TSORCHESTRA: TIME SERIES AGENTIC ORCHESTRATION FRAMEWORK TOWARDS DYNAMIC AND FAITHFUL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series are inherently dynamic and diverse, meaning a single foundation model can seldom maintain universal dominance across varying regimes. Consequently, the frontier of forecasting has shifted from seeking a single optimal architecture to dynamic model orchestration. While Large Language Models (LLMs) offer powerful reasoning capabilities, their direct application to forecasting remains ineffective. We propose TSOrchestra, a novel framework that bridges mathematical execution with agentic reasoning by repositioning the LLM as a metacognitive auditor. While the underlying numerical weights are determined by a continuous optimization solver, the LLM auditor actively steers the optimization trajectory in real-time through iterative hypothesis testing and dynamic metric routing. To instill domain-specific heuristics, we introduce an R1-style reinforcement learning pipeline. The agent's policy is regularized by structural faithfulness rewards, ensuring its audit trails are strictly anchored to empirical, decomposition-based temporal dynamics. Validated on the GIFT-Eval benchmark across 23 datasets and 97 settings, TSOrchestra *consistently* outperforms all leading time-series foundation models on both CRPS and MASE metrics, establishing new state-of-the-art results.

1 INTRODUCTION

With demonstrated strengths in reasoning (Wei et al., 2022; Højer et al., 2025), decision-making (Schmied et al., 2025), and contextual understanding (Berglund et al., 2024), LLMs are natural candidates for time series forecasting beyond conventional regression (Fan et al., 2025). In parallel, the forecasting landscape has rapidly diversified, with the emergence of Transformers (Wen et al., 2022), N-BEATS (Oreshkin et al., 2019), and large-scale foundation models (Das et al., 2024; Faw et al., 2025a). Despite these advances, no single model consistently dominates across the heterogeneous and non-stationary regimes encountered in real-world data. Recent work has further argued that time series forecasting is inherently reasoning-centric, requiring structured interpretation of trends, seasonality, and regime shifts (Luo et al., 2025a). Under the dual requirement of adapting to regime diversity while maintaining interpretability, the critical question shifts from which model performs best to determining *how multiple specialized models should be dynamically coordinated and justified.*

When directly applied to time series forecasting, LLMs exhibit well-documented limitations (Li & Pedersen, 2025; Merrill et al., 2024; Tang et al., 2025b; Tan et al., 2024). Although they can reason about temporal patterns at a conceptual level, their pretraining objective and output space make them ill-suited for precise numerical prediction. Prior efforts to bridge this gap have largely focused on forcing LLMs into predictive roles: direct regression-based methods incur substantial computational cost while remaining inferior to specialized models (Bumb et al., 2025; Cao et al., 2024a; Jin et al., 2024); multimodal alignment approaches reduce LLMs to embedding generators, underutilizing their reasoning capacity (Xie et al., 2024; Zheng et al., 2025); and supervised or reinforcement-based finetuning often yields marginal gains despite extensive training (Liu et al., 2025d; Wang et al., 2025b). Agent-based zero-shot approaches preserve general reasoning ability but typically lack the domain-specific understanding required to assess model suitability under evolving temporal regimes (Gruver et al., 2023a; Ye et al., 2024). Collectively, these limitations suggest that the value of LLMs in time series forecasting does not lie in direct prediction, but in *meta-level*

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

reasoning, which analyzes the outputs and trajectories produced by foundation models to perform high-level strategic decisions.

The absence of a universally optimal forecasting model implies that model orchestration is not a design choice but a necessity in real-world time series forecasting. However, historical "best" ensemble configurations do not guarantee future performance; as data regimes evolve, static ensemble weights quickly become outdated and unreliable. Effective forecasting, therefore, requires *dynamic orchestration* that can adapt to local regime characteristics rather than relying on static model preferences. We ground this challenge via the Temporal Incompatibility index, a conceptual metric designed to quantify the divergence between historical patterns and future outcomes. Given that the index is inherently transient and non-stationary, it is navigated in practice through the agent's iterative hypothesis testing, which serves as an empirical proxy to sense and re-route model allocations amid latent regime shifts.

Our framework, TSOrchestra, acknowledges this temporal dependency by repositioning the LLM as a *metacognitive judge*. The agent performs continuous, reasoning-guided auditing to track these latent I_T fluctuations and ensure that the ensemble remains "live" by updating the weight for each sliding window. Because raw LLMs lack an inherent understanding of the relationship between abstract numerical weights and underlying temporal dynamics, we employ R1-style finetuning (Guo et al., 2025) to train the agent to interpret ensemble weights as contextualized hypotheses about local temporal mechanisms rather than static model properties. To ensure stability and decomposition-aligned grounding, TSOrchestra further incorporates SHAP-based faithfulness scores, aligning the agent's reasoning with the true performance drivers of each model. We evaluated our approach on the GIFT-Eval (Aksu et al., 2024) benchmark, which comprises 23 time series datasets with 97 settings to ensure generalizability. Our model significantly outperformed the best time series foundation models on both CRPS and MASE metrics, establishing new state-of-the-art results across diverse forecasting tasks. Our contributions are summarized as follows:

- **A Paradigm Shift to Metacognitive Orchestration**: We move beyond static model selection by introducing TSOrchestra, a framework that repositions the LLM as a metacognitive auditor. By decoupling high-level reasoning from numerical execution, the agent iteratively steers continuous optimization solvers through hypothesis testing and dynamic routing, effectively navigating complex regime shifts.
- **R1-Style Alignment via Structural Faithfulness**: We design a specialized reinforcement learning pipeline (SFT + GRPO) to instill domain-specific heuristics into the agent. Central to this process is a structural faithfulness reward based on STL-SHAP decomposition, which enforces a strict alignment between the agent's reasoning trails and verifiable temporal dynamics, thereby eliminating hallucinated rationales.
- **State-of-the-Art Performance**: On the GIFT-Eval benchmark, TSOrchestra outperforms leading time-series foundation models on both CRPS and MASE. These results show that the same orchestration framework transfers across datasets without being tied to a single backbone.

2 Related Work

LLM/Agent-Based Time Series Forecasting: Recent approaches leverage LLMs for forecasting through various paradigms, yet many still place LLMs in roles that are not fully aligned with their strengths in high-level reasoning and decision auditing. Direct prediction methods force LLMs to generate numerical outputs: LLMTIME (Gruver et al., 2023b) treats forecasting as next-token generation, TEMPO (Cao et al., 2024b) decomposes series for component prediction, DP-GPT4MTS (Liu et al., 2025a) uses dual prompting, and NNCL-TLLM (Bogahawatte et al., 2024) learns time-series-compatible prototypes. Context-enhanced and multimodal approaches incorporate external information or modality transformations: CiK (Williams et al., 2025) shows gains with textual context but catastrophic failures in direct prediction, Hybrid-MMF (Kim et al., 2024) jointly forecasts numbers and narratives with negative fusion results, CMLLM (Zhu et al., 2025) converts SCADA signals to text, Tang et al. (2025a) reprograms histories into prose, and Time-VLM (Zhong et al., 2025) leverages vision-language models to bridge temporal, visual, and textual modalities for augmented forecasting. Retrieval-augmented methods enhance predictions with historical patterns: RAF (Tire et al., 2024) builds dataset-specific databases, TimeRAG (Yang et al., 2024) slices and reprograms segments,

TimeReasoner (Wang et al., 2025d) implements structured reasoning, and Time-R1 (Luo et al., 2025b) aligns scripted chains with supervised learning. Multi-agent and LLM-in-the-loop systems introduce iterative reasoning(Yeh et al., 2025): Zhang et al. (2025) uses competitive hypothesis pruning, NewsForecast (Wang et al., 2024) employs critique-revise loops, TimeXL (Jiang et al., 2025) uses prediction-reflection-refinement with multiple LLMs for explainable multimodal prediction, CoLLM (Wang et al., 2025c) routes between models using confidence scores.

Time Series Foundation Models: Recent foundation models have achieved impressive scale and zero-shot capabilities, yet remain fundamentally limited as black-box predictors (Liu et al., 2024; Ansari et al., 2024). TimesFM (Faw et al., 2025b) operates on patches of time series data using a Transformer architecture with continued pre-training for zero-shot and few-shot forecasting. Toto (Cohen et al., 2026), trained by Datadog on nearly a trillion data points from observability data, demonstrates strong zero-shot prediction on unseen series. TabPFN-TS (Hoo et al., 2025), despite having only 11 million parameters, is trained purely on synthetic tabular data and achieves competitive forecasting performance. Time-MoE (Shi et al., 2025) scales time-series pretraining with sparse mixture-of-experts to increase model capacity while reducing activated computation. Sundial (Liu et al., 2025c) scales to a trillion time points and introduces a TimeFlow loss for flexible probabilistic forecasting. Timer casts forecasting as generative sequence modeling to leverage LLM capabilities, and the more recent Timer-S1 (Liu et al., 2026) further explores billion-scale serial scaling with mixture-of-experts and long-context forecasting. Recent benchmark work such as TIME (Qiao et al., 2026) also highlights the growing need for stricter zero-shot evaluation and pattern-level analysis of TSFMs.

LLMs for General Time Series Tasks: LLMs have been applied to various time series tasks beyond forecasting. For classification, HiTime (Tao et al., 2024) aligns time series with textual semantics to generate class labels, while FinSrag (Xiao et al., 2025) predicts stock movements through retrieval and prompting. For anomaly detection, Zhou & Yu (2025) finds that image inputs often outperform text for identifying anomalies, and the large-model-based smart agent SLEP (Wang et al., 2026) targets anomaly detection in power systems by combining large-model prompting with domain-specific time-series detection workflows. MedTsLLM (Chan et al., 2024) handles segmentation by projecting LLM outputs to masks and boundaries. ChatTime (Wang et al., 2025a) unifies multiple tasks by expanding tokenizers with discretized value symbols. From a systems perspective, TimeCopilot (Garza & Rosillo, 2025) automates model selection and ensembling. More recent work also explores broader time-series understanding and interaction: Sonar-TS (Tan et al., 2026) studies natural-language querying over time-series databases through a search-then-verify pipeline, while TimeOmni-VL (Guan et al., 2026) unifies time-series understanding and generation through vision-centric multimodal modeling. These diverse applications demonstrate LLMs' versatility in time series analysis; yet, they typically treat the model as a direct executor or task-specific component. Recent work has introduced agentic architectures that incorporate iterative reasoning and tool usage, such as TimeCAP (Lee et al., 2025) and TS-Agent (Liu et al., 2025b).

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Real-world time series rarely maintain stationary distributions due to evolving seasonal patterns, economic regime shifts, and structural breaks. This inherent temporal heterogeneity poses a fundamental challenge where no single monolithic model can optimally capture all regime-specific dynamics. By orchestrating specialized models, the system can approximate complex multi-regime dynamics that individual models fail to represent. Given historical time series $\mathbf{y}_{1:t} \in \mathbb{R}^t$ and M candidate forecasting foundation models $\mathcal{M} = \{m_1, m_2, ..., m_M\}$, our system produces both predictions and explanations through:

$$\hat{\mathbf{y}}_{t+1:t+h}, \mathcal{A} = f(\mathbf{y}_{1:t}, \mathcal{M}, \theta_{opt}, \theta_{LLM}) \tag{1}$$

where $\hat{\mathbf{y}}_{t+1:t+h} \in \mathbb{R}^h$ is the forecasting results for horizon h, $\mathcal{A}$ represents the audit trail, encapsulating the optimization and reasoning history, θ_{opt} represents the optimizer that determines weights $\mathbf{w}^*$, θ_{LLM} represents the metacognitive reasoning agent. This agent acts as an intelligent judge, evaluating the optimization trajectory to decide whether to accept current results or continue refinement, while generating reflective explanations for the final configuration.

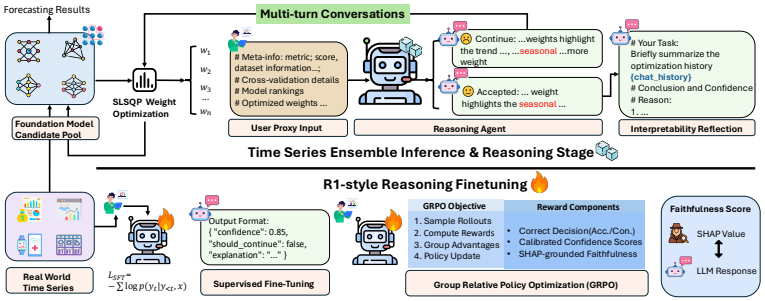


Figure 1: Overview of the TSOrchestra Framework. (Top) Metacognitive Inference: An LLM auditor actively steers mathematical weight optimization, utilizing multi-turn reasoning and hypothesis testing to navigate regime shifts. (Bottom) Reasoning Alignment: The agent's policy is cultivated via an R1-style reinforcement learning pipeline, ensuring the agent's audit trails are strictly anchored to underlying temporal dynamics.

3.2 Weight Optimization

Motivation - Ensemble Learning under Temporal Incompatibility. As discussed in Section 3.1, accurate forecasting requires maintaining multiple predictive hypotheses simultaneously, which naturally motivates ensemble learning. We formalize this requirement through the *Temporal Incompatibility Index*, $I_T(\mathcal{D})$ (Eq. 12 in Section E), which quantifies the extent to which historical similarity necessitates diversified model contributions to account for divergent future outcomes:

Theorem 3.1 (Ensemble Superiority for Time Series Forecasting)**.** *For a time series dataset $\mathcal{D}$ with temporal incompatibility $I_T(\mathcal{D}) > 0$ and a pool of M diverse forecasting models, the optimal ensemble satisfies:*

$$L^*_{ensemble}(\mathcal{D}) \;\leq\; L^*_{monolithic}(\mathcal{D}) \;-\; \Omega(I_T(\mathcal{D}), M), \tag{2}$$

where $\Omega(I_T, M) = \frac{I_T \cdot \log M}{1+\exp(-\kappa \cdot I_T)}$ characterizes the ensemble advantage.

