161 GB GPU memory. Notably, the latency overhead is marginal: our forecaster takes 0.075s per forward pass versus 0.054s for Kairos-Base†, an increase of only 0.021s, indicating that the cross-modal fusion path introduces minimal computational cost at inference time.

Table 12: Inference efficiency of the Stage II forecaster on a single NVIDIA TITAN RTX (input length 2048, horizon 96). †Kairos-Base with horizon-decoupled heads, serving as the unimodal backbone of our forecaster. Best results are in **red bold**, second-best in blue underline.

Model	Params (M)	GPU Mem. (GB)	Latency (s)
Moirai-Large	311.0	1.185	0.070
Sundial-Base	128.3	0.562	0.099
ChronosBolt-Base	205.3	0.786	0.055
Aurora	210.8	0.826	1.399
Kairos-Base†	**68.5**	**0.293**	**0.054**
Our Forecaster	109.1	0.454	0.075

F Full Evaluation Results

This section reports the complete forecasting results for both benchmarks. Table 13 shows the full Time-IMM results for irregular multimodal forecasting, and Table 14 presents the full Time-MMD results across all prediction lengths. The results further support the main conclusion that KAIROSAGENT delivers competitive zero-shot forecasting performance across diverse multimodal time series settings.

Table 13: Full Time-IMM forecasting results. Best results are in **red bold**, second-best in blue underline. Lower values are better.

Type	Zero-Shot Models		Full-Shot Models							
Models	KAIROSAGENT (Ours)		TiMi (2026)		IMM-TSF (2026)		Time-MMD (2024)		PatchTST (2023)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
GDELT	1.075	**0.642**	**1.029**	0.689	1.051	0.694	1.235	0.773	1.067	0.705
RepoHealth	**0.429**	**0.307**	0.532	0.416	0.552	0.418	0.579	0.453	0.573	0.419
FNSPID	**0.111**	**0.180**	0.127	0.222	0.128	0.221	0.141	0.245	0.130	0.224
ClusterTrace	0.887	**0.667**	**0.689**	0.676	1.037	0.839	0.830	0.745	2.037	0.990
ILINet	0.969	**0.569**	**0.962**	0.665	1.173	0.737	1.040	0.675	1.161	0.753
CESNET	1.040	0.769	**0.896**	**0.727**	1.124	0.819	0.971	0.761	1.318	0.880
EPA-Air	**0.583**	**0.553**	0.599	0.579	0.620	0.599	0.651	0.598	0.620	0.595
1st Count	3	**6**	**4**	1	0	0	0	0	0	0

Table 14: Full Time-MMD forecasting results across all prediction lengths. Best results are in **red bold**, second-best in blue underline. Lower values are better.

