

ChronoScribe: Semantic–Numerical Specialization for Instruction-Driven Time Series Forecasting

Yuchen Jiao, Miao Zhang, Yuxuan Liu, Chengxi Li

Beijing University of Posts and Telecommunications
<https://github.com/Thekigs/ChronoScribe>

Abstract

Recent advances in instruction-driven time series forecasting highlight a fundamental tension: systems must simultaneously master natural-language instruction following and precise numerical prediction. Current approaches often force a compromise. Large language models (LLMs) can interpret unstructured requests but may be inaccurate when asked to produce forecast values directly, whereas time series foundation models (TSFMs) provide numerical forecasts but typically require structured inputs specifying the target series, context window, and horizon. We present **ChronoScribe**, which applies semantic–numerical specialization to this workflow. A LoRA-adapted LLM proposes an execution path and context length; deterministic rules resolve the executable configuration. On the specialist path, a frozen TSFM produces quantile trajectories for the task-specified horizon. Deterministic trajectory summaries condition the response, while the LLM neither generates nor revises predicted values. We evaluate a Qwen3.5-9B–Chronos-2 implementation across 23 evaluation dimensions and 30 public datasets against eleven proprietary direct LLMs and three open TSFMs. ChronoScribe has the lowest error on sMAPE, MASE, WAPE, RMSE, and MAE among all evaluated systems, including fixed-context controls. Across these metrics, even the lowest direct-LLM error remains $2.0\text{--}2.7\times$ the corresponding ChronoScribe error. At the system level, ChronoScribe achieves a Task-Level Score (TLS) of 65.5, 10.6 points above the strongest direct-LLM baseline. The results support ChronoScribe’s division of labor, in which the LLM handles interpretation, configuration, and communication while the TSFM handles numerical forecasting.

Introduction

Time series forecasting is commonly formulated as the task of estimating future values from an observed history over a specified horizon. In practice, users may use natural language to request a forecast together with a trend summary or uncertainty statement. A request such as “forecast demand for the coming week and summarize the trend and uncertainty” requires a system to interpret the request, configure and run a forecaster, and report the result. Large language models (LLMs) and time series foundation models (TSFMs) can support this workflow, but they are suited to different parts of it.

TSFMs support zero-shot and probabilistic forecasting through large-scale pretraining (Ansari et al. 2025), but gen-

erally expect a structured interface that specifies the target series, horizon, and context window. LLMs can interpret open-ended instructions and produce natural-language responses, yet their training objectives do not directly optimize scale fidelity or temporal forecast accuracy (Gruver et al. 2023; Jin et al. 2024). Using a language model for both request interpretation and trajectory generation combines tasks with different modeling requirements. The resulting design problem is to let an LLM configure a numerical specialist without directly generating or revising the forecast trajectory.

Some prior approaches use language models to generate or represent time series; others use textual context to steer or revise forecasts (Gruver et al. 2023; Jin et al. 2024; Wang et al. 2025; Zhao et al. 2025; Liao et al. 2026). We focus on the interface between the two model classes. Such an interface must translate an open-ended request into a valid specialist call and communicate the result without becoming a second source of predicted values. Numerical provenance therefore depends on keeping the directive, specialist output, and response distinct.

Separating these stages makes numerical provenance easier to trace and failures easier to diagnose. A direct LLM must preserve scale, units, and long-range temporal structure while composing a fluent answer. A fixed-context specialist avoids direct language-model forecasting but cannot adapt its history length to individual requests. Figure 1 contrasts direct LLMs, fixed TSFMs, and ChronoScribe. Allowing the language model to generate or revise forecast values would obscure their source. A typed interface therefore exposes the proposal, executable configuration, specialist output, and final response as separate artifacts.

We call this division of responsibility *semantic–numerical specialization*. The LLM handles request interpretation, specialist configuration, and communication, whereas the TSFM alone generates forecast values. ChronoScribe implements this principle with a LoRA-adapted language model that emits a typed directive. The directive proposes a direct language path or a specialist path and, in the latter case, a context length. Deterministic rules parse and normalize the proposal or apply a fallback; raw language-model output is not executed directly. The task, rather than the directive, supplies the forecast horizon. A frozen TSFM then produces quantile trajectories from the selected history. Deterministic functions

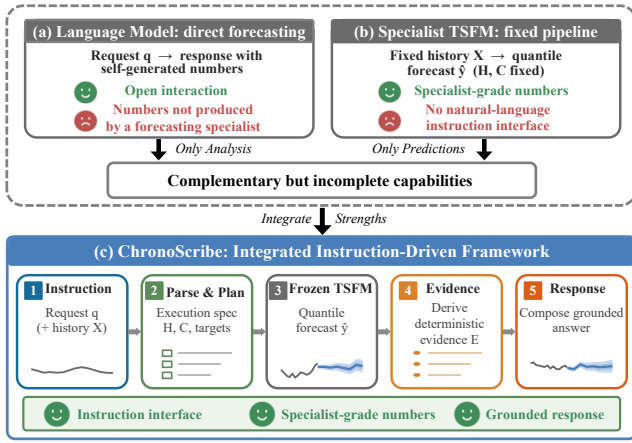


Figure 1: Direct LLMs, fixed TSFMs, and ChronoScribe.

derive numerical summaries, a direction label, and uncertainty intervals from these trajectories. The adapted language model uses this evidence to compose the response without generating or revising the trajectory.