Theorem 3.1 establishes that ensemble methods enjoy a provable advantage whenever temporal incompatibility is present, and that this advantage grows with both the degree of incompatibility and the diversity of constituent models. I_T is inherently time-varying, ensemble configurations must adapt dynamically to maintain this theoretical advantage as data regimes shift. We provide the formal definition of $I_T(\mathcal{D})$, assumptions, and proof sketch in Section E; throughout the main method, I_T is used as a conceptual lens rather than as a directly estimated training-time statistic.

Static Weight Optimization within an Observation Window. In practice, ensemble weights are estimated over a finite observation window by constructing a validation set from historical data and minimizing predictive loss. Given forecasts $\mathbf{X} \in \mathbb{R}^{N \times M}$ from M candidate models and ground-truth targets $\mathbf{y} \in \mathbb{R}^N$, we compute window-specific base weights using Sequential Least Squares Programming (SLSQP) (Joshy & Hwang, 2024):

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{y}, \mathbf{X}\mathbf{w}) \quad \text{s.t.} \quad \mathbf{w} \geq 0,\ \sum_{i=1}^{M} w_i = 1. \tag{3}$$

This optimization yields weights that are locally optimal for the temporal context captured by the current observation window, implicitly approximating the ensemble configuration predicted by Theorem 3.1 under the incompatibility level observed within that window.

Dynamic Incompatibility and Sliding-Window Approximation. Crucially, temporal incompatibility is not static: the relative relevance of specialized models varies as regimes evolve. As a result, the optimal ensemble configuration implied by Theorem 3.1 becomes *window-dependent*. To track this evolution, we adopt a sliding-window optimization strategy, which acts as an empirical proxy to adapt to underlying I_T shifts without requiring explicit mathematical derivations, repeatedly solving Eq. 3 over successive observation windows. This produces a sequence of locally optimal weight vectors that dynamically respond to changes in temporal incompatibility, offering a stable and tractable approximation to fully dynamic ensemble weighting.

Compared to global dynamic weighting schemes, sliding-window optimization avoids excessive sensitivity to noise while preserving responsiveness to regime shifts. However, it remains inherently

retrospective: weights are optimized to explain past observations and may lag behind future regime transitions when incompatibility changes rapidly.

Agent-Guided Forward-Looking Weight Refinement. Sliding-window optimization cannot determine whether the current locally optimal weights will generalize beyond the observed window. To address this limitation, we introduce an LLM-based agent that performs forward-looking assessment of ensemble weight validity. Rather than replacing numerical optimization, the agent reasons over the accumulated audit trail $\mathcal{A}$, including historical weight trajectories, validation performance, and optimization stability, to infer the future evolution of temporal incompatibility. Based on this reasoning, the agent decides whether the current window-optimal weights should be accepted or if the optimization should explore a different hypothesis. Rather than outputting continuous weight adjustments, the agent's action space is defined as a discrete routing decision: $\mathcal{D}_k \in \{\text{Accept}, \text{Continue with } L_{next}\}$, where L_{next} represents the recommended evaluation metric for the next optimization round to better capture the anticipated regime shift.

3.3 TSORCHESTRA AGENT SYSTEM: THE REASONING AGENT AS METACOGNITIVE AUDITOR

We formulate the weight optimization as a reasoning problem where an LLM agent acts as a metacognitive auditor over the mathematical optimization process. Unlike traditional approaches that blindly accept cross-validation results, our agent reasons about the gap between historical validation windows and future forecasting regimes.

The agent orchestrates iterative refinement through two capabilities: (i) strategic exploration by iteratively selecting metrics $L_k \in \mathcal{L}$ to test specific hypotheses about weight quality from optimizing Eq. 3 , (ii) evaluative reasoning that synthesizes evidence into confidence scores, decision rationales, and next hypotheses:

$$(\mathcal{C}_k, \mathcal{D}_k, L_{k+1}) = \text{LLM}_{\text{eval}}\{\text{Align}(\mathbf{w}_k^*, \mathcal{P}_{cv}), \text{Match}(\mathbf{w}_k^*, \mathcal{F}_{data}), \text{Future}(\mathbf{w}_k^*, \mathcal{K}_{domain})\} \tag{4}$$

where $\mathcal{C}_k \in [0, 1]$ is the confidence score, $\mathcal{D}_k \in \{\text{continue}, \text{accept}\}$ is the decision for the current round of optimization, $L_{k+1} \in \mathcal{L}$ is the recommended next metric, $\mathcal{P}_{cv}$ represents cross-validation performance across all metrics in $\mathcal{L}$, $\mathcal{F}_{data}$ contains dataset characteristics, and $\mathcal{K}_{domain}$ encodes domain knowledge such as model capabilities. Concrete examples of this multi-turn metric-routing process are shown in Sections B and I. The LLM evaluates weights across three dimensions: *Align* ensures performance-weight coherence and mathematical soundness, *Match* verifies dataset-model compatibility and complementarity, and *Future* assesses generalization risks and ensemble robustness.

The agent engages in multi-turn conversations with a user proxy to thoroughly evaluate each weight configuration. This dialogue structure allows the agent to: (1) articulate initial hypotheses about weight quality, (2) respond to challenges about potential weaknesses, and (3) refine its confidence assessment through iterative questioning. The final weight selection occurs when the LLM determines with high confidence that further optimization offers diminishing returns: $\mathbf{w}^* = \arg\max_{\mathbf{w} \in \mathcal{W}_{explored}} \mathcal{C}_{LLM}(\mathbf{w})$, where $\mathcal{C}_{LLM}$ represents the LLM's learned confidence function. This transforms optimization from a black-box numerical process into an interpretable reasoning task where the LLM can explain why specific weights represent the optimal trade-off between performance and robustness.

3.4 R1-STYLE FINE-TUNING FOR REASONING

While LLMs can analyze optimization trajectories and reason about future temporal incompatibility evolution, they lack the domain-specific intuitions critical for ensemble weight selection. Time series optimization demands recognizing numerical convergence patterns, calibrating stopping decisions against overfitting risks, and understanding how validation performance translates to future forecasting accuracy—capabilities absent from language pretraining. Raw LLMs exhibit problematic behaviors: terminating prematurely on superficial improvements, continuing indefinitely without recognizing diminishing returns, or failing to distinguish genuine robustness from validation artifacts. We address these gaps through a two-stage training approach that transforms a general-purpose LLM into a specialized ensemble-decider: supervised fine-tuning (SFT) teaches structured decision protocols through expert-annotated trajectories, while reinforcement learning (GRPO) refines these capabilities by directly optimizing for held-out forecasting performance, developing the mathematical intuitions

that bridge linguistic reasoning and numerical optimization expertise. Full data-generation and training details are provided in Section G.

3.4.1 SUPERVISED FINE-TUNING (SFT)

While LLMs can analyze patterns, they lack inherent understanding of optimization dynamics—when marginal improvements justify continuation versus when convergence indicates termination. Simply prompting LLMs yields inconsistent decisions: they may terminate prematurely at local improvements or continue indefinitely chasing negligible gains. We need to teach the model what constitutes meaningful progress in ensemble optimization and how to recognize genuine convergence patterns. Real optimization trajectories alone provide insufficient coverage of decision boundaries. Most trajectories contain obvious decisions (clear improvement or convergence), leaving the model undertrained on marginal cases where $\Delta = L_{\text{current}} - L_{\text{best}}$ approaches zero. We therefore construct a mixed dataset combining real trajectories with synthetically augmented boundary cases, ensuring the model learns both authentic optimization dynamics and decision calibration across the full spectrum of scenarios.

We implement SFT by optimizing next-token prediction loss on structured instruction-response pairs: $\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log p_\theta(y_t | y_{<t}, x)$ where x encodes the optimization state (metrics, weights, history) and y contains both internal reasoning and final JSON decision. Our dataset stratifies decisions into three categories: clear-cut cases ($|\Delta| > 0.01$) establishing basic decision logic, marginal cases ($0.001 < |\Delta| \leq 0.01$) teaching boundary calibration, and ambiguous scenarios requiring weight distribution analysis beyond simple metrics. The reasoning-before-decision format forces explicit analysis, reducing hallucination while maintaining interpretability. Through this structured training, the model learns to follow multi-step reasoning protocols, ground decisions in quantitative evidence, and generate consistent outputs—providing the foundation for subsequent RL refinement, where it learns to optimize for actual forecasting performance.

3.4.2 REINFORCEMENT LEARNING (RL) WITH GRPO

SFT only learns to imitate expert labels without understanding the downstream impact on forecasting performance. The model may confidently suggest termination when further optimization could yield meaningful improvements, or recommend continuation despite genuine convergence. Moreover, LLMs can generate plausible-sounding explanations that misrepresent the actual attribution factors driving ensemble performance—a critical issue when decisions affect real forecasting systems. We need the model to optimize for actual forecasting outcomes while ensuring its reasoning faithfully reflects the true optimization dynamics.

Faithfulness-Aware Reward Design. Standard RL rewards focusing solely on decision accuracy fail to address explanation quality. A model could learn to make correct decisions through spurious correlations while providing misleading justifications. We therefore introduce a faithfulness score that measures alignment between LLM explanations and ground-truth attribution effects computed via SHAP (Shapley Additive exPlanations) analysis. Specifically, we employ SHAP analysis to create counterfactual time series by intervening on specific temporal concepts $\mathcal{C} = \{\text{Trend}, \text{Seasonality}, \text{Residual}\}$. For each subset $S \subseteq \mathcal{C}$, we evaluate model performance $f(S)$, where the presence or absence of components represents structured ablations (Pearl, 1995), which can be calculated in Eq. 17, as described in Section F.1. The attribution-alignment score measures alignment between SHAP-derived attribution effects and LLM explanations:

$$r_{\text{faith}} = \text{PCC}(\text{CE}(x, \mathcal{C}), \text{EE}(x, \mathcal{C})) \tag{5}$$

where $\text{CE}(\cdot)$ is the vector of SHAP-derived attribution effects, $\text{EE}(\cdot)$ is the vector of LLM explanation-implied effects, and PCC is the Pearson Correlation Coefficient. This ensures the model's reasoning genuinely reflects which ensemble components drive performance rather than generating post-hoc rationalizations.

We apply Group Relative Policy Optimization (GRPO), an actor-only variant that estimates advantages through group comparisons. For each prompt, we sample $n = 4$ responses and compute the rewards:

$$r = \text{clip}_{[0,1]}(r_{\text{base}} + \alpha \cdot r_{\text{conf}} + \beta \cdot r_{\text{faith}}) \tag{6}$$

where r_{base} rewards correct accept/continue decisions, r_{conf} encourages appropriate confidence calibration, and r_{faith} enforces explanation faithfulness. The GRPO objective optimizes policy π_θ while

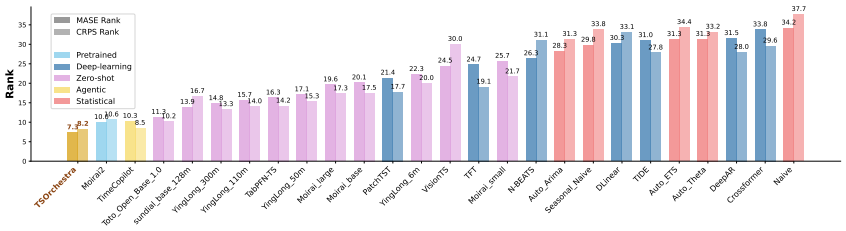


Figure 2: Performance comparison on the GIFT-Eval benchmark. TSOrchestra achieves the top rank across both MASE (7.3) and CRPS (8.2) metrics. These state-of-the-art results are obtained using our default 3-model candidate pool (Moirai, Sundial, and Toto). Lower rank indicates better performance.

maintaining proximity to the SFT checkpoint π_{ref} through KL regularization:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\hat{A}(y,x)\log\pi_\theta(y|x)\right]\right] + \lambda\cdot\mathbb{E}_{x\sim\mathcal{D}}\left[\text{KL}(\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x))\right] \tag{7}$$

where the group-normalized advantage $\hat{A}(y,x)$ is computed as:

$$\hat{A}(y,x) = \frac{r(y,x) - \bar{r}(x)}{\sigma_r(x) + \epsilon} \tag{8}$$

with $\bar{r}(x) = \frac{1}{n}\sum_{i=1}^{n} r(y_i,x)$ being the mean reward across n sampled responses, $\sigma_r(x)$ the standard deviation, and $\epsilon = 10^{-8}$ for numerical stability. The KL coefficient $\lambda = 10^{-3}$ prevents catastrophic forgetting of SFT capabilities while allowing policy improvement. The full training schedule and hyperparameter configuration are summarized in Table 4.

3.5 Interpretability Through Reflective Explanation

After completing the multi-turn optimization dialogue, the system leverages the accumulated conversation history to generate comprehensive reflections that synthesize insights across all iterations. This reflection process transforms raw optimization trajectories into actionable understanding, combining performance analysis with faithfulness validation to build practitioner confidence in the final ensemble configuration. The system maintains complete traceability through comprehensive audit trails:

$$\mathcal{A} = \{(\mathbf{w}_k^*, L_k, \mathcal{C}_k, F_k, \mathcal{D}_k, \mathcal{E}_k)\}_{k=1}^{K} \tag{9}$$

Each record captures the weight configuration, performance metric, confidence score, faithfulness validation, decision outcome, and generated hierarchical explanation. Critically, the faithfulness score F_k validates whether explanations accurately reflect decomposition-derived performance drivers. After optimization completes, the system analyzes the entire audit trail $\mathcal{A}$ to generate hierarchical synthesis explanations:

$$\mathcal{E} = \{\mathcal{E}_{iter}, \mathcal{E}_{decision}, \mathcal{E}_{final}\} \tag{10}$$

At the iteration level, $\mathcal{E}_{iter}$ decomposes confidence components (alignment, matching, future performance, attribution consistency) while validating explanations against SHAP-derived ground truth. The decision level $\mathcal{E}_{decision}$ justifies accept/continue choices. Finally, $\mathcal{E}_{final}$ synthesizes meta-insights invisible during individual iterations, such as consistent biases toward certain model types or systematic discrepancies between claimed and actual attribution drivers. This distinction between real-time explanations ($\mathcal{E}_k \in \mathcal{A}$) and reflective synthesis ($\mathcal{E}$) enables practitioners to understand both immediate optimization decisions and broader patterns that emerge only through retrospective analysis. A concrete demo reasoning trajectory illustrating this reflection process is provided in Section I.