Type		Zero-Shot Models										Full-Shot Multimodal Models						Full-Shot Unimodal Models			
Models		KairosAgent (Ours)		Aurora (2025b)		Sundial (2025c)		Moirai (2024)		ChronosBolt (2024)		T3Time (2026)		TimeCMA (2025a)		CALF (2025b)		PatchTST (2023)		DLinear (2023)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Agriculture	6	**0.124**	**0.237**	0.193	0.308	0.163	0.272	0.146	0.250	0.133	0.248	0.131	0.246	0.191	0.284	0.138	0.244	0.154	0.253	0.224	0.304
	8	**0.168**	**0.268**	0.258	0.352	0.260	0.331	0.206	0.290	0.188	0.287	0.189	0.281	0.265	0.325	0.207	0.291	0.208	0.289	0.320	0.367
	10	**0.217**	**0.296**	0.299	0.361	0.366	0.394	0.266	0.324	0.243	0.318	0.263	0.320	0.356	0.383	0.266	0.324	0.271	0.321	0.411	0.417
	12	**0.270**	**0.327**	0.380	0.405	0.520	0.468	0.341	0.362	0.308	0.354	0.334	0.365	0.460	0.446	0.353	0.383	0.358	0.370	0.554	0.497
Climate	6	**0.860**	**0.736**	**0.860**	0.747	0.919	0.766	0.979	0.791	0.934	0.783	1.396	0.963	1.453	0.986	1.397	0.968	1.186	0.896	1.035	0.803
	8	0.862	**0.739**	**0.860**	0.745	0.918	0.765	0.973	0.790	0.948	0.788	1.189	0.891	1.257	0.922	1.173	0.885	1.174	0.891	1.031	0.803
	10	0.865	**0.741**	**0.864**	0.746	0.918	0.765	0.992	0.796	0.953	0.789	1.116	0.861	1.212	0.900	1.131	0.872	1.170	0.887	1.039	0.807
	12	**0.865**	**0.740**	0.868	0.748	0.926	0.766	0.984	0.793	0.957	0.790	1.124	0.863	1.206	0.897	1.096	0.854	1.173	0.889	1.040	0.812
Economy	6	0.152	0.307	0.218	0.366	0.162	0.308	0.155	0.307	**0.145**	**0.302**	0.272	0.404	0.239	0.386	0.207	0.358	0.194	0.355	0.192	0.347
	8	**0.175**	**0.328**	0.260	0.401	0.196	0.334	0.183	0.334	**0.175**	0.330	0.233	0.385	0.254	0.406	0.208	0.362	0.210	0.369	0.211	0.364
	10	**0.199**	**0.346**	0.295	0.429	0.234	0.362	0.212	0.357	0.208	0.357	0.212	0.362	0.271	0.424	0.234	0.376	0.241	0.395	0.224	0.375
	12	**0.219**	**0.360**	0.326	0.451	0.272	0.387	0.240	0.380	0.240	0.381	0.239	0.386	0.285	0.434	0.245	0.385	0.247	0.400	0.244	0.392
Energy	12	0.094	0.208	0.117	0.254	**0.091**	**0.205**	0.094	0.211	0.110	0.223	0.117	0.241	0.166	0.310	0.100	0.218	0.096	0.220	0.096	0.221
	24	**0.187**	0.309	0.220	0.349	0.191	0.311	0.194	**0.308**	0.225	0.334	0.228	0.365	0.302	0.428	0.223	0.349	0.203	0.333	0.196	0.327
	36	**0.262**	**0.375**	0.288	0.406	0.279	0.382	0.310	0.393	0.319	0.402	0.317	0.423	0.407	0.490	0.316	0.429	0.293	0.398	0.281	0.388
	48	**0.324**	**0.429**	0.380	0.470	0.377	0.451	0.444	0.476	0.398	0.459	0.400	0.484	0.530	0.559	0.394	0.496	0.379	0.462	0.361	0.447
Environment	48	0.397	0.446	**0.281**	**0.380**	0.383	0.442	0.420	0.447	0.413	0.447	0.490	0.498	0.529	0.524	0.487	0.485	0.457	0.488	0.496	0.551
	96	0.387	0.441	**0.285**	**0.382**	0.379	0.442	0.417	0.447	0.417	0.452	0.541	0.527	0.591	0.554	0.539	0.513	0.492	0.509	0.583	0.618
	192	0.370	0.430	**0.271**	**0.376**	0.374	0.442	0.404	0.443	0.420	0.459	0.534	0.543	0.541	0.537	0.581	0.530	0.527	0.532	0.683	0.702
	336	0.360	0.424	**0.269**	**0.378**	0.379	0.448	0.409	0.447	0.458	0.492	0.391	0.461	0.484	0.517	0.541	0.507	0.507	0.521	0.603	0.637
Security	6	74.071	4.122	67.570	3.896	77.447	4.512	68.090	3.898	67.642	3.831	66.019	3.988	**64.089**	**3.818**	68.267	3.886	69.745	4.203	76.819	4.593
	8	75.795	4.270	71.094	4.042	81.592	4.759	72.088	4.040	71.864	4.025	69.641	**3.949**	**69.320**	3.972	72.714	4.019	73.877	4.386	80.335	4.799
	10	77.655	4.428	74.418	4.147	85.253	4.937	76.205	4.206	75.915	4.199	**73.503**	**4.043**	75.710	4.295	74.454	4.067	79.029	4.557	84.521	5.007
	12	79.112	4.540	77.970	4.254	89.321	5.137	80.614	4.370	80.488	4.411	79.287	4.299	78.925	4.366	**77.635**	**4.188**	81.767	4.633	88.408	5.164
Social Good	6	**0.677**	0.326	0.689	0.429	0.719	**0.315**	0.750	0.333	0.792	0.332	0.783	0.377	0.901	0.457	0.734	0.354	0.779	0.402	0.749	0.389
	8	**0.750**	0.361	0.794	0.483	0.795	**0.357**	0.840	0.373	0.914	0.373	0.948	0.416	1.039	0.543	0.866	0.404	0.903	0.450	0.845	0.431
	10	**0.804**	**0.395**	0.879	0.534	0.854	0.398	0.910	0.411	1.009	0.409	1.072	0.450	1.156	0.620	0.953	0.443	1.029	0.502	0.939	0.468
	12	**0.847**	**0.423**	0.952	0.580	0.909	0.435	0.973	0.448	1.091	0.440	1.191	0.483	1.274	0.691	1.007	0.461	1.123	0.545	1.029	0.505
Traffic	6	**0.148**	**0.228**	0.155	0.286	0.210	0.272	0.172	0.251	0.201	0.229	0.346	0.439	0.349	0.457	0.273	0.372	0.202	0.307	0.216	0.319
	8	**0.150**	**0.229**	0.160	0.287	0.222	0.285	0.182	0.259	0.214	0.242	0.297	0.373	0.287	0.402	0.212	0.285	0.206	0.314	0.221	0.317
	10	**0.151**	**0.230**	0.163	0.289	0.233	0.297	0.190	0.266	0.226	0.255	0.250	0.326	0.271	0.393	0.208	0.279	0.211	0.320	0.218	0.312
	12	**0.155**	**0.236**	0.169	0.295	0.248	0.312	0.200	0.274	0.247	0.269	0.264	0.335	0.282	0.395	0.217	0.285	0.218	0.325	0.223	0.310
1st Count		**20**	**19**	7	4	1	3	0	1	2	1	1	2	2	1	1	1	0	0	0	0