Our evaluation spans 23 dimensions in forecasting, temporal reasoning, instruction following, uncertainty, and efficiency, using 30 public datasets. We compare ChronoScribe with eleven proprietary direct LLMs, three open TSFMs, fixed-context controls, and component ablations. ChronoScribe has the lowest sMAPE, MASE, WAPE, RMSE, and MAE among all evaluated systems, including fixed-context controls. Across these metrics, the lowest direct-LLM error remains $2.0\text{--}2.7\times$ the corresponding ChronoScribe error. Its predefined Task-Level Score (TLS), a weighted aggregate of the five capability groups, is 65.5, 10.6 points above the strongest direct-LLM baseline. The controller returns a parseable directive with the protocol-defined route for 95.9% of requests. In a rule-based audit of 100 specialist-path responses, 87.4% of 230 claims agree with deterministic forecast evidence, and 89 responses pass every applicable check.

This work contributes (i) ChronoScribe, a framework for semantic-numerical specialization that assigns request interpretation and evidence-based communication to an adapted LLM while reserving trajectory generation for a frozen TSFM; (ii) typed execution and forecast-evidence interfaces that allow raw directives, executable configurations, specialist forecasts, and evidence-conditioned responses to be inspected separately; and (iii) a 23-dimension evaluation across 30 public datasets against eleven proprietary direct LLMs, three open TSFMs, fixed-context controls, and component ablations, together with audits of directive reliability and forecast-claim grounding.

Related Work

Transformer forecasters such as Informer, PatchTST, iTransformer, and TimesNet use sparse attention, patching, inverted tokenization, or multi-scale representations for long-horizon prediction (Zhou et al. 2021; Nie et al. 2023; Liu et al. 2024b; Wu et al. 2023). Time series foundation models, including Chronos, Chronos-2, TimesFM, Moirai, Timer, and MO-

MENT, are pretrained on heterogeneous series for zero-shot or transfer forecasting (Ansari et al. 2024, 2025; Das et al. 2024; Woo et al. 2024; Liu et al. 2024c; Goswami et al. 2024). At inference, these models operate on a selected series with a specified context window and horizon rather than on an open-ended request.

Several approaches incorporate pretrained language models into numerical forecasting. LLTime serializes observations as text-like tokens and generates continuations directly. GPT4TS, TEMPO, UniTime, Time-LLM, TimeCMA, and CALF use fine-tuning, prompting, reprogramming, or cross-modal alignment to adapt language models or their representations (Gruver et al. 2023; Zhou et al. 2023; Cao et al. 2024; Liu et al. 2024a; Jin et al. 2024; Liu et al. 2025a,b). M3Time, T3Time, and ExoTimer incorporate multimodal, multiscale, or exogenous information (Jia et al. 2026; Chowdhury, Akter, and Arib 2026; Wu et al. 2026). In these systems, language-model generation or representations contribute to the numerical forecast. ChronoScribe assigns request interpretation, specialist configuration, and response composition to the LLM, while the frozen TSFM alone produces the trajectory.

Tool use offers a precedent for separating language interpretation from external execution (Schick et al. 2023; Yao et al. 2023). ChronoSteer converts textual events into revision instructions that steer a TSFM (Wang et al. 2025). TimeSeriesScientist assigns data diagnosis, model selection, forecasting, and reporting to specialized agents (Zhao et al. 2025). Last-mile forecasting agents retrieve contextual evidence and apply explicit revision actions to a forecasting backbone (Liao et al. 2026). In ChronoScribe, the LLM proposes whether and how to invoke a frozen specialist and communicates deterministic summaries of its output; it does not revise the specialist trajectory.

Established forecasting evaluations focus on numerical outputs, using point-error metrics and proper scoring rules (Makridakis, Spiliotis, and Assimakopoulos 2020; Hyndman and Koehler 2006; Gneiting and Raftery 2007). GIFT-Eval broadens zero-shot TSFM evaluation across datasets, domains, frequencies, and forecast lengths (Aksu et al. 2024). These evaluations do not test request interpretation or whether a language response is supported by forecast evidence. Our protocol retains the raw forecasting metrics and adds request interpretation, instruction compliance, uncertainty communication, and execution cost.

Problem Formulation

Instruction-driven Forecasting Setting

Let q denote a natural-language request and $X_{1:T} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ the available history, where $\mathbf{x}_t \in \mathbb{R}^d$; setting $d = 1$ gives the univariate case. Each task also supplies $\tau = (H_\tau, j^*)$, comprising the forecast horizon H_τ and target index j^* , with $j^* = 1$ for a univariate series. Given $(q, X_{1:T}, \tau)$, the system decides whether to invoke a specialist and, if so, how much history to pass to it. The evaluation therefore holds the horizon fixed and measures routing and context configuration rather than autonomous horizon inference. A request may additionally specify trend, uncertainty,

or response-format requirements.

Forecast Specification and Evidence Construction

We separate the language model’s raw directive from the executable configuration. Define the target history as $X_{1:T}^{(j^*)} = (x_{1,j^*}, \dots, x_{T,j^*})$. Its deterministic representation $\rho(X_{1:T}^{(j^*)})$ contains the history length, summary statistics, a linear-trend estimate, a uniformly subsampled trajectory, and recent observations. Let $a_X \in \{0, 1\}$ indicate whether this history satisfies the specialist’s preprocessing requirements. The directive proposes a path and context length; it also repeats the supplied horizon for schema compliance. The latter field does not determine the executable horizon:

$$\begin{aligned} \hat{s} &= \Psi_\theta(q, \rho(X_{1:T}^{(j^*)}), \tau), \\ (u, C) &= \Gamma(\hat{s}; q, X_{1:T}^{(j^*)}, \tau), \quad H = H_\tau. \end{aligned} \quad (1)$$

The specialist runs only when the resolved action selects it and the target history is admissible:

$$Y = \begin{cases} F_\phi(X_{T-C+1:T}^{(j^*)}; H), & u = 1 \text{ and } a_X = 1, \\ \emptyset, & \text{otherwise,} \end{cases} \quad (2)$$

When $Y \neq \emptyset$, the deterministic map $e = g(Y)$ formats the forecast into an evidence record, and the language model samples $r \sim p_\theta(\cdot | q, e)$. When $Y = \emptyset$, the language model instead samples $r \sim p_\theta(\cdot | q)$. Here, $\hat{s}$ is the JSON-formatted directive, and $(u, C) \in \mathcal{Z}$ is the resolved action. The variable $u \in \{0, 1\}$ selects the path, while $C \in \mathbb{N}_+$ is clipped to the admissible range for the available history and forecaster. The deterministic operator Γ parses and normalizes $\hat{s}$ or applies a fallback if parsing fails. The repeated horizon field is audited for compliance, but H_τ always determines execution. Only the frozen forecaster F_ϕ generates future values; the language model produces either an evidence-conditioned response or a direct response.