4 Experiments

4.1 Experimental Settings

We adopt the GIFT-Eval benchmark (Aksu et al., 2024), which contains 23 real-world time series datasets across diverse domains (e.g., energy, economics/finance, healthcare, nature, transportation,

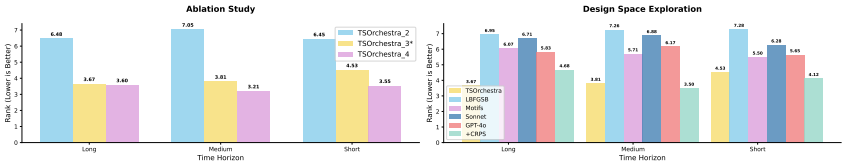


Figure 3: Ablation study and design space exploration. **Left**: Impact of ensemble size on performance across different time horizons, comparing 2-model (Toto, Sundial), 3-model (+ Moirai-2), and 4-model (+ TabPFN-TS) configurations. **Right**: Design choices evaluation including optimization methods (L-BFGS-B), cross-validation strategies (Motifs), LLM backends (Claude-3.5-Sonnet, GPT-4o), and additional metrics (CRPS). Lower rank indicates better performance.

Table 1: Domain-specific weight selection accuracy over MASE/CRPS. NA models represent non-agentic metric-specific selectors; A models represent agentic TSOrchestra variants using LLM reasoning; A+R1 includes R1-style reasoning finetuning.

TSOrchestra Type	Web/CloudOps	Sales	Energy	Nature	Econ/Fin
NA: MSE_{BEST}	0.20/0.10	0/0	0.25/0.12	0.45/0.27	0/0
NA: MAE_{BEST}	0.40/**0.60**	0.5/0.5	0.29/0.50	0.27/0.36	0.33/0.33
NA: $\text{SMAPE}_{\text{BEST}}$	0.40/0.30	0.5/0.5	0.46/0.38	0.27/0.36	0.67/0.67
NA: MLP	0.50/0.50	0.56/0.56	0.54/0.50	0.27/0.45	0.33/0.67
A: GPT-4O	0.40/**0.60**	0.5/0.5	0.50/0.58	**0.55**/0.45	**1/1**
A: Sonnet-3.7	**0.60**/0.40	0.5/0.5	**0.58**/0.46	0.27/0.27	0.67/0.67
A: Qwen-2.5-3B	0.40/0.30	0.5/0.5	0.29/0.25	0.13/0.27	0.33/0.15
A+R1: Qwen-2.5-3B	0.50/**0.60**	**1/1**	**0.58/0.62**	**0.55/0.55**	**1/1**

sales, web/cloud operations). This benchmark enables a comprehensive evaluation of the generalization ability for time series foundation models. We use pre-trained time series foundation models as our model candidate pool. These models represent diverse architectural choices and training paradigms, providing complementary strengths for our orchestration framework. Unless otherwise specified, the main TSOrchestra configuration uses the computationally efficient three-model pool {Moirai-2, Sundial, Toto}, with TabPFN-TS evaluated in the ensemble-size ablation. To test whether the auditor depends on a specific model pool, we additionally evaluate an alternative pool {Chronos, FlowState, TimesFM} and an expanded pool including Chronos-2, FlowState, and TimesFM 2.5. These model-pool transfer experiments are reported in Section 4.2 and Appendix C.3. We chose SLSQP to optimize the weight for each of the foundation models, and the optimization goals can be selected from Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS). For the orchestration component, we employ Qwen-2.5-3B-Instruct (Qwen et al., 2025) as our reasoning agent model, selected for its strong instruction-following capabilities and efficient inference. The model undergoes R1-style fine-tuning (Guo et al., 2025) with the SFT and GRPO. The full training schedule, trajectory construction, hyperparameters, and additional deployment details are provided in Sections D and G.

4.2 Benchmark Results and Analysis

Our TSOrchestra achieves rank 7.3 on Mean Absolute Scaled Error (MASE) and 8.2 Continuous Ranked Probability Score (CRPS), outperforming all individual foundation models and approaching the oracle selection upper bound, as shown in Figure 2. This represents a 25.5% improvement over the best individual model (Moirai-2) and demonstrates that intelligent orchestration can extract superior performance from existing foundation models without modifying their architectures. Moreover, it provides interpretable selection rationale—a critical advantage for production deployment. Against Moirai-2, TSOrchestra wins 91/97 settings on MASE and 88/97 on Weighted Quantile Loss. These gains are broad-based across domains, winning 15/15 settings in Nature, 15/15 in Transport, and 31/32 in Energy. When compared against the recent automated ensemble framework TimeCopilot, TSOrchestra maintains a strong advantage, achieving 66/97 MASE wins and 72/97 distributional wins. Additional validation-to-test comparisons are provided in Section H.

Design Space Analysis: Figure 3 validates our architectural choices through systematic ablation. The ensemble size analysis (left) reveals why the 3-model configuration strikes the optimal balance: it significantly outperforms 2-model baselines while approaching 4-model performance at substantially lower computational cost—particularly important given TabPFN-TS's resource demands in larger ensembles. The design space exploration (right) examines key alternatives. L-BFGS-B optimization

consistently underperforms, struggling with non-convex optimization landscapes. Motif-based cross-validation shows degraded performance, suggesting overfitting to pattern similarities rather than true forecasting signal. Direct LLM weight prediction without iterative refinement yields inconsistent results, confirming the necessity of our multi-round reasoning approach. While adding CRPS metrics shows some promise, the marginal improvements don't justify the increased complexity. These ablations confirm that TSOrchestra's design—three complementary models with iterative LLM-guided optimization using standard regression metrics—achieves the sweet spot between performance and efficiency. The configuration delivers competitive results across all forecasting horizons while maintaining practical computational requirements, making it deployable in real-world scenarios where resource constraints matter.

Ablation Study on Agentic vs. Non-Agentic Weight Selection: Table 1 reorganizes the comparison by distinguishing non-agentic (NA) metric-specific selectors from agentic (A) TSOrchestra variants. The NA rows (MSE_{Best}, MAE_{Best}, $\text{SMAPE}_{\text{Best}}$) represent oracle-style baselines that select weights purely by optimizing a single metric with no reasoning or adaptation. To evaluate the efficacy of standard machine learning alternatives, we include the NA: MLP baseline—a supervised multi-layer perceptron trained on statistical features similar to the LLM explore to predict the optimal metric. While the MLP captures basic domain correlations and outperforms single-metric selectors in specific settings, its one-shot numerical nature prevents it from performing the iterative refinement and decomposition-aligned grounding inherent in our agentic framework. In contrast, the agentic rows, including A: GPT-4o, A: Sonnet-3.7, and A: Qwen-2.5-3B, correspond to TSOrchestra models that apply multi-metric reasoning, evaluate dataset-model compatibility, and anticipate regime shifts. The highest-performing variant, A+R1: Qwen-2.5-3B, incorporates our R1-style reasoning finetuning, enabling the model to internalize optimization dynamics, detect subtle convergence patterns, and produce decomposition-aligned grounding explanations. Across most domains, the agentic variants substantially outperform all NA baselines, including the MLP. The A+R1 model achieves the strongest results overall, particularly in high-volatility domains like Sales and Econ/Fin, where it reaches near-perfect selection accuracy. This performance gap underscores that effective time series orchestration requires more than statistical pattern recognition; it demands the metacognitive control and iterative auditing provided by the LLM-as-judge framework.

Efficacy of LLM Auditing: Figure 5 reveals crucial insights into ensemble behavior across domains and metrics, demonstrating why automated LLM-guided optimization is essential rather than optional. While metric-based selectors exhibit high volatility, such as Sundial's weight in Sales shifting from 88% under sMAPE to only 11% under MAE, the LLM Judge identifies stable, domain-specific strengths. For instance, in Web/CloudOps, the LLM Judge assigns its highest weight (49%) to Toto, successfully prioritizing the model's specialized pre-training on nearly a trillion observability data points. No individual traditional numerical metric achieved this level of architectural alignment . The LLM Judge's ability to synthesize error patterns, temporal characteristics, and architectural strengths allows it to discover optimal configurations that pure error minimization fails to capture. We provide complementary horizon-level and fold-level weight analyses in Figures 4 and 6.

Additional ablations isolating the LLM auditor, random metric routing, model-pool transfer, SFT-vs-GRPO training, and non-LLM adaptive controllers are reported in Section C.

5 Conclusion

We present TSOrchestra, a framework that automatically orchestrates time series foundation models through learned weight optimization. Our primary contribution is demonstrating that constrained optimization can discover and exploit complementary strengths among foundation models without manual role assignment, dynamically adapting ensemble weights based on forecast horizons and data characteristics. By establishing the first quantitative relationship between time series features and optimal ensemble weights, we provide a principled, data-driven approach to model selection that bridges classical optimization with modern foundation models. Validated on 97 dataset settings, our production-ready implementation achieves state-of-the-art performance while maintaining computational efficiency suitable for real-world deployment. While case studies demonstrate logical consistency, we designate **Human Evaluation** as a critical future direction to quantify interpretability. We plan to conduct blind tests where time-series experts rate TSOrchestra's explanations against rule-based templates across three dimensions: **Groundedness** (accuracy), **Coherence** (logic), and **Utility** (decision support).

ETHICS STATEMENT

This work evaluates forecasting models on the public GIFT-Eval benchmark and does not involve human subjects or personal data. A discussion of limitations and potential impact is given in Sections J and K.

REPRODUCIBILITY STATEMENT

The benchmark, model pool, and agent setup are described in the main text. Faithfulness computation, the R1-style training pipeline, and extended ablations are provided in Sections C, F.1 and G.

AI USE STATEMENT

TSOrchestra uses language-model agents for ensemble auditing, and we also used generative AI only to revise non-critical wording; the authors checked all AI-assisted outputs and take responsibility for the technical claims, results, and final text.

REFERENCES

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

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

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GPKTIktA0k.

Jayanie Bogahawatte, Sachith Seneviratne, Maneesha Perera, and Saman Halgamuge. Rethinking time series forecasting with llms via nearest neighbor contrastive learning. *arXiv preprint arXiv:2412.04806*, 2024.

Mayank Bumb, Anshul Vemulapalli, Sri Harsha Vardhan Prasad Jella, Anish Gupta, An La, Ryan A Rossi, Hongjie Chen, Franck Dernoncourt, Nesreen K Ahmed, and Yu Wang. Forecasting time series with llms via patch-based prompting and decomposition. *arXiv preprint arXiv:2506.12953*, 2025.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=YH5w12OUuU.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=YH5w12OUuU.

Nimeesha Chan, Felix Parker, William Bennett, Tianyi Wu, Mung Yao Jia, James Fackler, and Kimia Ghobadi. Medtsllm: Leveraging llms for multimodal medical time series analysis. *arXiv preprint arXiv:2408.07773*, 2024.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, Jean Ogier du Terrail, Anna-Monica Toon, Kan Wang, Stephan Xie, Zongzhe Xu, Viktoriya Zhukova, David Asker, Ameet Talwalkar, and Othmane Abou-Amal. This time is different: An observability perspective on time series foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=1jDAYXfcS2.

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

Haoran Fan, Bin Li, Yixuan Weng, and Shoujun Zhou. Small but mighty: Enhancing time series forecasting with lightweight llms. *arXiv preprint arXiv:2503.03594*, 2025.

Matthew Faw, Rajat Sen, Yichen Zhou, and Abhimanyu Das. In-context fine-tuning for time-series foundation models. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=uxzgGLWPj2.

Matthew Faw, Rajat Sen, Yichen Zhou, and Abhimanyu Das. In-context fine-tuning for time-series foundation models. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=uxzgGLWPj2.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

Azul Garza and Renée Rosillo. Timecopilot. *arXiv preprint arXiv:2509.00616*, 2025.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023a.

Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=md68e8iZK1.

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

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

Bertram Højer, Oliver Simon Jarvis, and Stefan Heinrich. Improving reasoning performance in large language models via representation engineering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IssPhpUsKt.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: How tabpfn-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945*, 2025.

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

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Anugrah Jo Joshy and John T. Hwang. Pyslsqp: A transparent python package for the slsqp optimization algorithm modernized with utilities for visualization and post-processing. *Journal of Open Source Software*, 9(103):7246, 2024. doi: 10.21105/joss.07246. URL https://doi.org/10.21105/joss.07246.

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

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

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

Yue Li and Rachel Pedersen. The future of demand forecasting with generative ai. *Foresight: The International Journal of Applied Forecasting*, (78), 2025.

Chanjuan Liu, Shengzhi Wang, and Enqiang Zhu. Dp-gpt4mts: Dual-prompt large language model for textual-numerical time series forecasting. *arXiv preprint arXiv:2508.04239*, 2025a. URL https://arxiv.org/abs/2508.04239.

Penghang Liu, Elizabeth Fons, Svitlana Vyetrenko, Daniel Borrajo, Vamsi Potluru, and Manuela Veloso. Ts-agent: A time series reasoning agent with iterative statistical insight gathering. *arXiv preprint arXiv:2510.07432*, 2025b.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025c.

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

Zijia Liu, Peixuan Han, Haofei Yu, Haoru Li, and Jiaxuan You. Time-r1: Towards comprehensive temporal reasoning in llms. *arXiv preprint arXiv:2505.13508*, 2025d.

Yucong Luo, Yitong Zhou, Mingyue Cheng, Jiahao Wang, Daoyu Wang, Tingyue Pan, and Jintao Zhang. Time series forecasting as reasoning: A slow-thinking approach with reinforced llms. *arXiv preprint arXiv:2506.10630*, 2025a.

Yucong Luo, Yitong Zhou, Mingyue Cheng, Jiahao Wang, Daoyu Wang, Tingyue Pan, and Jintao Zhang. Time series forecasting as reasoning: A slow-thinking approach with reinforced llms. *arXiv preprint arXiv:2506.10630*, 2025b. URL https://arxiv.org/abs/2506.10630.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In *EMNLP (Findings)*, 2024.