G Showcases

We present four representative model rollouts to illustrate different capabilities of KairosAgent. Each case is selected to highlight a distinct scenario: Table 15 shows strong periodic structure (SHMetro), Table 16 shows sparse intermittent patterns (Alibaba Cluster Trace), Table 17 shows irregular burst-like activity (Extended Web Traffic), and Table 18 demonstrates reasoning without metadata (Electricity Hourly). Throughout the rollouts, gray denotes prompt and final output, green denotes agent reasoning, red denotes tool calls, and blue denotes tool observations. Long history values, tool schemas, and repeated interactions are abbreviated with ellipses.

Table 15: Model rollout on SHMetro (Transport, 15T). Demonstrates tool-grounded reasoning on a strongly periodic time series with clear cyclical structure.

Prompt: Dataset: SHMetro; Domain: Transport; Frequency: 15T; Variable: inbound_passenger_count; History window: 2016-08-02 05:30:00 to 2016-08-23 13:15:00; Future window (short term): 2016-08-23 13:30:00 to 2016-08-24 13:15:00, horizon: 96; Future window (long term): 2016-08-23 13:30:00 to 2016-08-31 01:15:00, horizon: 720. History values: 0, 13.00, 22.00, 50.00, 72.00, 120.0, 168.0, 264.0, 457.0, . . . , 177.0, 154.0, 241.0, 198.0, 235.0.