Forecast-grounded Response Generation

For each forecast-claim type $k \in \mathcal{K} = \{\text{num}, \text{dir}, \text{unc}\}$, let $\text{Claims}_k(r)$ denote the numerical, directional, or uncertainty claims expressed in the specialist-path response r , and let $\text{Derivable}_k(e)$ denote the corresponding statements supported by the forecast evidence. We call a response *forecast-grounded* when

$$\text{Claims}_k(r) \subseteq \text{Derivable}_k(e), \quad \forall k \in \mathcal{K}. \quad (3)$$

Equation (3) is the traceability test used in our audit, not a constraint enforced at runtime. Grounding does not require the response to reproduce the complete trajectory; it requires every forecast-related claim to be supported by a deterministic transformation of the specialist output. The test measures agreement with specialist-derived evidence, not forecast accuracy or calibration. It is applied only to the specialist path because a direct language response has no forecast evidence.

ChronoScribe Framework

Overview

ChronoScribe supports a direct language path and a specialist path. Figure 2 outlines this workflow. Every request first passes through the adapted language model, which emits a structured directive that deterministic rules resolve into a typed action. On the direct path, the model answers without invoking the forecaster. On the specialist path, a frozen forecaster receives the selected history window, and deterministic functions convert its quantile trajectories into an evidence record for the response. The same adapted language model emits the directive and composes the final response; the forecaster remains fixed.

Routing and Forecast Specification

The language component produces the raw directive $\hat{s}$ in Eq. (1). The parser $\mathcal{P}$ maps it to $\tilde{z} = \mathcal{P}(\hat{s})$, with $\tilde{z} = \perp$ denoting failure. If parsing succeeds, Γ applies the normalizer $\mathcal{N}(\tilde{z}; T, \tau)$, which casts fields to their required types and clips C to the admissible range. If parsing fails, Γ instead applies the deterministic fallback $\mathcal{B}(q, X_{1:T}^{(j^*)}, \tau)$, which uses a predefined route and context policy. Both cases return the same typed action (u, C) ; raw language-model output is never executed directly. A specialist action is executed only when the historical observations also pass the validity check a_X . Parsing failures are counted as controller failures even if the fallback allows the request to complete.

The resolved action controls only the route and the history suffix supplied to the specialist. It cannot change the externally specified horizon H_τ or the specialist parameters ϕ ; F_ϕ produces all forecast values.

Specialist Forecasting, Evidence, and Response

On the specialist path, ChronoScribe extracts the selected history window of the target variable, $X^{(C,j^*)} = X_{T-C+1:T}^{(j^*)}$. In our implementation, the frozen specialist receives this subsequence and returns quantile trajectories over H future steps:

$$\begin{aligned} Y &= F_\phi(X^{(C,j^*)}; H) \in \mathbb{R}^{|\mathcal{A}| \times H}, \\ Y_{\alpha,h} &= \hat{x}_{T+h,j^*}^{(\alpha)}, \quad \alpha \in \mathcal{A}, \quad h \in \{1, \dots, H\}. \end{aligned} \quad (4)$$

The symbol ϕ denotes the frozen specialist parameters, $\mathcal{A}$ is the set of quantile levels, and $Y_{\alpha,h}$ is the α -quantile prediction for the target variable at step h . Our Chronos-2 instantiation uses $\mathcal{A} = \{0.1, 0.5, 0.9\}$, corresponding to P10, P50, and P90. The forecaster receives the task-specified univariate subsequence. The framework requires the specialist to accept the selected history and horizon and return at least a point forecast. Probabilistic output is needed only to construct uncertainty evidence.

The deterministic map g converts the quantile trajectories into the fixed evidence record

$$e = g(Y) = (e_{\text{num}}, e_{\text{dir}}, e_{\text{unc}}, e_{\text{aux}}).$$

The component e_{num} stores the initial, final, mean, minimum, and maximum P50 values together with relative

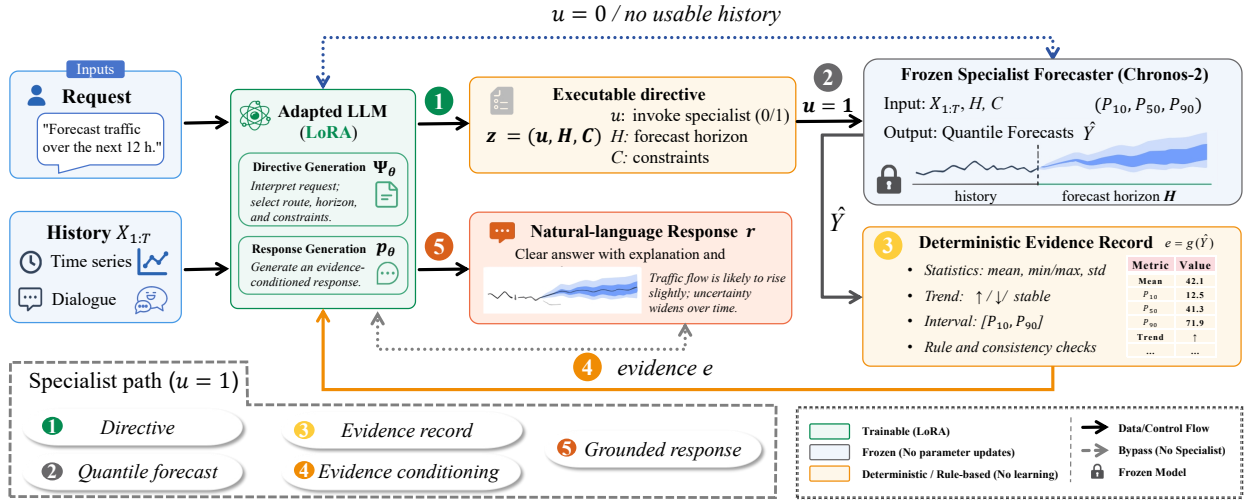


Figure 2: Typed-directive workflow with deterministic evidence.