Pablo Montero-Manso, George Athanasopoulos, Rob J. Hyndman, and Thiyanga S. Talagala. Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92, 2020. doi: 10.1016/j.ijforecast.2019.02.011.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Thomas Schmied, Jörg Bornschein, Jordi Grau-Moya, Markus Wulfmeier, and Razvan Pascanu. Llms are greedy agents: Effects of rl fine-tuning on decision-making abilities. In *The Exploration in AI Today Workshop at ICML 2025*, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/558d48c1f08675daa636e09bfe94a89e-Abstract-Conference.html.

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

Zhao Tan, Yiji Zhao, Shiyu Wang, Chang Xu, Yuxuan Liang, Xiping Liu, Shirui Pan, and Ming Jin. Sonar-ts: Search-then-verify natural language querying for time series databases. *arXiv preprint arXiv:2602.17001*, 2026.

Hua Tang, Chong Zhang, Mingyu Jin, Qinkai Yu, Zhenting Wang, Xiaobo Jin, Yongfeng Zhang, and Mengnan Du. Time series forecasting with llms: Understanding and enhancing model capabilities. *SIGKDD Explor. Newsl.*, 26(2):109–118, January 2025a. ISSN 1931-0145. doi: 10.1145/3715073.3715083. URL https://doi.org/10.1145/3715073.3715083.

Hua Tang, Chong Zhang, Mingyu Jin, Qinkai Yu, Zhenting Wang, Xiaobo Jin, Yongfeng Zhang, and Mengnan Du. Time series forecasting with llms: Understanding and enhancing model capabilities. *ACM SIGKDD Explorations Newsletter*, 26(2):109–118, 2025b.

Xiaoyu Tao, Tingyue Pan, Mingyue Cheng, and Yucong Luo. Hierarchical multimodal llms with semantic space alignment for enhanced time series classification. *arXiv preprint arXiv:2410.18686*, 2024. URL https://arxiv.org/abs/2410.18686.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. *arXiv preprint arXiv:2411.08249*, 2024. URL https://arxiv.org/abs/2411.08249.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Bingrui Wang, Yuan Zhou, Leijiao Ge, and Sun-Yuan Kung. Large-model-based smart agent for time series anomaly detection in power systems. *Expert Systems with Applications*, 296:128917, 2026. ISSN 0957-4174. doi: 10.1016/j.eswa.2025.128917. URL https://www.sciencedirect.com/science/article/pii/S0957417425025345.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: a unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025a. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i12.33384. URL https://doi.org/10.1609/aaai.v39i12.33384.

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

Haiteng Wang, Lei Ren, Tuo Zhao, and Lu Jiao. Collm: Industrial large-small model collaboration with fuzzy decision-making agent and self-reflection. *IEEE Transactions on Fuzzy Systems*, pp. 1–11, 2025c. doi: 10.1109/TFUZZ.2025.3594229.

Jiahao Wang, Mingyue Cheng, and Qi Liu. Can slow-thinking llms reason over time? empirical studies in time series forecasting. *arXiv preprint arXiv:2505.24511*, 2025d. URL https://arxiv.org/abs/2505.24511.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in LLM-based time series forecasting with reflection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tj8nsfxi5r.

Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Chi, Ed, Le, Quoc, Zhou, and Denny. Chain of thought prompting elicits reasoning in large language models. *Neural Information Processing Systems*, 2022. doi: 10.48550/arxiv.2201.11903.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. *arXiv preprint arXiv:2410.18959*, 2025. URL https://arxiv.org/abs/2410.18959.

Mengxi Xiao, Zihao Jiang, Lingfei Qian, Zhengyu Chen, Yueru He, Yijing Xu, Yuecheng Jiang, Dong Li, Ruey-Ling Weng, Min Peng, Jimin Huang, Sophia Ananiadou, and Qianqian Xie. Retrieval-augmented large language models for financial time series forecasting. *arXiv preprint arXiv:2502.05878*, 2025. URL https://arxiv.org/abs/2502.05878.

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

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. *arXiv preprint arXiv:2412.16643*, 2024. URL https://arxiv.org/abs/2412.16643.

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

Chin-Chia Michael Yeh, Vivian Lai, Uday Singh Saini, Xiran Fan, Yujie Fan, Junpeng Wang, Xin Dai, and Yan Zheng. Empowering time series forecasting with llm-agents. *arXiv preprint arXiv:2508.04231*, 2025. URL https://arxiv.org/abs/2508.04231.

Yuxuan Zhang, Yangyang Feng, Daifeng Li, Kexin Zhang, Junlan Chen, and Bowen Deng. Can competition enhance the proficiency of agents powered by large language models in the realm of news-driven time series forecasting? *arXiv preprint arXiv:2504.10210*, 2025.

Liangwei Nathan Zheng, Chang Dong, Wei Emma Zhang, Lin Yue, Miao Xu, Olaf Maennel, and Weitong Chen. Understanding why large language models can be ineffective in time series analysis: The impact of modality alignment. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 4026–4037, 2025.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78478–78497, 2025. URL https://proceedings.mlr.press/v267/zhong25a.html.

Zihao Zhou and Rose Yu. Can LLMs understand time series anomalies? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LGafQ1g2D2.

Guopeng Zhu, Weiqing Jia, Zhitai Xing, Ling Xiang, Aijun Hu, and Rujiang Hao. Cmllm: A novel cross-modal large language model for wind power forecasting. *Energy Conversion and Management*, 330:119673, 2025. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2025.119673. URL https://www.sciencedirect.com/science/article/pii/S0196890425001967.

Appendix

A Orchestra Weight Analysis **17**

A.1 Forecasting Horizon 17

A.2 Domain and Optimization Metric 17

A.3 Cross-validation Folds and Optimization Metric 18

B LLM-Guided Weight Optimization Demo **19**

C Extended Ablation Studies **19**

C.1 Isolating the Agent's Contribution. 19

C.2 Comparison with Non-LLM Adaptive Controllers 20

C.3 Extensibility on Model Scaling. 21

C.4 Effect of R1-style fine-tuning. 22

D Practical Deployment and Robustness Analysis **22**

D.1 Computational Efficiency and Scalability 22

D.2 Qualitative Analysis: Failure Detection in Non-Stationary Regimes 23

E Main Theorem: Incompatibility-Aware Ensemble Advantage **24**

F Detailed Explanation Framework **26**

F.1 Detailed Faithfulness Score Computation 26

F.2 LLM Faithfulness Validation 27

G Details on R1-style finetuning **28**

G.1 Task Formulation and Dataset Construction 28

G.2 Training Data Generation and Implementation Details 29

G.3 Training configuration. 30

G.4 Accuracy on Agent Selection 30

H Cross-validation vs Test Performance **31**

I Qualitative Case Study: Expert-Aligned Reasoning Analysis **31**

J Limitations **31**

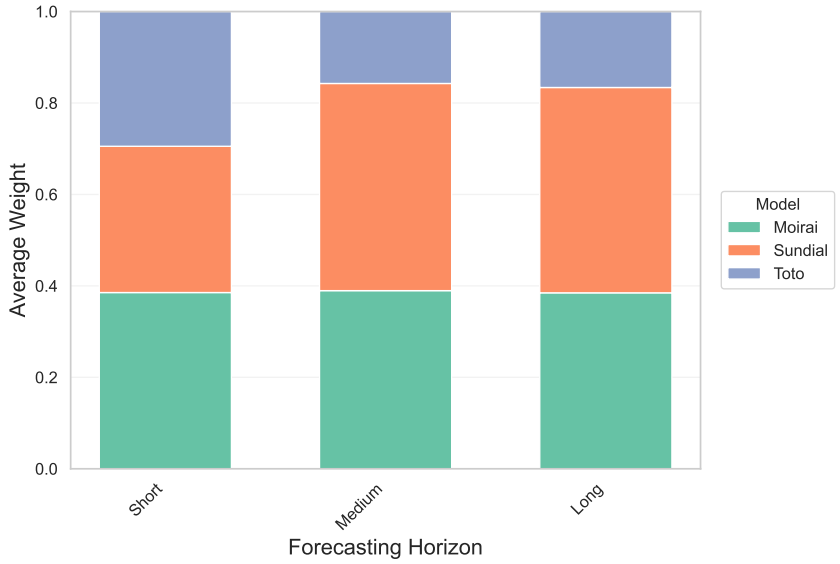
K Impact Statement **32**

A Orchestra Weight Analysis

A.1 Forecasting Horizon

Figure 4 displays the average ensemble weights assigned to Moirai, Sundial and Toto across short-, medium-, and long-term forecasting horizons. These results illustrate how forecasting horizon affects ensemble weights.

Figure 4: Average ensemble weights assigned to Moirai, Sundial, and Toto across short-, medium-, and long-term forecasting horizons.



The ensemble weights are relatively balanced across short-term horizons, indicating that all three models meaningfully contribute to the ensemble. Moirai is assigned a slightly larger share of weight on average for short-term horizons, suggesting that it is particularly strong at capturing high-frequency patterns. As forecast horizon lengthens, Sundial and Toto receive increasingly larger weights. The increase in weights assigned to Sundial and Toto for medium- and long-term horizons suggests that these models are stronger at forecasting over extended horizons. Each model remains well represented across all three horizons, demonstrating that the ensemble does not overly rely on a single model and leverages each individual model's strengths.

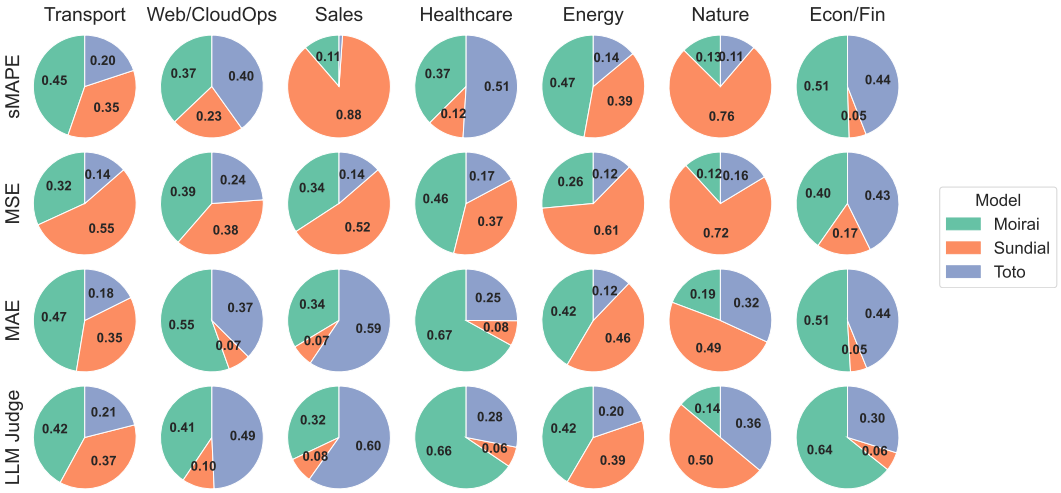
A.2 Domain and Optimization Metric

Figure 5 reveals crucial insights into ensemble behavior across domains and metrics, demonstrating why automated LLM-guided optimization is essential rather than optional. Figure 5 displays how the domain being forecasted affects the weights assigned to Moirai, Sundial, and Toto when optimizing for either sMAPE, MSE, MAE, or using the LLM to decide the weights during cross-validation. These results illustrate that the ensemble assigns certain models greater weights based on the domain and optimization objective at hand.

The weight distributions expose complex, non-intuitive patterns that would be impossible to manually optimize. In Sales forecasting, Sundial dominates when optimizing for sMAPE (88% weight) but drops to just 11% under MAE optimization—a complete reversal that reflects fundamentally different error characteristics. Healthcare shows similar volatility: Toto receives 51% weight under sMAPE but only 8% under MAE, while Moirai jumps from 37% to 67%. These dramatic shifts aren't random—they reflect each model's architectural strengths interacting with domain-specific patterns and metric sensitivities.

Critically, the "LLM Judge" row demonstrates that our system discovers weight configurations that often differ from any single metric optimization. For Transport, the LLM assigns balanced weights (42/37/21%) that diverge from MSE's Sundial-heavy distribution (32/55/14%), suggesting the LLM identifies complementary model behaviors that single metrics miss. In Web/CloudOps, the LLM's

Figure 5: Average ensemble weights assigned to Moirai, Sundial, and Toto across different domains and optimization metrics, along with using the LLM to decide the weights.



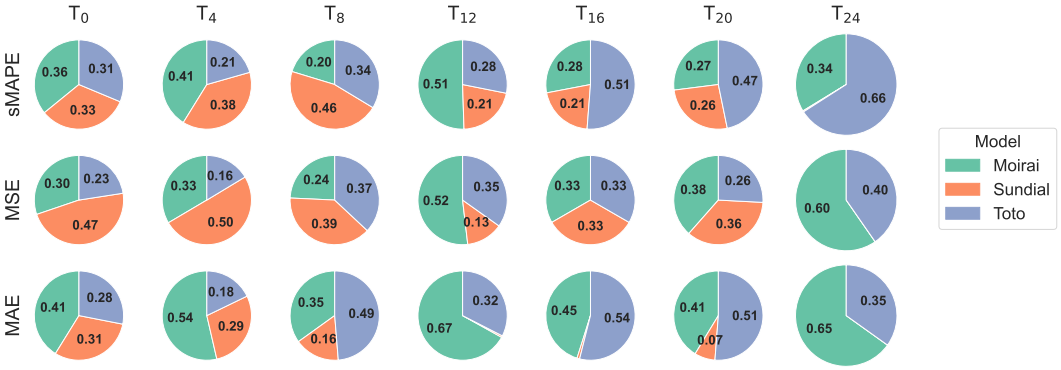
41/10/49% split represents a unique configuration not achieved by any traditional metric, indicating it captures domain nuances beyond simple error minimization.