Reasoning: The user wants me to forecast the morphology of a time series for short-term and long-term horizons. I need to inspect a broad window covering most of the history and a targeted local window near the end. Let me first look at the broad statistics and patterns, then zoom in on the recent regime. . . .

Tool call: {"call_id": "linear_trend:0", "name": "linear_trend", "arguments": {"left": 0, "right": 2048}}

Observation: {"slope": -0.0165, "intercept": 195.8754, "rvalue": -0.0519, "pvalue": 0.0187, "stderr": 0.0070}

Reasoning: The linear trend shows a very slight negative slope with weak correlation, suggesting the series is roughly stationary in terms of level over the long term. Now let me look at the recent regime near the end of the history to understand the current state. . . .

Tool call: {"call_id": "standard_deviation:1", "name": "standard_deviation", "arguments": {"left": 0, "right": 2048}}

Observation: {"value": 187.5966}

Tool call: {"call_id": "mean_abs_change:2", "name": "mean_abs_change", "arguments": {"left": 0, "right": 2048}}

Observation: {"value": 35.9497}

Reasoning: The recent window shows a slight positive trend but still weak correlation. The recent standard deviation is similar to the overall series, and the autocorrelation suggests weak or complex periodicity. Let me look at more specific features to understand the recurrent structure and peaks. . . .

Tool call: {"call_id": "linear_trend:3", "name": "linear_trend", "arguments": {"left": 1856, "right": 2048}}

Observation: {"slope": 0.4720, "intercept": 135.3051, "rvalue": 0.1308, "pvalue": 0.0706, "stderr": 0.2596}

Tool call: . . .

Observation: . . .

Reasoning: The additional evidence indicates many peaks, a large-amplitude extreme range, and repeated crest–trough behavior. The final response should avoid exact numbers, timestamps, dataset names, units, and causal stories, and describe only the expected morphology for the two horizons. . . .

Final morphology forecast:
In the short term, the series will likely transition from its current moderate level into a sharp ascent toward a pronounced crest, followed by sustained elevated activity through the subsequent periods before reaching a secondary peak and then declining steeply toward minimal levels. The pattern should replicate the established cyclical structure with distinct high-amplitude crests separated by intermediate plateaus, maintaining the sharp transition characteristics observed in recent history. The volatility during the upcoming elevated phase may vary slightly but should remain consistent with the recent regime's amplitude range.
In the long term, the morphology should continue to exhibit strong periodic repetition over multiple cycles, maintaining the characteristic bimodal crest structure with alternating high-intensity periods and deep troughs. The overall level appears approximately stationary with negligible long-term drift, suggesting that the amplitude and range of oscillations will persist without systematic trend. The pattern will likely retain its sharp asymmetry between rapid ascents and descents, with intermittent variations in peak intensity reflecting the inherent irregularity of the underlying process.

Table 16: Model rollout on Alibaba Cluster Trace (CloudOps, 5T). Demonstrates reasoning on a sparse, intermittent time series with predominantly near-zero values and rare isolated spikes.

Prompt: Dataset: alibaba_cluster_trace_2018; Domain: CloudOps; Frequency: 5T; Variable: cpu_utilization_percent; History window: 2018-01-02 12:00:00 to 2018-01-05 23:55:00; Future window (short term): 2018-01-06 00:00:00 to 2018-01-06 07:55:00, horizon: 96; Future window (long term): 2018-01-06 00:00:00 to 2018-01-08 11:55:00, horizon: 720. History values: 0, 0, 0, 0, 0, 0, 0, 0, 0, . . . , 0.03333, 0, 0, 0, 0, 0, 0.

Reasoning: The user wants me to analyze a time series of CPU utilization percentage. The history consists mostly of zeros with occasional small spikes, suggesting a very sparse, intermittent time series with low activity. I need to inspect a broad window covering most of the history and a targeted local window near the end. Let me start with the broad window analysis. . . .