change. The component e_{dir} records a trend label derived using fixed thresholds, and e_{unc} stores interval information from P10 and P90. The optional e_{aux} holds task-specific summaries such as peak properties; $e_{\text{aux}} = \emptyset$ when none is required. The grounding analysis covers numerical, directional, and uncertainty claims; auxiliary claims are outside its scope. For multivariate inputs, every summary refers to the task-specified target variable. Quantile-ordering checks, direction thresholds, and peak-detection rules are fixed before evaluation and documented in the supplementary protocol.

The adapted language model generates a response conditioned on the original request and the evidence record. Figure 3 gives a specialist-path example. Because the response remains free-form text, separating the language model from trajectory generation does not by itself guarantee Eq. (3); the grounding audit checks each forecast-related claim against the evidence.

Language Model Adaptation

We train the language component on two interface tasks. A directive-generation record uses a request, history representation, and task constraints as input and a structured directive containing the route and context length as its target. A response-generation record uses a request and forecast evidence as input and an evidence-supported response as its target. Both tasks update a single LoRA adapter (Hu et al. 2022); the resulting model produces both directives and evidence-conditioned responses.

Let $c_i = (q_i, \rho_i, \tau_i)$ denote a directive input, where ρ_i is the deterministic history representation defined above. Let $\mathcal{D}_z = \{(c_i, z_i)\}_{i=1}^{n_z}$ denote the directive-generation examples and $\mathcal{D}_r = \{((q_j, e_j), r_j)\}_{j=1}^{n_r}$ the evidence-to-response examples. We combine them as $\mathcal{D} = \mathcal{D}_z \cup \mathcal{D}_r$ and optimize only the LoRA parameters θ using the target-token-averaged autoregressive objective

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N} \sum_{(a,b) \in \mathcal{D}} \sum_{t=1}^{|b|} \log p_\theta(b_t | a, b_{<t}), \quad (5)$$

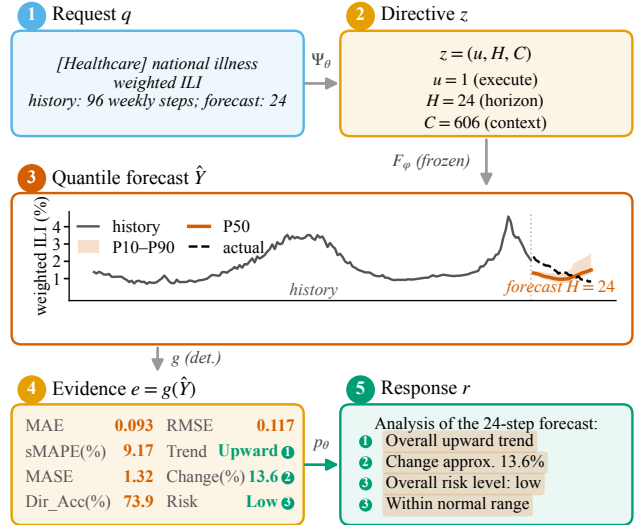


Figure 3: End-to-end specialist-path example.

where $(a, b) = (c_i, z_i)$ for directive-generation examples and $(a, b) = ((q_j, e_j), r_j)$ for evidence-to-response examples. The normalization factor $N = \sum_{(a,b) \in \mathcal{D}} |b|$ is the total number of supervised target tokens. Prompt and padding tokens are masked from the loss. The adapter is trained only for directive and response generation; future numerical trajectories never appear as language-model targets.

The adaptation records use public time series datasets that are disjoint from the 30 evaluation datasets. No evaluation series, windows, or future targets are used during training. The supplementary material documents the source datasets, split rules, prompt templates, transformation procedures, and output schemas used to construct the records.

LoRA is applied to all attention and MLP projection layers with rank 64, scaling parameter 128, and dropout 0.05. Training uses AdamW for two epochs with an effective batch

size of 32, a learning rate of 2×10^{-4} , weight decay 0.01, and a 5% linear warmup followed by cosine decay. The random seed is 42. The base Qwen3.5-9B model and the Chronos-2 forecaster remain frozen throughout training.

Experiments

Experimental Setup and Evaluation

Our evaluation covers request interpretation, forecast execution, and response generation. The suite comprises 23 evaluation dimensions divided into five groups, with ten forecasting, six temporal-reasoning, five instruction-following, one uncertainty, and one efficiency dimension. The forecasting dimensions measure point accuracy, direction, peak behavior, cross-domain transfer, robustness, and horizon degradation. The remaining dimensions evaluate temporal judgments, instruction compliance, probabilistic communication, and inference cost.

The benchmark pool contains 8,120 source instances drawn from 30 public datasets in 14 families spanning energy, transportation, weather, finance, healthcare, retail, and monitoring. Primary comparisons use a fixed manifest of 460 records, with 20 records per dimension selected once with seed 2026. Each record contains a request q and an observed history $X_{1:T}$; the future segment $X_{T+1:T+H}^*$ is reserved for scoring. Every system receives the same request, history, and output requirements.