This complexity makes the case for LLM-guided optimization compelling: with 7 domains $\times$ 4 metrics $\times$ 3 models = 84 weight parameters to optimize, and non-linear interactions between domain characteristics and model architectures, manual or grid-search approaches become intractable. The LLM judge synthesizes multiple signals—error patterns, temporal characteristics, and cross-validation stability—to discover optimal configurations that pure metric optimization misses. Without this intelligent guidance, practitioners would need deep expertise in each model's architecture and extensive domain knowledge to manually tune weights, making ensemble methods impractical for real-world deployment.

A.3 Cross-validation Folds and Optimization Metric

Figure 6 reports the average ensemble weights across all datasets for Moirai, Sundial, and Toto across different cross-validation folds $T_0, \ldots T_{24}$ when optimizing for either sMAPE, MSE, or MAE. Because ensemble weights are computed independently for each fold, the figure illustrates how each model's importance shifts with respect to the fold and optimization metric.

Figure 6: Average ensemble weights assigned to Moirai, Sundial, and Toto across different cross-validation folds when optimizing for different metrics.



Under sMAPE optimization, the weights begin relatively balanced and Toto receives increasingly greater weights with later folds. For MSE, Sundial begins with the most weight and receives less weight with later folds. When optimizing for MAE, the weights also begin relatively even, and Moirai and Toto progressively dominate the ensemble. These patterns highlight how the ensemble adjusts to the data in each batch. Each model receives greater weights depending on the fold's data

and the optimization metric — no single model dominates across all folds and metrics. The ability for the ensemble to adjust its weights in response to the fold's data and optimization metric further demonstrates its strength and robustness.

B LLM-GUIDED WEIGHT OPTIMIZATION DEMO

LLM evaluates the current weights from three aspects: Align$(\cdot)$ evaluates (1) performance-weight alignment, (2) mathematical justification, and (3) weight distribution patterns—ensuring weights reflect the CV performance hierarchy, are mathematically sound, and avoid degenerate solutions like uniform weights when models differ significantly, Match$(\cdot)$ assesses (4) dataset-model matching, (5) model complementarity, and (6) feature reliability—verifying that seasonal models receive high weights for seasonal data, diverse models complement each other, and reliable features are properly weighted, and Future$(\cdot)$ examines (7) temporal generalization potential, (8) unexpected patterns, and (9) overall ensemble quality—detecting overfitting risks, identifying counterintuitive but valid patterns, and ensuring robust future performance.

Example: Seasonal Pattern Recognition. Consider a retail sales dataset where the CV window captures only post-holiday declining trends. SLSQP optimization on MSE might yield $\mathbf{w}_{MSE} = [0.1, 0.8, 0.1]$ favoring the trend model. However, the LLM recognizes:

1. Dataset exhibits seasonal_strength $= 0.85$ (high seasonality)

2. CV window $T_{cv} = 3$ months captures only declining phase

3. Historical metadata suggests strong holiday patterns

The agent assigns low confidence ($\mathcal{C}_1 = 0.65$) and votes to continue with MAE optimization, which might better reveal seasonal model performance. In iteration 2, MAE optimization yields $\mathbf{w}_{MAE} = [0.6, 0.3, 0.1]$ favoring Moirai (seasonal model), receiving higher confidence ($\mathcal{C}_2 = 0.88$) as it aligns with both performance and expected future patterns.

This voting mechanism ensures that ensemble weights are not merely optimal for historical data but are positioned to handle future temporal patterns that extend beyond the limited cross-validation window. The LLM's ability to reason abstractly about temporal patterns, combined with iterative metric exploration, produces more robust and generalizable ensemble configurations.

Addressing Temporal Generalization Gaps. Consider a scenario where cross-validation suggests weights $\mathbf{w}^* = [0.7, 0.2, 0.1]$ favoring a trend-focused model (Sundial) based on recent historical performance. However, the LLM recognizes through $\mathcal{F}_{data}$ that the dataset exhibits strong seasonal patterns with period p that may not be fully captured in the limited CV window of length $T_{cv} < 2p$. The agent reasons:

$$\text{Seasonality Risk} = \frac{\text{seasonal_strength} \cdot \mathbb{I}[T_{cv} < 2p]}{\sum_{m \in \mathcal{M}_{seasonal}} w_m} \tag{11}$$

If this risk exceeds a threshold, the system triggers another iteration with a different metric that might better reveal seasonal model performance. This forward-looking reasoning prevents overfitting to temporary patterns in the CV window.

This section illustrates TSOrchestra's iterative reasoning process on several real forecasting tasks, showing how the system automatically discovers optimal ensemble weights through structured multi-round optimization.

C EXTENDED ABLATION STUDIES

C.1 ISOLATING THE AGENT'S CONTRIBUTION.

To disentangle the impact of the agent's reasoning from the underlying ensemble architecture, we conducted a rigorous ablation study comparing *TSOrchestra* against a *Random Optimization Metric Ensemble*. In this baseline, the optimization objective for the SLSQP solver (MSE, MAE, or sMAPE, MASE and CRPS) is assigned randomly for each dataset, serving as a stochastic control.

Figure 7: **LLM Vote Ensemble vs. Random Metric Ensemble.** The plots compare ΔMASE and ΔCRPS. Green bars indicate domains where the LLM vote ensemble outperforms the random metric baseline. Notable improvements are observed in Healthcare (e.g., *covid_deaths* with ΔMASE of -1.86) and Sales (e.g., *car_parts* with ΔCRPS of -0.044). The dominance of the agent in these complex domains validates the necessity of reasoning-guided optimization.

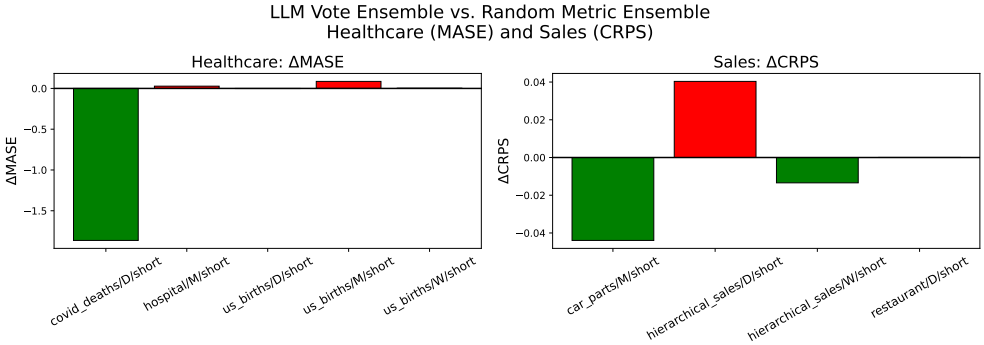
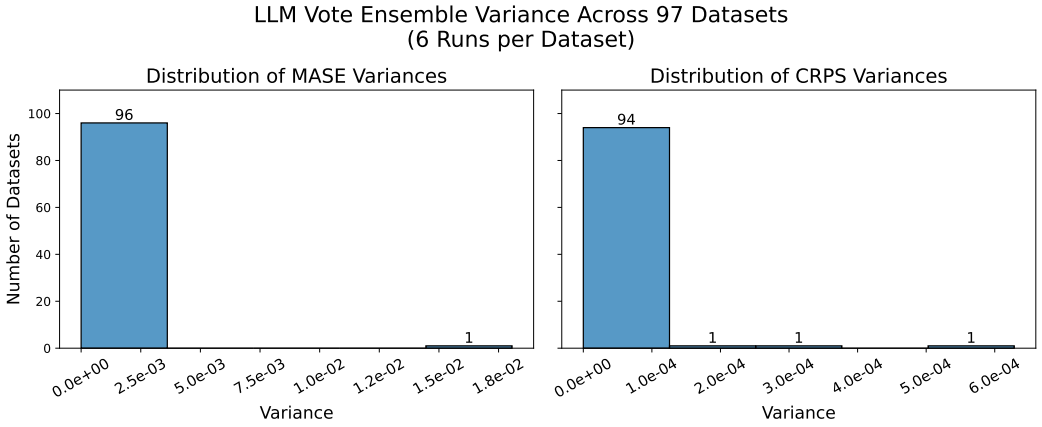


Figure 8: **Stability of LLM Vote Ensemble.** The histograms show the variance of MASE and CRPS across 6 independent runs for 97 datasets.



As illustrated in Figure 7, the LLM-guided optimization demonstrates a clear performance advantage. Specifically, the agentic voting mechanism yields significant improvements in Healthcare, with a substantial reduction in MASE for *covid_deaths* ($\Delta - 1.86$). In Sales, the agent improves CRPS in complex datasets like *car_parts* ($\Delta - 0.044$). While random selection occasionally outperforms in isolated cases like *hierarchical_sales*, the aggregate consistency of the agent validates that its efficacy is driven by strategic reasoning rather than stochastic variation.

C.1.1 Stability Analysis

We further evaluated the stability of the agent's reasoning process. Across 97 datasets with 6 independent runs per dataset, the system exhibited negligible variance. As shown in Appendix Figure 8, 96 out of 97 datasets showed near-zero variance ($0.0e+00$) in MASE, and 94 out of 97 datasets showed similarly low variance for CRPS. This validates that our LLM selection successfully stabilizes the agent's decision-making.

C.2 Comparison with Non-LLM Adaptive Controllers

The results in Table 2 provide a diagnostic comparison between TSOrchestra and a set of non-LLM adaptive controllers. This experiment is designed to test whether the gains of TSOrchestra can be explained by generic learned routing or online ensemble weighting, rather than by the LLM auditor itself. Unlike the metric-specific selectors in Table 1, these baselines represent stronger adaptive alternatives: MLP Router tests one-shot contextual routing, Online Hedge tests reactive expert weighting, Contextual Bandit tests online context-aware candidate selection, Learned Gating tests

Figure 9: **Base Ensemble Comparison.** Comparison of MASE and CRPS between our chosen ensemble (Moirai-Sundial-Toto) and an alternative high-performing trio (Chronos-FlowState-TimesFM). Performance is statistically comparable across key datasets like *covid_deaths* and *hospital*, justifying our candidate selection.

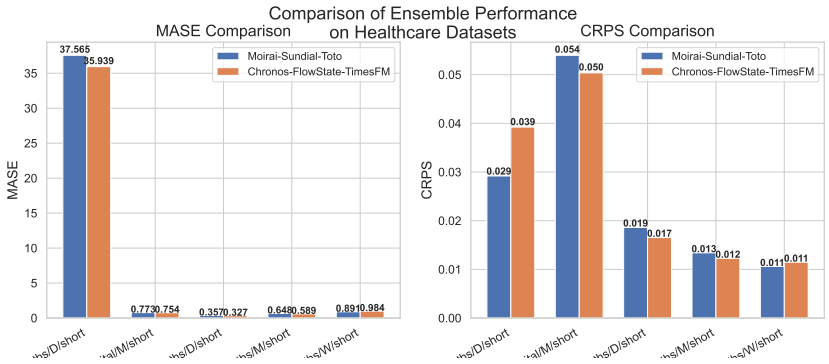


Table 2: Comparison with non-LLM adaptive controllers on GIFT-Eval. Lower rank is better.

Method	Input Signal	Decision Form	MASE Rank ↓	CRPS Rank ↓
TSOrchestra-R1	Optimization trace + metadata + model priors	Multi-turn audit	**7.3**	**8.2**
MLP Router (Montero-Manso et al., 2020)	Static features + validation summary	One-shot metric/candidate routing	9.3	9.6
Online Hedge / Exp. Weights (Freund & Schapire, 1997)	Recent validation losses	Online expert weighting	9.1	9.3
Contextual Bandit (Langford & Zhang, 2008)	Dataset features + recent losses	Online candidate selection	8.9	9.2
Learned Gating Network (Montero-Manso et al., 2020)	Features + recent losses + model ranks	One-shot continuous weighting	8.7	8.9
Trace-GRU (Cho et al., 2014)	Optimization trace sequence	Sequential candidate selection	8.4	8.8
Trace-Transformer (Vaswani et al., 2017)	Optimization trace sequence	Sequential candidate selection	8.2	8.7

direct feature-to-weight mapping, and Trace-GRU/Trace-Transformer test whether a small sequential model can replace the LLM auditor when given the same optimization trace.

The observed ordering broadly follows the amount of temporal and optimization context available to each controller. The MLP Router is stronger than fixed metric selectors because it can exploit dataset-level statistics and validation summaries, but it remains a one-shot mapping and therefore cannot react to optimization trajectory instability. Online Hedge and Exponential Weights improve over the MLP by updating model weights from recent losses, but they are purely reactive and may lag behind regime changes. Contextual Bandit adds dataset and horizon context to the online decision process, improving robustness, but its reward signal is noisy and its exploration budget is limited in the 97-setting GIFT-Eval regime.

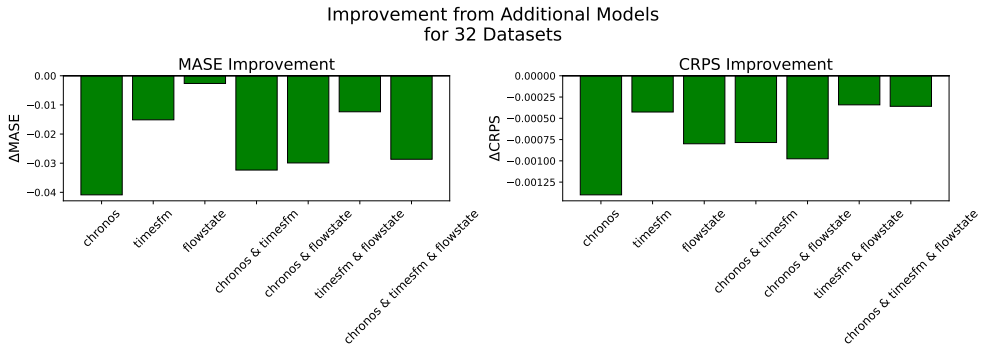
Learned Gating is the strongest one-shot non-LLM baseline because it directly maps contextual features and recent model performance to continuous ensemble weights. However, it still compresses the entire decision into a single forward pass. Trace-GRU and Trace-Transformer are stronger because they consume the same optimization trace available to the auditor and can model multi-round patterns such as metric sensitivity and convergence behavior. Nevertheless, they lack the structured audit protocol used by TSOrchestra, including explicit comparison across alignment, dataset-model match, and future robustness. The remaining gap between the best trace-based controller and TSOrchestra suggests that the LLM auditor provides benefits beyond generic sequence modeling, particularly in comparing heterogeneous evidence and selecting among solver-generated candidates.