Tool call: {"call_id": "linear_trend:0", "name": "linear_trend", "arguments": {"left": 0, "right": 1008}}

Observation: {"slope": 4.7536e-07, "intercept": 0.000259, "rvalue": 0.0319, "pvalue": 0.3116, "stderr": 4.6956e-07}

Tool call: {"call_id": "autocorrelation:1", "name": "autocorrelation", "arguments": {"left": 0, "right": 1008, "lag": 288}}

Observation: {"value": 0.0644}

Tool call: {"call_id": "standard_deviation:2", "name": "standard_deviation", "arguments": {"left": 0, "right": 1008}}

Observation: {"value": 0.0043}

Reasoning: The linear trend is practically zero and not significant, so there is no meaningful global trend. The standard deviation is very low, indicating most values are near zero, and the long-lag autocorrelation is weak, suggesting no strong periodic structure. Now let me check the recent window to see the short-term regime. . . .

Tool call: {"call_id": "linear_trend:3", "name": "linear_trend", "arguments": {"left": 800, "right": 1008}}

Observation: {"slope": 9.1124e-07, "intercept": 0.000386, "rvalue": 0.0138, "pvalue": 0.8435, "stderr": 4.6112e-06}

Tool call: {"call_id": "number_peaks:4", "name": "number_peaks", "arguments": {"left": 800, "right": 1008, "n": 1}}

Observation: {"value": 3}

Tool call: {"call_id": "mean_abs_change:5", "name": "mean_abs_change", "arguments": {"left": 800, "right": 1008}}

Observation: {"value": 0.000966}

Reasoning: The recent window also shows no significant trend and very low mean absolute change, indicating mostly flat zeros with occasional small spikes. There are only a few peaks in this local window. Let me check the distribution of values and whether excursions are brief or clustered. . . .

Tool call: {"call_id": "quantile:6", "name": "quantile", "arguments": {"left": 0, "right": 1008, "q": 0.95}}

Observation: {"value": 0.0}

Tool call: . . .

Observation: . . .

Reasoning: The high quantile of the whole history is zero, meaning nearly all values are at the baseline. Excursions above the mean are very brief, and the recent autocorrelation does not indicate clustering. The final forecast should therefore describe a stationary sparse pattern with prolonged near-zero plateaus and rare isolated spikes, while avoiding exact numbers, timestamps, dataset names, units, and domain-specific causal stories. . . .

Final morphology forecast:
In the short term, the series will maintain its established sparse intermittent morphology characterized by prolonged near-zero plateaus punctuated by occasional isolated spikes of limited amplitude. The volatility will remain low with no discernible trend, and any excursions above the baseline will be brief and scattered rather than clustered. The pattern of activity will closely mirror the recent regime, with the vast majority of observations holding at minimal levels and sporadic small increases appearing randomly in time without sustained elevation or systematic drift.
In the long term, this stationary low-activity regime will persist throughout the horizon, continuing the historical pattern of predominantly flat near-zero values interrupted by infrequent small spikes. There is no indication of emerging trend, seasonal structure, or regime transition that would alter the fundamental intermittency. The amplitude distribution and sparsity rate will remain consistent with the observed history, preserving the morphology of a quiescent baseline with rare, short-lived minor perturbations.

Table 17: Model rollout on Extended Web Traffic (Web, D). Demonstrates reasoning on an irregular, burst-like time series with missing values and no strong periodicity.

Prompt: Dataset: extended_web_traffic; Domain: Web; Frequency: D; Variable: daily_page_views; History window: 2015-07-01 00:00:00 to 2020-07-10 00:00:00; Future window (short term): 2020-07-11 00:00:00 to 2020-10-14 00:00:00, horizon: 96; Future window (long term): 2020-07-11 00:00:00 to 2022-06-30 00:00:00, horizon: 720. History values: 2.000, 0, 4.000, 6.000, 3.000, 0, 0, 7.000, 6.000, . . . , 10.00, 7.000, 7.000, 5.000.