All benchmark prompts use Simplified Chinese; cross-lingual transfer is outside the present evaluation. Dataset provenance, preprocessing, templates, and scoring rules are provided in the supplementary material.

We report task-specific metrics before computing aggregate scores. The five primary point-error metrics are sMAPE, MASE, WAPE, RMSE, and MAE; directional and peak-oriented measures serve as additional forecasting diagnostics. Uncertainty is evaluated with interval coverage, calibration error, quantile-based criteria, and the continuous ranked probability score (CRPS). Temporal reasoning and instruction following are scored using task correctness, request interpretation, format validity, and required-field completeness. Efficiency is measured by latency, memory, throughput, and token consumption. Missing required outputs count as failures rather than being imputed. For the paired sMAPE comparison in the main results, we form a 95% confidence interval from 1,000 dimension-level bootstrap replicates.

For a bounded higher-is-better metric $m \in [0, 1]$, the score is $100m$. A lower-is-better metric is scored with the clipped linear map $100 \max(0, 1 - m/b_m)$; an unbounded higher-is-better metric uses $100 \min(1, m/b_m)$. A reference bound b_m is fixed for each metric before systems are compared. Boolean criteria receive 100 for success and 0 for failure. Each dimension score combines its applicable sub-metrics using fixed weights. The supplementary protocol lists every raw metric, reference bound, intra-dimension weight, direction of preference, and group assignment.

After mapping each evaluation dimension to $[0, 100]$, we average the dimension scores within each group without additional weighting. Let S_F , S_R , S_I , S_U , and S_E denote the resulting forecasting, temporal-reasoning, instruction-

following, uncertainty, and efficiency group scores. The Task-Level Score (TLS) is

$$\text{TLS} = 0.60S_F + 0.15S_R + 0.15S_I + 0.05S_U + 0.05S_E. \quad (6)$$

We fix the weights in Eq. (6) before comparing systems. Numerical forecasting receives 60% because it is the benchmark’s central task. Temporal reasoning and instruction following each receive 15%; uncertainty communication and efficiency each receive 5%. TLS is the primary system-level metric, while raw metrics and group scores retain capability-level detail. A higher TLS does not indicate better performance in every group. In the sensitivity analysis, we vary the forecasting weight from 0.40 to 0.80 and allocate the remainder to reasoning, instruction following, uncertainty, and efficiency in the fixed ratio 3:3:1:1. Setting $w_F = 0.60$ recovers the primary weights.

Our implementation pairs a LoRA-adapted Qwen3.5-9B language component (Qwen Team 2026) with a frozen Chronos-2 forecaster. The comparisons include eleven proprietary direct LLMs, three open TSFMs (TimesFM, Moirai, and MOMENT), fixed-context controls, and component ablations. Each direct LLM receives the same request and history and must return the requested forecast or forecast-related response without access to Chronos-2 or the evidence module. API baselines use temperature 0 where supported, a maximum output budget of 8,192 tokens, and a 300-second timeout.

Five fixed-context controls retain the language and specialist components but replace the proposed context length with a fixed global $C \in \{64, 128, 256, 512, 1024\}$. The *w/o LoRA* variant keeps the two-path framework and substitutes the unadapted language model. The *w/o Chronos-2* variant removes the specialist and forecasts directly with the language model. The *C2-only* variant runs Chronos-2 with $C = 512$ but omits language-guided routing, configuration, and response generation. These are comparisons between system variants, not estimates of additive component effects.

All systems are evaluated on the same records, with future targets reserved for scoring. The same parsers and scoring rules process every system’s output. Model identifiers, prompts, inference settings, and access dates are listed in the supplementary material. Each system is evaluated three times on the fixed manifest. The bootstrap interval reported below resamples dimensions rather than repeated runs.

Reproducibility

Training and local inference use an NVIDIA A800. The artifact contains code, the fixed evaluation manifest, and reference configurations for adaptation and reported variants. The supplement specifies hardware/software details, data provenance and splits, prompts, transformations, output schemas, and evidence-verification rules.

Main Results

ChronoScribe achieves the lowest error on each of the five primary point-error metrics (sMAPE, MASE, WAPE, RMSE, and MAE) among all evaluated systems, including the fixed-context controls. Table 2 reports the raw errors.

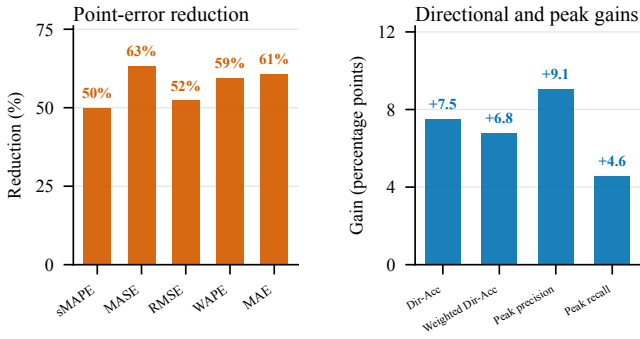


Figure 4: Per-metric gains over the best direct LLM.

Table 1: System scores; column maxima are bold.