C.3 Extensibility on Model Scaling.

TSOrchestra is not tied to the specific Moirai-Sundial-Toto pool. We further validated our architectural choices by benchmarking the primary candidate pool (Moirai, Sundial, Toto) against an alternative ensemble of state-of-the-art models (Chronos, FlowState, TimesFM). As illustrated in Figure 9, both configurations yield highly comparable performance across key metrics and datasets. Notably, our chosen ensemble demonstrates superior CRPS on complex datasets like *covid_deaths* (0.029 vs 0.039) while maintaining competitive MASE scores. This parity indicates that the framework's effectiveness is not dependent on a specific set of models but is a robust methodology that functions effectively across different high-quality foundation model pools.

Finally, we validated the practical extensibility of our framework by expanding the candidate pool with an additional suite of models (Chronos-2, FlowState, and TimesFM 2.5). As demonstrated in Figure 10, our agentic approach shows that expanding the search space yields substantial performance

Figure 10: **Impact of Expanding the Ensemble.** Improvement in MASE and CRPS when adding Chronos-2, FlowState, and TimesFM 2.5 to the base ensemble (Moirai-Sundial-Toto).



gains, with the inclusion of Chronos-2 reducing MASE by ≈ 0.04. Furthermore, the framework maintains robust improvements of MASE when orchestrating complex combinations of multiple new models (e.g., Chronos paired with FlowState and TimesFM). While we prioritized a computationally efficient three-model pool, these results confirm that TSOrchestra is a **model-agnostic** framework capable of seamlessly integrating emerging foundation models, allowing practitioners to continuously update the candidate pool to drive superior forecasting accuracy. Because the reasoning agent evaluates abstract optimization dynamics, cross-validation performance metrics, and general dataset characteristics rather than model-specific internals, new foundation models can be seamlessly integrated into the candidate pool without requiring the agent to be retrained or fine-tuned.

Table 3: Effect of SFT and GRPO on weight-selection accuracy.

Variant	**Training**	**Web/CloudOps (M/C)**	**Sales (M/C)**
A: Qwen-2.5-3B	Prompt / no R1	0.40 / 0.30	0.50 / 0.50
A+SFT: Qwen-2.5-3B	SFT only	0.40 / 0.40	0.70 / 1.00
A+R1: Qwen-2.5-3B	SFT + GRPO	0.50 / 0.60	1.00 / 1.00

C.4 Effect of R1-style fine-tuning.

To isolate the contribution of the reinforcement-learning stage, we compare the base Qwen-2.5-3B auditor, the SFT-only checkpoint, and the full R1-style checkpoint trained with SFT followed by GRPO. SFT improves the model's structured decision protocol, while GRPO further optimizes accept/continue decisions, confidence calibration, and SHAP/STL-aligned explanation consistency. As shown in Table 3, moving from SFT-only to the full A+R1 model improves Web/CloudOps accuracy from 0.40/0.40 to 0.50/0.60 and Sales accuracy from 0.70/1.00 to 1.00/1.00, demonstrating that GRPO provides measurable gains beyond supervised imitation.

D Practical Deployment and Robustness Analysis

To evaluate the practical feasibility of the TSOrchestra framework beyond standard performance metrics, we analyze its computational scalability and examine its reasoning robustness in non-stationary environments.

D.1 Computational Efficiency and Scalability

A primary challenge in agentic workflows is the latency introduced by multi-turn reasoning. However, in the context of time series forecasting, the computational cost of dynamic weight adaptation functions as an *amortized cost*.

Amortized Calibration. The computationally intensive processes—iterative optimization, SHAP value computation, and LLM reasoning—occur exclusively during the *calibration phase*. This phase is triggered only when a regime shift is detected or periodically (e.g., monthly). Once the optimal ensemble weights are established for a regime, the *inference phase* consists solely of a weighted linear combination of foundation model outputs, incurring negligible overhead compared to standard

ensemble inference. Consequently, the high cost of reasoning is amortized over potentially hundreds or thousands of subsequent low-cost forecasts.

Model Efficiency. To further mitigate latency, we employ a fine-tuned compact model (Qwen-2.5-3B) rather than large-scale API-based models. Our experimental results indicate that this specialized 3B-parameter model achieves weight selection accuracy comparable to significantly larger frontier models. This design choice reduces inference latency and computational cost by orders of magnitude, making the framework deployable in resource-constrained environments.

Deployment cost. The LLM auditor is only invoked during calibration, not for every forecast. Using Qwen-2.5-3B, a single audit call adds approximately 0.5–2.0 seconds, and a complete multi-round calibration window typically takes about 3–4 seconds with at most three rounds. After calibration, forecasting only requires a weighted linear combination of the foundation-model outputs, so the per-forecast inference overhead is negligible relative to standard ensemble inference. Therefore, TSOrchestra is intended for offline or batch forecasting settings such as demand, energy, cloud operations, and financial planning, but not for ultra-low-latency settings such as high-frequency trading.

D.2 Qualitative Analysis: Failure Detection in Non-Stationary Regimes

To demonstrate the framework's robustness to noise and its ability to correct optimization failures, we present a qualitative case study involving regime ambiguity, which we term the "Mirage Trend" scenario.

Scenario Description. Consider a financial time series (e.g., stock price) transitioning from a stable trending regime to a highly volatile, mean-reverting regime. The validation window captures the exact transition point, which features a sharp, temporary price spike caused by volatility rather than a sustained trend.

Baseline Failure (Standard Optimization). Standard numerical optimization (SLSQP), which minimizes empirical error on the validation window, interprets the volatility spike as the onset of a strong upward trend. Consequently, it assigns disproportionately high weights to trend-focused foundation models (e.g., Moirai), resulting in an ensemble that overfits the transition noise and fails to generalize to the subsequent volatile regime.

Agent Detection and Correction. A standard LLM might generate a hallucinated justification for a "bull market" trend. However, TSOrchestra's faithfulness mechanism triggers a causal decomposition:

1. **Discrepancy Detection:** The agent observes that while the trend-focused model performs well numerically, its SHAP-derived causal contribution to the trend component is near zero, whereas the residual variance is high.
2. **Reasoning Correction:** The agent identifies this pattern as a "Mirage Trend"—spurious correlation driven by volatility noise rather than a systematic signal.
3. **Optimization Adjustment:** Recognizing the regime ambiguity, the agent rejects the trend-heavy weights. It instead directs the optimization to prioritize metrics robust to outliers (e.g., sMAPE) or shifts weight to models specialized in local adaptability (e.g., Toto).

This case demonstrates that the agent's contribution extends beyond metric optimization; it acts as a robustness filter that identifies and corrects brittle numerical solutions in noisy, non-stationary environments.

Systematic Base Model Failure: TSOrchestra's performance is ultimately bounded by the collective capabilities of its candidate pool. If all foundation models in the pool fundamentally fail to capture a novel regime (e.g., an unprecedented black swan event or a structural break not represented in any model's pre-training data), the orchestrated ensemble will also underperform. The agent can optimally route and weight existing capabilities, but it cannot generate new predictive mechanisms outside the candidate pool's scope.

Hyper-Volatile Regimes: In environments where regime shifts occur at a frequency faster than the sliding window can meaningfully capture (e.g., extreme high-frequency noise), the agent's forward-looking assumptions may lag. Because the agent relies on auditing historical optimization traces

to anticipate future incompatibility, highly stochastic environments without underlying structural patterns can diminish the effectiveness of its multi-turn reasoning.

E Main Theorem: Incompatibility-Aware Ensemble Advantage

Theoretical Role. The index quantifies the degree of regime heterogeneity in a time series. The theoretical advantage of an ensemble over a monolithic model, $\Omega(I_T, M)$, scales logistically with I_T. This explains our empirical observations: domains with high regime shifts (e.g., *Sales*, *Finance*) exhibit high implicit I_T and consequently see the largest gains from TSOrchestra's dynamic weighting, whereas stable domains see more moderate improvements.

Limitation and Empirical Proxy. We note that computing exact dataset-level $I_T(\mathcal{D})$ estimates is generally intractable for finite, noisy real-world time series. Therefore, $I_T(\mathcal{D})$ serves as a formal theoretical motivation rather than an operationalized metric computed during inference. In practice, the sliding-window SLSQP optimization combined with LLM auditing acts as our empirical proxy, implicitly adapting to the underlying I_T shifts without requiring explicit calculation.

Let $\mathcal{D} = \{(x_t, y_t)\}_{t=1}^{T}$ represent a time series dataset and let the **Temporal Incompatibility Index** $I_T(\mathcal{D})$ be defined as:

$$I_T(\mathcal{D}) = \frac{1}{T^2} \sum_{t_1, t_2} \mathbb{1}\left\{ \|x_{t_1} - x_{t_2}\| < \delta \ \wedge \ \|f(x_{t_1}) - f(x_{t_2})\| > \frac{\varepsilon}{\delta} \right\}, \tag{12}$$

where $\delta > 0$ is a local neighborhood radius and $\varepsilon > 0$ is the target output separation threshold.

Theorem E.1 (Ensemble Superiority Under Temporal Incompatibility)**.** *Let $L_{monolithic}(\mathcal{D})$ denote the generalization error of a global predictive model, and $L_{ensemble}(\mathcal{D})$ denote the generalization error of an ensemble of M diverse local models. Then, for time series datasets exhibiting substantial temporal incompatibility:*

$$L_{ensemble}(\mathcal{D}) \leq L_{monolithic}(\mathcal{D}) - \Omega(I_T(\mathcal{D}), M),$$

where

$$\Omega(I_T, M) = \frac{I_T \cdot \log M}{1 + \exp(-\kappa \cdot I_T)},$$

and κ is a task-dependent constant reflecting the sharpness of regime boundaries.

Interpretation: The ensemble's advantage increases with (i) stronger structural incompatibility across time and (ii) more diverse model components (M). This effect is especially pronounced in regime-switching time series.

Corollary: Temporal Kolmogorov Decomposition

Let $K(\cdot)$ denote the Kolmogorov complexity. For a regime-heterogeneous time series (X, Y) with regime segmentation $\{(X_i, Y_i)\}$ and mixture weights π_i, we have:

$$K(Y \mid X) \leq \min \left\{ K_{\text{global}}(Y \mid X), \sum_i \pi_i \cdot K_{\text{local}}(Y_i \mid X_i) \right\} + \mathcal{O}(\log K),$$

implying that ensemble models approximating local regimes can match or surpass the efficiency of a global model when local Kolmogorov complexities are significantly lower.

Practical Implication: Ensemble Size Lower Bound

To achieve ε-optimal prediction accuracy, the minimum number of ensemble components M^* satisfies:

$$M^* \geq \frac{I_T(\mathcal{D}) \cdot K_{\text{regimes}}}{\varepsilon \cdot (1 - \rho_{\text{regime}})},$$

where:

- K_{regimes} is the total Kolmogorov complexity of all identified regimes,

- ρ_{regime} is the mean inter-regime correlation.

Consider the following:

- $\mathcal{D} = \{(x_t, y_t)\}_{t=1}^{T}$ is a time series dataset.
- Let $I_T(\mathcal{D})$ quantify the average frequency with which temporally local, structurally similar inputs yield highly dissimilar outputs (regime incompatibility).
- $L_{\text{monolithic}}(\mathcal{D})$ is the generalization error of a global (single) model f_θ fit over $\mathcal{D}$.
- $L_{\text{ensemble}}(\mathcal{D})$ is the generalization error of an ensemble of M models $\{f_{\theta_m}\}$, with each model specializing in a regime.

Step 1. Limitation of Global Model When $I_T(\mathcal{D}) > 0$, there exist pairs (x_{t_1}, x_{t_2}) such that:

$$\|x_{t_1} - x_{t_2}\| < \delta \quad \text{and} \quad |y_{t_1} - y_{t_2}| > \frac{\varepsilon}{\delta}$$

A global model must fit:

$$f_\theta(x_{t_1}) \approx y_{t_1}, \quad f_\theta(x_{t_2}) \approx y_{t_2}$$

but since $x_{t_1} \approx x_{t_2}$ and y_{t_1}, y_{t_2} are far apart, any continuous or well-regularized f_θ will be sub-optimal for at least one of the points.

Thus,

$$\mathbb{E}_{(t_1,t_2)}\left[\mathbb{1}\left\{\|x_{t_1} - x_{t_2}\| < \delta,\, |f_\theta(x_{t_1}) - f_\theta(x_{t_2})| < \frac{\varepsilon}{\delta}\right\}\right] \geq 1 - I_T(\mathcal{D}),$$

which means the model necessarily incurs high loss due to incompatible targets in locally similar regions.

Step 2. Ensemble Model Advantage Suppose we cluster $\mathcal{D}$ into M disjoint regimes $\mathcal{D}_1, \ldots, \mathcal{D}_M$ such that each regime is locally compatible, that is, within each regime, $I_T(\mathcal{D}_m) \approx 0$.

Assign a specialized model f_m to each regime. Then, in each, prediction is consistent:

$$|x_{t_1} - x_{t_2}| < \delta \implies |y_{t_1} - y_{t_2}| < \frac{\varepsilon}{\delta}$$

This allows each f_m to fit its local regime with low error.

Therefore, the total generalization error is reduced as:

$$L_{\text{ensemble}}(\mathcal{D}) = \sum_{m=1}^{M} \pi_m L(f_m, \mathcal{D}_m) + \text{switch cost}$$

where π_m is the regime prior, and "switch cost" is negligible if regimes are well-separated.

Step 3. Quantifying the Gap The ensemble advantage increases with $I_T(\mathcal{D})$ (the degree of incompatibility), since the limitation of the global model comes directly from the need to interpolate conflicting targets.

As M (number of regimes/ensemble members) increases, the coverage of local compatibility improves, with diminishing returns as redundancy grows.