Reasoning: The user wants me to analyze a daily web traffic time series and forecast the morphology for short-term and long-term horizons. The values show many small observations with occasional larger spikes, so I need to inspect the broad historical pattern and then the recent regime near the end. Let me start with a broad analysis of the full history. . . .

Tool call: {"call_id": "linear_trend:0", "name": "linear_trend", "arguments": {"left": 0, "right": 1837}}

Observation: {"slope": 0.000883, "intercept": 3.9672, "rvalue": 0.0715, "pvalue": 0.0022, "stderr": 0.000287}

Tool call: {"call_id": "standard_deviation:1", "name": "standard_deviation", "arguments": {"left": 0, "right": 1837}}

Observation: {"value": 6.5471}

Tool call: {"call_id": "mean_abs_change:2", "name": "mean_abs_change", "arguments": {"left": 0, "right": 1837}}

Observation: {"value": 3.3034}

Reasoning: The broad history shows a very weak positive trend, noticeable but small relative to the variability, and frequent local changes. Now I should look at the recent regime to see whether the latest portion has a different level, volatility, or trend. . . .

Tool call: {"call_id": "linear_trend:3", "name": "linear_trend", "arguments": {"left": 1600, "right": 1837}}

Observation: {"slope": 0.0112, "intercept": 5.0009, "rvalue": 0.1761, "pvalue": 0.0066, "stderr": 0.0041}

Tool call: {"call_id": "standard_deviation:4", "name": "standard_deviation", "arguments": {"left": 1600, "right": 1837}}

Observation: {"value": 4.3447}

Tool call: {"call_id": "fft_coefficient:5", "name": "fft_coefficient", "arguments": {"left": 0, "right": 1837, "coeffs": [1, 7, 30, 365], "attr": "abs"}}

Observation: {"coeff_1": 639.5266, "coeff_7": 567.0431, "coeff_30": 519.6249, "coeff_365": 385.2758}

Reasoning: The recent window has a stronger but still weak upward tendency, and its volatility is lower than the broad-history volatility. Spectral magnitudes suggest possible recurrent components, but I need direct checks for autocorrelation, peak frequency, and outlier behavior. . . .

Tool call: {"call_id": "autocorrelation:6", "name": "autocorrelation", "arguments": {"left": 1500, "right": 1837, "lag": 7}}

Observation: {"value": 0.0109}

Tool call: . . .

Observation: . . .

Reasoning: The recent period has moderate typical levels, many local peaks, weak short-lag autocorrelation, and only a small fraction of extreme observations. This supports an intermittent, burst-like morphology with bounded recent volatility, occasional higher excursions, and no strong periodic regularity. The final response must avoid exact numbers, timestamps, dataset names, units, and domain labels. . . .

Final morphology forecast:
In the short term, the series should maintain the moderate activity level observed in recent periods, with typical values clustering at low to moderate magnitudes and occasional excursions into higher ranges. Sporadic bursts of elevated activity will likely appear intermittently, reflecting the historical pattern of irregular surges. Volatility should remain relatively constrained compared to earlier historical extremes, with runs of consecutive observations showing similar moderate levels before brief transitions. Missing value gaps may continue to appear sporadically throughout the window. In the long term, the morphology should persist as an intermittent series characterized by predominantly modest counts punctuated by infrequent larger spikes. A very gradual upward drift visible in recent history may continue subtly, but the overall amplitude and variability will likely remain bounded within historical ranges. The pattern of many low-value observations mixed with scattered moderate peaks and rare high outliers should dominate, without developing strong periodic regularity or sustained directional growth. The series will likely retain its erratic, burst-like nature rather than smoothing into predictable cycles.

Table 18: Model rollout on Electricity Hourly with masked metadata. Demonstrates the agent reasoning purely from numeric history and tool evidence when no textual context is available.