Model	TLS	F	R	I	U	E
ChronoScribe	65.5	73.7	36.2	59.0	71.2	69.7
<i>Direct LLMs</i>						
Gemini-3.1 Flash	55.0	51.8	52.1	53.8	75.7	84.9
GPT-5.5	51.7	51.5	43.5	49.8	73.6	62.3
GPT-5.6-Sol	50.9	49.1	45.5	49.3	73.5	71.3
DeepSeek-V4 Flash	50.3	46.0	43.5	65.8	65.7	61.1
Grok-4.3	50.3	49.4	40.2	49.7	73.5	69.2
GPT-5.4	50.3	46.5	43.3	53.7	74.9	81.2
Claude Sonnet-5	50.0	47.6	46.1	48.2	70.9	75.8
MiniMax-M3	50.0	51.8	40.2	44.7	62.5	61.6
GLM-5.2	49.3	47.9	47.8	48.6	69.6	51.9
DeepSeek-V4 Pro	47.3	40.0	44.4	64.7	68.9	70.1
Claude Opus-4.8	46.2	41.1	40.3	50.9	72.3	84.6
<i>TSFMs</i>						
TimesFM	49.1	44.3	26.8	63.9	82.9	95.1
Moirai	47.3	41.2	26.9	64.1	82.1	96.7
MOMENT	47.2	41.1	26.8	64.1	80.3	96.8

Across these metrics, the lowest direct-LLM error remains $2.0\text{--}2.7\times$ the corresponding ChronoScribe error. This range represents relative reductions of approximately 50–63%. Figure 4 shows these gains. Relative to GPT-5.5 (OpenAI 2026), the best direct-LLM baseline on aggregate sMAPE, ChronoScribe lowers the metric from 50.85% to 25.51%. The 95% dimension-level bootstrap CI for this reduction is $[17.4, 35.2]$ percentage points.

Using unrounded scores, ChronoScribe attains a TLS of 65.5, 10.6 points above Gemini-3.1-Flash (Google 2026) (TLS 55.0). Table 1 reports the system scores; Figure 5 profiles the five groups.

The raw error ratios and the TLS margin summarize performance on different scales. Point-error ratios are computed directly from the original metrics. TLS instead maps heterogeneous metrics to $[0, 100]$ using fixed reference bounds, averages dimension scores within each group, and applies the predefined 60/15/15/5/5 weights. This process converts multiplicative forecasting-error gaps into bounded additive score differences, which are further attenuated by the 0.60 forecasting weight. A 10.6-point TLS lead is therefore compatible with the $2.0\text{--}2.7\times$ raw error ratios: it summarizes the complete workflow rather than forecasting accuracy alone.

ChronoScribe ranks first at every tested forecasting weight

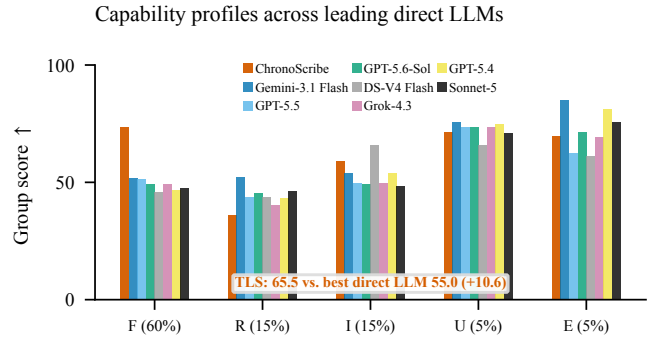


Figure 5: Group scores: ChronoScribe vs. direct LLMs.

Table 2: Point-forecast errors; column minima are bold.

Model	sMAPE	MASE	RMSE	WAPE	MAE
ChronoScribe	25.5	4.20	1.77	24.4	1.13
<i>Direct LLMs</i>					
Gemini-3.1 Flash	62.4	24.23	6.14	102.1	4.36
GPT-5.5	50.9	14.81	3.70	60.3	2.87
GPT-5.6-Sol	57.3	14.35	3.93	68.0	3.51
DeepSeek-V4 Flash	62.7	13.80	4.46	70.3	3.62
Grok-4.3	64.2	16.28	5.14	82.9	3.77
GPT-5.4	63.4	15.68	4.13	74.9	3.48
Claude Sonnet-5	65.0	15.01	4.53	72.6	3.80
MiniMax-M3	57.5	13.20	4.62	60.6	3.52
GLM-5.2	62.1	11.43	4.38	66.2	3.51
DeepSeek-V4 Pro	66.7	15.29	4.28	73.2	3.68
Claude Opus-4.8	66.2	16.54	5.86	92.7	6.00
<i>TSFMs</i>					
TimesFM	69.7	16.12	8.54	105.0	7.59
Moirai	78.0	16.78	9.16	113.1	8.12
MOMENT	77.2	16.70	8.88	116.0	7.85

from 0.40 to 0.80. Figure 6 shows normalized scores across 23 dimensions; darker cells indicate higher scores. Its margin over the runner-up grows from 4.9 points at $w_F = 0.40$ to 10.6 points at the primary weight of 0.60 and 15.5 points at 0.80.

Analysis

TLS falls from 65.5 to 51.7 without LoRA adaptation and to 25.5 without Chronos-2. Figure 7 shows the component ablations. The LoRA ablation changes both the directive and response interfaces, so its effect cannot be attributed to context selection alone. Removing Chronos-2 requires the language model to forecast directly; the resulting drop shows that direct language-model forecasting is not an adequate substitute for the numerical specialist in this setting. C2-only reaches a TLS of 35.3. Because it omits language-side routing and communication, we treat it as an incomplete-workflow control rather than an estimate of the controller’s additive contribution.