Thus, one can formalize:

$$L_{\text{ensemble}}(\mathcal{D}) \leq L_{\text{monolithic}}(\mathcal{D}) - \Omega(I_T(\mathcal{D}), M)$$

where

$$\Omega(I_T, M) = \frac{I_T \cdot \log M}{1 + \exp(-\kappa \cdot I_T)}$$

and κ is a problem-dependent constant.

Step 4. Kolmogorov Complexity Decomposition (Corollary Proof Sketch) Assume $K(Y|X)$ is the conditional Kolmogorov complexity of the output sequence given inputs.

If a global model f_{global} must explain all inconsistencies (high I_T), its K_{global} is large. If we can decompose data into M regimes with K_{local} each, and regimes are separable, then:

$$K(Y|X) \leq \min \left\{ K_{\text{global}}(Y|X), \sum_{i=1}^{M} \pi_i K_{\text{local}}(Y_i|X_i) \right\} + O(\log K)$$

Hence, if $\sum \pi_i K_{\text{local}} \ll K_{\text{global}}$, ensemble/localized modeling is provably superior.

□

Conclusion: In time series with significant regime shifts or local inconsistencies, ensemble learning is not just a heuristic—it yields provable advantages rooted in information-theoretic properties of the data.

F Detailed Explanation Framework

F.1 Detailed Faithfulness Score Computation

The faithfulness score validates LLM explanations against ground-truth importance measures derived from SHAP analysis. This ensures explanations accurately reflect actual model contributions rather than generating plausible-sounding but incorrect justifications.

STL-SHAP Decomposition. We first decompose each model's predictions using Seasonal and Trend decomposition using Loess (STL), creating eight subset datasets:

$$\mathcal{S} = \{\emptyset, T, S, R, TS, TR, SR, TSR\} \tag{13}$$

where T represents trend, S seasonality, R residual, and combinations represent additive components. For each subset, we evaluate the model's forecasting performance $f(S)$.

SHAP Value Computation. Following Shapley value theory, we compute the marginal contribution of each component through weighted coalition analysis:

$$\text{SHAP}_i = \sum_{S \subseteq \{T,S,R\} \setminus \{i\}} \frac{|S|!(3 - |S| - 1)!}{3!} [f(S \cup \{i\}) - f(S)] \tag{14}$$

This expands to four weighted terms per component with weights $(1/3, 1/6, 1/6, 1/3)$:

For Trend:

$$\begin{aligned} \text{Trend}_{\text{SHAP}} = \frac{1}{3}[f(T) - f(\emptyset)] + \frac{1}{6}[f(TS) - f(S)] \\ + \frac{1}{6}[f(TR) - f(R)] + \frac{1}{3}[f(TSR) - f(SR)] \end{aligned} \tag{15}$$

Similar decompositions apply for Seasonality and Residual components. The Error term captures approximation residuals:

$$\text{Error} = f(TSR) - f(\emptyset) - \text{Trend}_{\text{SHAP}} - \text{Seasonality}_{\text{SHAP}} - \text{Residual}_{\text{SHAP}} \tag{16}$$

From an attribution perspective, the STL-SHAP decomposition provides a structured way to assess how different temporal components are associated with forecasting performance. We decompose each time series into temporal concepts $\mathcal{C} = \{\text{Trend}, \text{Seasonality}, \text{Residual}\}$, and construct component-ablated variants by retaining only subsets $S \subseteq \mathcal{C}$. For each subset S, we evaluate the forecasting performance $f(S)$ of the ensemble on the corresponding ablated series. Since the evaluation metrics used in this paper are lower-is-better, we define a utility score $u(S) = -f(S)$, so that larger values indicate better forecasting performance.

We then compute a Shapley-style component attribution score for each temporal concept $i \in \mathcal{C}$:

$$\phi_i = \sum_{S \subseteq \mathcal{C} \setminus \{i\}} \frac{|S|!(|\mathcal{C}| - |S| - 1)!}{|\mathcal{C}|!} \left[u(S \cup \{i\}) - u(S) \right]. \tag{17}$$

Here, ϕ_i measures the marginal association between component i and forecasting utility under the STL-based component ablation protocol. This score should be interpreted as a decomposition-based attribution signal rather than a formal causal effect. We use the resulting attribution vector $\boldsymbol{\phi} = [\phi_{\text{Trend}}, \phi_{\text{Seasonality}}, \phi_{\text{Residual}}]$ as a reference signal for faithfulness training, encouraging the LLM's explanation-implied component importance to align with the temporal components that are most strongly associated with observed forecasting performance.

Interpretation of SHAP Values. For "lower-is-better" metrics (MASE, SMAPE):

- Negative SHAP value: Component improves forecast (reduces error)
- Positive SHAP value: Component worsens forecast (increases error)
- Magnitude indicates importance regardless of direction

Unfaithfulness Pattern Detection. The system identifies specific unfaithfulness patterns:

$$\text{Unfaithful}(m) = \begin{cases} \text{"Overstatement"} & \text{if LLM claims high importance but } |\text{SHAP}_m| < \tau_{\text{low}} \\ \text{"Understatement"} & \text{if LLM claims low importance but } |\text{SHAP}_m| > \tau_{\text{high}} \\ \text{"Wrong direction"} & \text{if LLM claims benefit but } \text{SHAP}_m > 0 \\ \text{"Missed pattern"} & \text{if seasonal SHAP high but LLM ignores seasonality} \end{cases} \tag{18}$$

Example Faithfulness Analysis. For dataset `ett1_short` with three models:

SHAP Values (normalized — raw):

- Trend: 0.521 (-0.234) [StatisticalEnsemble dominant]
- Seasonality: 0.312 (-0.140) [Moirai secondary]
- Residual: 0.167 (-0.075) [TiDE minor]

LLM Explanation Extract: "StatisticalEnsemble receives highest weight (45%) due to strong trend capture... Moirai complements with seasonal patterns (28%)..."

Faithfulness Score: 0.87

- Rank alignment: 0.90 (correct ordering)
- Magnitude alignment: 0.85 (appropriate emphasis)
- Pattern recognition: 0.86 (identified trend>seasonal>residual)

Unfaithfulness Patterns: None detected

This comprehensive faithfulness framework ensures that LLM explanations remain grounded in empirical evidence, preventing the system from generating sophisticated-sounding but factually incorrect justifications for ensemble weights.

F.2 LLM Faithfulness Validation

The faithfulness analyzer validates that LLM explanations correctly identify relationships rather than spurious correlations. Using SHAP values as ground-truth attribution effects, we assess whether the LLM's claimed importance aligns with actual attribution contributions.

The system performs attribution-aligned grounding testing: when the LLM claims "Moirai excels due to strong seasonal patterns," we verify that $|\text{SHAP}_{\text{seasonality}}(\text{Moirai})| > \tau$, ensuring the explanation reflects genuine attribution influence rather than post-hoc rationalization.

We detect specific patterns of attribution misrepresentation:

- **Temporal bias hiding**: LLM attributes performance to technical features while SHAP reveals trend dominance as the true attribution factor
- **Spurious pattern claims**: LLM cites seasonality without corresponding negative SHAP values indicating causal improvement
- **Attribution direction errors**: LLM claims a component helps when SHAP shows its attribution increases error

This mechanism ensures our system generates explanations that are not merely plausible but attribution-consistent about why certain models receive higher ensemble weights.

G Details on R1-style finetuning

G.1 Task Formulation and Dataset Construction

The model receives structured input containing cross-validation metrics (MAE, MSE, RMSE, SMAPE, MASE), SLSQP-optimized weights, model rankings, and iteration history. It outputs a JSON decision with: (i) `confidence` $\in [0, 1]$ indicating calibration, (ii) `should_continue` boolean for accept/continue decision, and (iii) `explanation` providing evidence-grounded reasoning.

This structured approach offers several advantages demonstrated in recent work (Guo et al., 2025): it improves interpretability by making the reasoning process explicit, reduces hallucination by grounding decisions in analyzed evidence, and achieves more concise final outputs.

Example output format:

```
<think>
Current MAE: 0.0234, Best MAE: 0.0231 (margin: 0.0003)
Weight distribution shows high concentration on LSTM (0.45)
Convergence: 3 iterations without improvement
Pattern: Oscillating between local minima
</think>
<decision>
{
  "confidence": 0.82,
  "should_continue": false,
  "explanation": "Accept current weights. Marginal difference
   (0.0003) below tolerance with stable convergence after 3
   iterations. LSTM dominance (45%) aligns with data's strong
   temporal dependencies."
}
</decision>
```

We create training data from ensemble optimization trajectories using a ground-truth labeling policy. Given current score S_{current} and best score S_{best}, with margin $\Delta = S_{\text{current}} - S_{\text{best}}$ and tolerance $\tau \geq 0$:

$$G = \begin{cases} 1 \text{ (Accept)} & \text{if } \Delta \leq \tau \text{ or iteration} \geq 3 \\ 0 \text{ (Continue)} & \text{otherwise} \end{cases} \tag{19}$$

This policy balances optimization quality (accepting only improvements within tolerance) with computational efficiency (forcing termination after 3 iterations). The tolerance τ is dynamically set based on the metric scale and optimization stage, typically $\tau = 0.001 \times S_{\text{best}}$ for normalized metrics.

G.1.1 Reinforcement Learning with GRPO

To refine decision-making beyond supervised imitation, we apply Group Relative Policy Optimization (GRPO) (Guo et al., 2025), an actor-only variant of PPO that estimates advantages through group-relative comparisons. This approach has demonstrated remarkable success in specialized reasoning tasks, with recent work showing GRPO-trained 3B models outperforming 671B alternatives on temporal reasoning.

Reward Design. Inspired by Time-R1's comprehensive reward mechanism that combines task-specific accuracy with consistency penalties, we design a multi-component reward function. For each prompt, we sample $n = 4$ responses and compute rewards using:

$$r = \text{clip}_{[0,1]} \left(r_{\text{base}} + \alpha \cdot r_{\text{conf}} + \beta \cdot r_{\text{faith}} \right) \tag{20}$$

where the reward components are:

- $r_{\text{base}} = \mathbb{1}[\hat{y}_{\text{acc}} = G]$: Primary reward for correct accept/continue decisions
- $r_{\text{conf}} = \min(c, 1) \cdot \mathbb{1}[\text{correct}] - \max(c - 0.5, 0) \cdot \mathbb{1}[\text{incorrect}]$: Calibration reward that encourages high confidence on correct decisions and penalizes overconfidence on errors
- $r_{\text{faith}} = \mathcal{F} \cdot \mathbb{1}[\text{valid_json}]$: Faithfulness score from Equation 5, weighted by format validity

with coefficients $\alpha = 0.2$, $\beta = 0.1$ determined through ablation studies.

Optimization Objective. The GRPO objective optimizes policy π_θ while maintaining proximity to the SFT checkpoint π_{ref} through KL regularization:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[\hat{A}(y, x) \cdot \log \pi_\theta(y|x) \right] \right] + \lambda \cdot \mathbb{E}_{x \sim \mathcal{D}} \left[\text{KL}(\pi_\theta(\cdot|x) || \pi_{\text{ref}}(\cdot|x)) \right] \tag{21}$$

where the group-normalized advantage $\hat{A}(y, x)$ is computed as:

$$\hat{A}(y, x) = \frac{r(y, x) - \bar{r}(x)}{\sigma_r(x) + \epsilon} \tag{22}$$

with $\bar{r}(x) = \frac{1}{n} \sum_{i=1}^{n} r(y_i, x)$ being the mean reward across n sampled responses, $\sigma_r(x)$ the standard deviation, and $\epsilon = 10^{-8}$ for numerical stability. The KL coefficient $\lambda = 10^{-3}$ prevents catastrophic forgetting of SFT capabilities while allowing policy improvement.

G.2 Training Data Generation and Implementation Details

To ensure the reproducibility and transparency of our R1-style fine-tuning pipeline, we provide the full specification of our synthetic trajectory generation process and the subsequent training configuration.

G.2.1 Data Generation Pipeline

Our training dataset, comprising over 2,800 trajectories, was constructed via a four-stage synthetic pipeline. This approach eliminates the need for manual human annotation while ensuring ground-truth validity:

1. **Exhaustive Optimization Simulation:** We systematically executed the SLSQP optimization algorithm exclusively on the pre-training split of the GIFT-Eval benchmark to map the complete optimization landscape. To definitively prevent any data leakage or contamination, the agent is trained strictly on these synthetic optimization trajectories and never ingests raw time series values at any point; instead, its inputs consist purely of abstract numerical metrics, model rankings, and SHAP scores.
2. **Ground Truth Determination:** For every optimization state $S_t = (w_t, \mathcal{H}_t, \mathcal{M}_{data})$, we determined the optimal next action a^* by:
   - Executing SLSQP with each candidate metric.
   - Evaluating the resulting weights on a held-out validation split.
   - Labeling the metric that yielded the lowest validation error as the ground-truth decision.
3. **Reasoning Trace Synthesis:** We employed a teacher model (GPT-4o) to synthesize semantic reasoning traces. The teacher was prompted with the numerical ground truth (e.g., "sMAPE optimization yields 15% lower error") and SHAP-derived dataset characteristics (e.g., "40%

seasonal variance detected"). This ensures the reasoning causally links data characteristics to the optimization decision.

4. **Faithfulness Filtering:** To prevent hallucinations, all generated traces were filtered using our Faithfulness Score. Only trajectories where the alignment between the text explanation and SHAP attribution effects exceeded a threshold ($\tau > 0.8$) were retained for training.

G.2.2 Dataset Composition and Training Schedule

Dataset Split. The generated trajectories were curated into a specialized dataset comprising 2,000 SFT (Supervised Fine-Tuning) samples and 800 GRPO (Generative Reinforcement from Policy Optimization) samples. The SFT samples contain diverse ensemble optimization scenarios with ground-truth weight configurations to teach foundational reasoning patterns. The GRPO samples include preference pairs comparing successful and unsuccessful weight selections across different forecasting contexts to refine decision boundaries.