Prompt: Metadata availability: unavailable. Avoid exact numbers and timestamps in final morphology output. History window: start_time: unavailable; end_time: unavailable; length: 2048. Future window (short term): start_time: unavailable; end_time: unavailable; horizon: 96. Future window (long term): start_time: unavailable; end_time: unavailable; horizon: 720. History values: 4467, 5182, 6249, 6326, 6117, 6062, 6227, . . . , 5743, 5666, 6326, 6315, 5842, 6282, 6315, 6601.

Reasoning: The metadata is unavailable, so I need to rely purely on the numeric values and tool-based analysis. I should inspect the broad window covering most of the history and a targeted local window near the end. Let me start with the most important trend and volatility features. . . .

Tool call: {"call_id": "linear_trend:0", "name": "linear_trend", "arguments": {"left": 0, "right": 2048}}

Observation: {"slope": -0.1042, "intercept": 4798.8373, "rvalue": -0.0391, "pvalue": 0.0768, "stderr": 0.0589}

Tool call: {"call_id": "linear_trend:1", "name": "linear_trend", "arguments": {"left": 1900, "right": 2048}}

Observation: {"slope": 1.2326, "intercept": 4462.9935, "rvalue": 0.0370, "pvalue": 0.6553, "stderr": 2.7554}

Tool call: {"call_id": "standard_deviation:2", "name": "standard_deviation", "arguments": {"left": 1800, "right": 2048}}

Observation: {"value": 1471.5611}

Reasoning: The full window has only a very slight negative trend with low correlation, essentially flat. The recent window has a slight positive slope but also very low correlation and is not significant. The recent volatility is high, so I need to examine autocorrelation, periodicity, and peaks. . . .

Tool call: {"call_id": "autocorrelation:3", "name": "autocorrelation", "arguments": {"left": 0, "right": 2048, "lag": 100}}

Observation: {"value": 0.2789}

Tool call: {"call_id": "autocorrelation:4", "name": "autocorrelation", "arguments": {"left": 1500, "right": 2048, "lag": 50}}

Observation: {"value": 0.7293}

Tool call: {"call_id": "number_peaks:5", "name": "number_peaks", "arguments": {"left": 1900, "right": 2048, "n": 5}}

Observation: {"value": 7}

Reasoning: The full-window autocorrelation is moderate, while the recent-window autocorrelation is high, suggesting strong serial correlation or periodicity in the recent regime. Several recent peaks indicate oscillatory behavior. Let me check change magnitude, spectral structure, and stationarity. . . .

Tool call: {"call_id": "mean_abs_change:6", "name": "mean_abs_change", "arguments": {"left": 1900, "right": 2048}}

Observation: {"value": 392.0340}

Tool call: . . .

Observation: . . .

Reasoning: The recent window ends near a local maximum after rising from an earlier trough. Combined with stationarity and negative aggregated autocorrelation, this suggests mean reversion: after peaks the series tends to fall, and after troughs it tends to recover. The forecast should describe a near-term decline from an elevated point and a long-term bounded oscillation, without using unavailable metadata. . . .

Final morphology forecast:
In the short term, the series is positioned near a local peak following a sustained ascent, suggesting an imminent reversal or slowdown. Given the strong mean-reverting behavior and stationary character of the recent regime, the trajectory should pivot toward declining values, descending from the current elevated level with moderate-to-high volatility. The oscillatory nature implies a downward arc that may accelerate initially before finding support, with rapid fluctuations typical of the observed regime transitions.
In the long term, the morphology should persist as a bounded, non-trending oscillation between contrasting elevation states. The series exhibits a rhythmic alternation between prolonged plateaus at higher levels and sharp descents into lower valleys, followed by gradual recoveries. This cyclical pattern of expansion and contraction should continue, maintaining the established range without sustained directional drift, characterized by intermittent spikes and troughs with asymmetric transition speeds.