The controller returns a parseable directive with the protocol-defined route for 95.9% of requests. On forecasting requests, its horizon field matches the supplied horizon in every case. Because the horizon is fixed by the task, this result measures constraint compliance rather than autonomous

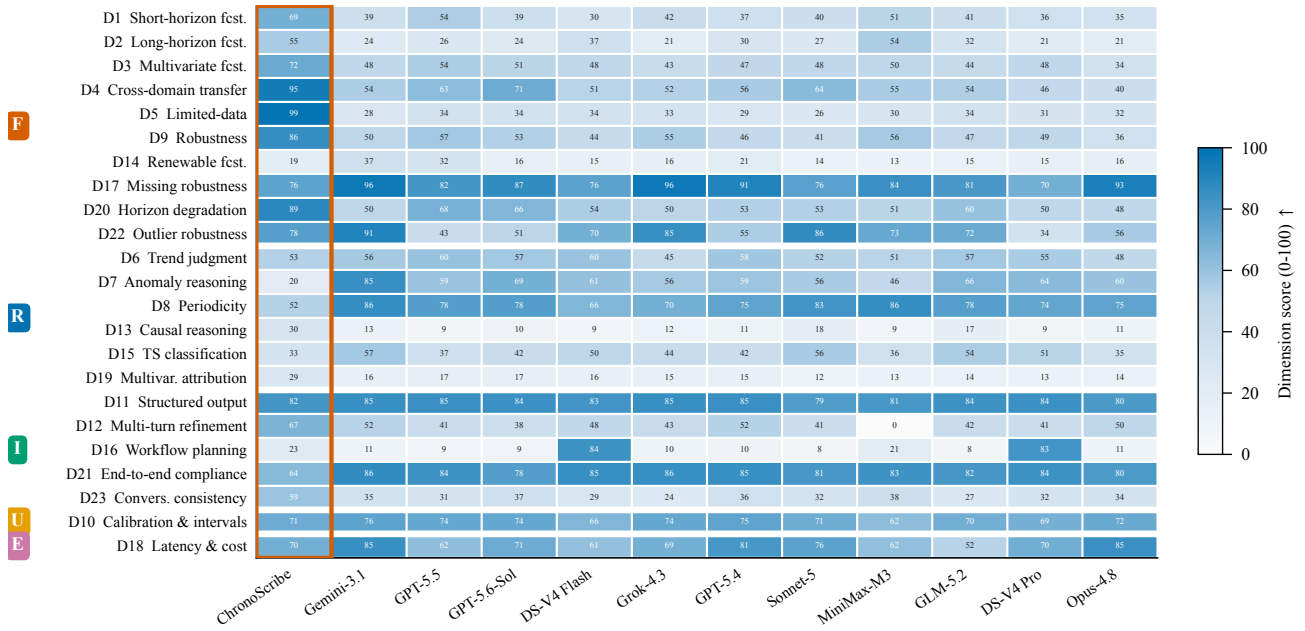


Figure 6: Normalized scores across 23 dimensions.

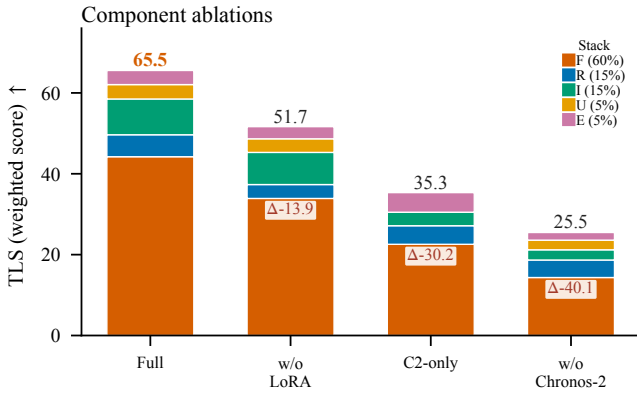


Figure 7: TLS decomposition under component ablations.

horizon inference.

We also audit 100 randomly sampled specialist-path responses using a rule-based verifier. Across 230 extracted claims, agreement with deterministic forecast evidence is 87.4%, and 89 responses pass every applicable check. Agreement ranges from 79.6% for numerical values under a 15% relative-error tolerance to 92.1% for directional claims. This audit measures evidence traceability, not forecast accuracy or calibration; the supplementary material provides the complete verification rules.

Discussion and Limitations

ChronoScribe makes numerical provenance auditable, but the specialist path adds orchestration and an extra LLM call; latency depends on horizon, response length, and backend. We evaluate one LLM–forecaster pairing, Simplified

Chinese prompts, and benchmark instruction families, limiting generalization. Although adaptation and evaluation data are disjoint, Chronos-2 was pretrained on public and synthetic corpora, so exposure to individual evaluation series cannot be ruled out (Ansari et al. 2025). Evidence conditioning improves traceability but cannot guarantee grounded decoding. Lower numerical-claim agreement motivates constrained rendering or verifier-based revision for high-stakes use.

Conclusion

ChronoScribe applies semantic–numerical specialization to instruction-driven time series forecasting. An adapted LLM interprets requests and proposes structured directives, which deterministic rules convert to executable settings. The frozen TSFM alone produces numerical trajectories; deterministic summaries ground the final response. Across 23 dimensions and 30 public datasets, ChronoScribe achieves the lowest error on all five primary point-error metrics, including fixed-context controls, while the best direct LLM remains 2.0–2.7× worse on each. Its TLS of 65.5 exceeds the strongest of eleven proprietary direct-LLM baselines by 10.6 points. Routing and grounding audits indicate that the directive and response interfaces perform reliably in most cases. The design retains a single source for forecast values while using the LLM for interpretation, configuration, and communication.

References

- Abdulaal, A.; Liu, Z.; and Lancewicki, T. 2021. Practical Approach to Asynchronous Multivariate Time Series Anomaly Detection and Localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2485–2494.
- Aksu, T.; Woo, G.; Liu, J.; Liu, X.; Liu, C.; Savarese, S.; Xiong, C.; and Sahoo, D. 2024. GIFT-Eval: A Benchmark for General Time Series Forecasting Model Evaluation. *arXiv preprint arXiv:2410.10393*.
- Ansari, A. F.; Shchur, O.; Küken, J.; Auer, A.; Han, B.; Mercado, P.; Rangapuram, S. S.; Shen, H.; Stella, L.; Zhang, X.; Goswami, M.; Kapoor, S.; Maddix, D. C.; Gueron, P.; Hu, T.; Yin, J.; Erickson, N.; Desai, P. M.; Wang, H.; Rangwala, H.; Karypis, G.; Wang, Y.; and Bohlke-Schneider, M. 2025. Chronos-2: From Univariate to Universal Forecasting. *arXiv preprint arXiv:2510.15821*.
- Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Pineda Arango, S.; Kapoor, S.; Zschiegner, J.; Maddix, D. C.; Wang, H.; Mahoney, M. W.; Torkkola, K.; Wilson, A. G.; Bohlke-Schneider, M.; and Wang, Y. 2024. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*.
- Bagnall, A.; Dau, H. A.; Lines, J.; Flynn, M.; Large, J.; Bostrom, A.; Southam, P.; and Keogh, E. 2018. The UEA Multivariate Time Series Classification Archive, 2018. *arXiv preprint arXiv:1811.00075*.
- Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2024. TEMPO: Prompt-Based Generative Pre-Trained Transformer for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.
- Chowdhury, A. M.; Akter, R.; and Arib, S. H. 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, 20597–20605.
- Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A Decoder-Only Foundation Model for Time-Series Forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 10148–10167.
- Dau, H. A.; Bagnall, A.; Kamgar, K.; Yeh, C.-C. M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C. A.; and Keogh, E. 2019. The UCR Time Series Archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6): 1293–1305.
- Dempster, A.; Foumani, N. M.; Tan, C. W.; Miller, L.; Mishra, A.; Salehi, M.; Pelletier, C.; Schmidt, D. F.; and Webb, G. I. 2025. MONSTER: Monash Scalable Time Series Evaluation Repository. *Transactions on Machine Learning Research*.
- Gneiting, T.; and Raftery, A. E. 2007. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477): 359–378.
- Google. 2026. Gemini API Release Notes. <https://ai.google.dev/gemini-api/docs/changelog>. Accessed: 2026-07-20.
- Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. MOMENT: A Family of Open Time-Series Foundation Models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 16115–16152.
- Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large Language Models Are Zero-Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.
- Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; and Soderstrom, T. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–395.
- Hyndman, R. J.; and Koehler, A. B. 2006. Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, 22(4): 679–688.
- Jia, S.; Song, B.; Ye, C.; and Yuan, C. 2026. M3Time: LLM-Enhanced Multi-Modal, Multi-Scale, and Multi-Frequency Multivariate Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 22265–22273.
- Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; and Wen, Q. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations (ICLR)*.
- Laptev, N.; Amizadeh, S.; and Billawala, Y. 2015. S5: A Labeled Anomaly Detection Dataset, Version 1.0 (16M). Yahoo Research Webscope.
- Lavin, A.; and Ahmad, S. 2015. Evaluating Real-Time Anomaly Detection Algorithms—The Numenta Anomaly Benchmark. In *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications*, 38–44.
- Liao, Y.; Wang, Z.; Nie, Q.; and Zhang, Z. 2026. Bridging the Last Mile of Time Series Forecasting with LLM Agents. *arXiv preprint arXiv:2606.02497*.
- Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025a. TimeCMA: Towards LLM-Empowered Multivariate Time Series Forecasting via Cross-Modality Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18780–18788.
- Liu, P.; Guo, H.; Dai, T.; Li, N.; Bao, J.; Ren, X.; Jiang, Y.; and Xia, S.-T. 2025b. CALF: Aligning LLMs for Time Series Forecasting via Cross-Modal Fine-Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18915–18923.
- Liu, X.; Hu, J.; Li, Y.; Diao, S.; Liang, Y.; Hooi, B.; and Zimmermann, R. 2024a. UniTime: A Language-Empowered Unified Model for Cross-Domain Time Series Forecasting. In *Proceedings of the ACM Web Conference (WWW)*, 4095–4106.

- Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024b. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.
- Liu, Y.; Zhang, H.; Li, C.; Huang, X.; Wang, J.; and Long, M. 2024c. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 32369–32399.
- Makridakis, S.; Spiliotis, E.; and Assimakopoulos, V. 2020. The M4 Competition: 100,000 Time Series and 61 Forecasting Methods. *International Journal of Forecasting*, 36(1): 54–74.
- Makridakis, S.; Spiliotis, E.; and Assimakopoulos, V. 2022. M5 Accuracy Competition: Results, Findings, and Conclusions. *International Journal of Forecasting*, 38(4): 1346–1364.
- Mathur, A. P.; and Tippenhauer, N. O. 2016. SWaT: A Water Treatment Testbed for Research and Training on ICS Security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks*, 31–36.
- Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*.
- OpenAI. 2026. GPT-5.5 Model. <https://developers.openai.com/api/docs/models/gpt-5.5>. Accessed: 2026-07-20.
- Qwen Team. 2026. Qwen3.5: Towards Native Multimodal Agents. <https://qwen.ai/blog?id=qwen3.5>. Accessed: 2026-07-20.
- Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.
- Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; and Pei, D. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2828–2837.
- Wang, C.; Qi, Q.; Rao, Z.; Pan, L.; Wang, J.; and Liao, J. 2025. ChronoSteer: Bridging Large Language Model and Time Series Foundation Model via Synthetic Data. *arXiv preprint arXiv:2505.10083*.
- Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 53140–53164.
- Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations (ICLR)*.
- Wu, L.; Wang, X.; Ge, C.; Chu, R.; and Wang, L. 2026. Exo-Timer: Leveraging Large Language Models for Time Series Forecasting with Exogenous Variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 26956–26964.
- Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.
- Zhao, H.; Zhang, X.; Wei, J.; Xu, Y.; He, Y.; Sun, S.; and You, C. 2025. TimeSeriesScientist: A General-Purpose AI Agent for Time Series Analysis. *arXiv preprint arXiv:2510.01538*.
- Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.
- Zhou, T.; Niu, P.; Wang, X.; Sun, L.; and Jin, R. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.