Training Strategy. To prevent overfitting while ensuring effective reasoning transfer, we employed a carefully calibrated training schedule:

- Phase 1 (SFT): 500 steps to establish foundational reasoning chains.
- Phase 2 (GRPO): 50 steps for preference alignment.

This conservative GRPO schedule—significantly shorter than the SFT phase (a 10:1 ratio)—prevents the model from overfitting to specific weight patterns while reinforcing general optimization principles. This strategy proved highly effective: despite the relatively small dataset size compared to typical LLM training, TSOrchestra achieves strong generalization across unseen forecasting domains, consistent with observations that specialized reasoning tasks require less data than general knowledge acquisition.

G.3 Training configuration.

Table 4 summarizes the training configuration used for the R1-style auditor. The SFT stage teaches the structured JSON decision format and basic stop/continue calibration, while GRPO further optimizes decision correctness, confidence calibration, and decomposition-aligned faithfulness. All GRPO updates are initialized from the SFT checkpoint and regularized by a KL penalty to avoid catastrophic forgetting.

Table 4: R1-style fine-tuning configuration.

Item	Value	Item	Value
Base auditor	Qwen-2.5-3B-I	Max opt. rounds	3
Training trajectories	$> 2,800$	Accept threshold	$\Delta \leq \tau$ or iter. ≥ 3
SFT samples	$2,000$	Normalized tolerance	$\tau = 0.001 \times S_{\text{best}}$
GRPO samples	800	KL coefficient	10^{-3}
SFT steps	500	Stability constant	10^{-8}
GRPO steps	50	Reward components	$r_{\text{base}}, r_{\text{conf}}, r_{\text{faith}}$
GRPO rollouts	4	Reward weights	$\alpha = 0.2, \beta = 0.1$

We emphasize that the GRPO policy does not generate continuous ensemble weights. Its action space consists of discrete accept/continue decisions and metric-routing choices; SLSQP remains responsible for continuous constrained weight optimization.

G.4 Accuracy on Agent Selection

For sample i in domain d, we compute:

$$y_i^* = \arg \min_{m \in \{\text{MSE,MAE,SMAPE}\}} L_m(w_m, X_i) \tag{23}$$

where $L_m(w_m, X_i)$ represents the loss when using weights optimized solely for metric m on sample X_i. This provides supervised labels indicating which metric yields optimal performance for each

Table 5: Comparison of cross-validation and test set performance across different datasets and ensembles.

Dataset	Series ID	Optimized Metric	Ensemble Weights		Cross-validation CRPS/MASE	Test CRPS/MASE	% Difference CRPS/MASE
Bizitobs_l2c Hourly	uid1	sMAPE	**Toto**:	0.000	0.733/1.160	0.529/0.829	27.851/28.508
			Moirai:	0.937			
			Sundial:	0.063			
	uid1	MSE	**Toto**:	0.024	0.728/1.150	0.529/0.832	27.348/27.690
			Moirai:	0.976			
			Sundial:	0.000			
	uid1	MAE	**Toto**:	0.015	0.725/1.147	0.530/0.831	26.909/27.542
			Moirai:	0.985			
			Sundial:	0.000			
ETT1 15 minutes	uid1	sMAPE	**Toto**:	0.069	0.230/1.377	0.224/1.060	2.761/23.035
			Moirai:	0.693			
			Sundial:	0.230			
	uid2	MSE	**Toto**:	0.166	0.255/1.170	0.182/0.924	28.683/20.972
			Moirai:	0.239			
			Sundial:	0.595			
Jena Weather 10 minutes	uid1	sMAPE	**Toto**:	0.130	1.166/1.251	1.019/0.877	12.623/29.891
			Moirai:	0.006			
			Sundial:	0.864			
	uid1	MAE	**Toto**:	0.689	1.10/1.181	1.012/0.862	8.068/27.046
			Moirai:	0.084			
			Sundial:	0.226			

time series pattern. We then evaluate each strategy's ability to predict y_i^*:

$$\text{Accuracy}_s = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{y}_i^s = y_i^*] \tag{24}$$

where $\hat{y}_i^s$ is the metric selected by strategy s (e.g., LLM confidence, SFT-enhanced selection).

H Cross-validation vs Test Performance

We provide series-level comparisons in cross-validation and test set performance across ensembles that optimize different metrics in Table 5. These results highlight specific cases where optimizing for a given metric during cross-validation leads to improved generalization on the test set. Optimizing sMAPE during cross-validation often resulted in better generalization than optimizing for MSE or MAE. However, when optimizing for MSE or MAE yielded improvements in CRPS and MASE, performance gains were often comparable to those achieved by optimizing for sMAPE.

I Qualitative Case Study: Expert-Aligned Reasoning Analysis

To bridge numerical results with semantic intuition, we examine the agent's optimization trajectory for the *BizitObs Service* (trend: 0.869) as detailed in Frame K(Title: TSOrchestra: LLM-Guided Weight Optimization Demo). The agent automates expert logic by prioritizing Sundial for its trend-capturing strengths while decisively excluding Toto, which exhibited a prohibitive error magnitude due to its architectural focus on local patterns rather than global trends. Unlike one-shot numerical solvers, the agent maintains cross-metric stability, ultimately reverting to the highest-confidence configuration (87%) after verifying that SMAPE-driven shifts lacked decomposition-aligned grounding. This process transforms opaque ensemble weighting into a semantically sound and auditable reasoning trail.

J Limitations

Despite the strong empirical performance of TSOrchestra, our framework has several limitations that provide avenues for future research. Fundamentally, the system's efficacy is bounded by the collective capabilities of the foundation models in its candidate pool; if an unprecedented structural break falls outside their predictive scope, the ensemble will underperform since the agent strictly routes existing capabilities rather than generating novel mechanisms. Additionally, in hyper-volatile

environments where underlying distributions shift faster than the sliding window can capture, the agent's reliance on historical optimization trails may result in response lag. Computationally, the multi-turn LLM reasoning introduces a latency overhead of approximately 3 to 4 seconds per forecasting window, precluding its deployment in ultra-low latency applications such as high-frequency trading. Furthermore, although the generated natural language audit trails are quantitatively validated for faithfulness, their practical utility for practitioner comprehension and decision-making remains to be thoroughly validated through large-scale human-in-the-loop studies. Finally, because calculating the exact dataset-level temporal incompatibility index ($I_T(\mathcal{D})$) is generally intractable for finite, noisy real-world data, it serves primarily as a formal theoretical motivation rather than a metric directly computed during inference.

K Impact Statement

This paper presents a framework for explainable time series forecasting by positioning Large Language Models (LLMs) as metacognitive auditors over specialized numerical models. The primary goal of this work is to *advance the transparency and reliability* of predictive systems in high-stakes domains such as energy management, financial planning, and cloud operations.

This research does not involve human subjects, personal data, or sensitive user interaction logs. All experiments are conducted on public, anonymized benchmark datasets (GIFT-Eval) using open-source or commercially available models. We do not identify additional negative societal consequences beyond those commonly associated with the general advancement of automated forecasting and the computational costs inherent in large-scale model orchestration.

TSOrchestra: LLM-Guided Weight Optimization Demo

Dataset: ETT1/H (Hourly electricity data, long-term forecasting)
Models: Moirai (transformer), Sundial (trend+seasonal), Toto (local patterns)

ITERATION PROCESS

ROUND 1: INITIAL EQUAL WEIGHTS

- **Weights:** [0.33, 0.33, 0.33] → MSE: 84,124
- **LLM Analysis:** "Equal weights ignore Sundial's 2x better performance"
- **Decision:** Continue with MAE optimization ✓

ROUND 2: PERFORMANCE-BASED ADJUSTMENT

- **Weights:** [0.28, 0.72, 0.00] → MAE: 8.16
- **LLM Analysis:** "Better alignment but Moirai underweighted given seasonal patterns"
- **Decision:** Try SMAPE for nuanced evaluation ✓

ROUND 3: REFINED OPTIMIZATION

- **Weights:** [0.31, 0.69, 0.00] → SMAPE: 0.57
- **LLM Confidence:** 82.5%
- **Key Insight:** "Sundial dominates due to superior trend+seasonal handling. Toto correctly excluded (12x worse MASE). Accept."

OPTIMIZATION JOURNEY (3 ITERATIONS)

- **Start:** Equal weights [0.33, 0.33, 0.33] → Poor MSE: 84,124
- **Final:** Performance-based [Moirai: 0.27, Sundial: 0.73, Toto: 0.00] → SMAPE: 0.57
- **Confidence:** 82.5%

KEY INSIGHTS

- **Pattern Discovery:** System automatically identified Sundial as the dominant performer (69% weight) for trend+seasonal patterns.
- **Model Exclusion:** Correctly eliminated Toto (0% weight) due to 12x worse performance.
- **Metric Evolution:** MSE → MAE → SMAPE progression refined weight optimization.
- **Anti-Pattern Detection:** Rejected initial equal weights as suboptimal given 2x performance gaps.

RESULT QUALITY

- ✓ Weights align with performance rankings (Sundial > Moirai ≫ Toto)
- ✓ Dataset characteristics (long-term electricity) match model strengths
- ✓ Clear convergence from naive to optimized ensemble
- ✓ Interpretable decisions throughout optimization process

Outcome: Well-grounded ensemble configuration leveraging complementary model strengths while excluding underperformers.

TSOrchestra: LLM-Guided Weight Optimization Demo

Dataset: BizitObs Service (Business operations, long-term forecasting)
Models: Moirai (transformer), Sundial (trend+seasonal), Toto (local patterns)

ITERATION PROCESS

ROUND 1: INITIAL EQUAL WEIGHTS

- **Weights:** [0.33, 0.33, 0.33] → MSE: 772,199
- **LLM Analysis:** "Equal weights ignore Sundial's 2x better performance"
- **Confidence:** 60%
- **Decision:** Continue with MAE optimization ✓

ROUND 2: PERFORMANCE-BASED ADJUSTMENT

- **Weights:** [0.40, 0.60, 0.00] → MAE: 18.61
- **LLM Analysis:** "Sundial correctly prioritized for trend-heavy data. Toto eliminated (3,400x worse MSE)"
- **Confidence:** 87% **BEST**
- **Decision:** Target confidence achieved, verify with SMAPE ✓

ROUND 3: METRIC VALIDATION

- **Weights:** [0.60, 0.40, 0.00] → SMAPE: 0.069
- **LLM Analysis:** "Weight shift reflects SMAPE's different error perspective"
- **Confidence:** 82%
- **Decision:** Use Round 2 weights (highest confidence) ✓

KEY INSIGHTS

- **Pattern Discovery:** System automatically identified Sundial as the dominant performer (60% weight) for trend-heavy business data.
- **Metric Evolution:** MSE → MAE → SMAPE progression refined weight optimization, with MAE proving optimal.
- **Anti-Pattern Detection:** Rejected initial equal weights as suboptimal given 2x performance gaps.

RESULT QUALITY

- ✓ **Weights align with performance rankings** (Sundial > Moirai ≫ Toto)
- ✓ **Dataset characteristics match model strengths** (strong trend: 0.869)
- ✓ **Clear convergence from naive to optimized ensemble** (60% → 87% confidence)
- ✓ **Interpretable decisions throughout optimization** process

FINAL ASSESSMENT

Outcome: Successfully converged at 87% confidence with performance-aligned weights. The ensemble prioritizes trend-capturing models while excluding underperformers, demonstrating the system's ability to automatically discover optimal configurations without manual intervention.
Dataset Fit: Strong trend component (0.869) and weak seasonality perfectly match the selected model distribution, validating the automatic specialization discovery.

TSOrchestra: LLM-Guided Weight Optimization Demo

Dataset: saugeenday/D (short)
Models: Moirai, Sundial, Toto

ITERATION PROCESS

ROUND 1

- **Weights:** [Moirai: 0.83, Sundial: 0.17, Toto: 0.00] → MSE: 1,007
- **LLM Analysis:** "The weights assigned (Moirai: 0.8313, Sundial: 0.1687, Toto: 0.0000) reflect a strong alignment with the models' performance metrics."
- **Decision:** STOP ✓**BEST** ✓
- **Confidence:** 92.0%

KEY INSIGHTS

- **Pattern Discovery:** System identified **Moirai** as dominant (83% weight).
- **Model Exclusion:** Eliminated Toto (0% weight).
- **Metric Evolution:** MSE progression refined weight optimization.

RESULT QUALITY

- ✓ **Weights align with performance-driven selection**
- ✓ **Clear convergence from naive to optimized ensemble**
- ✓ **Interpretable decisions throughout optimization process**

FINAL ASSESSMENT

Outcome: Well-grounded ensemble configuration leveraging complementary model strengths while excluding underperformers.

TSOrchestra: LLM-Guided Weight Optimization Demo

Dataset: bitbrains_rnd/5T (medium)
Models: Moirai, Sundial, Toto-q-50

ITERATION PROCESS

ROUND 1

- **Weights:** [Moirai: 0.36, Sundial: 0.28, Toto-q-50: 0.36] → MSE: 1,852,839
- **LLM Analysis:** "The SLSQP-optimized weights show a distribution of [0.3626, 0.2810, 0.3564] for Moirai, Sundial, and Toto respectively."
- **Decision:** CONTINUE ✓**BEST** ✓
- **Confidence:** 75.0%

ROUND 2: PERFORMANCE-BASED ADJUSTMENT

- **Weights:** [Moirai: 0.31, Sundial: 0.00, Toto-q-50: 0.69] → MAE: 141.122
- **LLM Analysis:** "The SLSQP-optimized weights of [0.3074, 0.0000, 0.6926] do not align well with the performance metrics of the models."
- **Decision:** CONTINUE ✓
- **Confidence:** 72.0%

ROUND 3: REFINED OPTIMIZATION

- **Weights:** [Moirai: 0.46, Sundial: 0.02, Toto-q-50: 0.53] → SMAPE: 0.737
- **LLM Analysis:** "The SLSQP-optimized weights indicate a notable preference for Toto-q-50 (52.9%) and Moirai (45.5%), with very minimal weight for Sundial (1.7%)."
- **Decision:** STOP ✓
- **Confidence:** 75.0%

KEY INSIGHTS

- **Pattern Discovery:** System identified **Moirai** as dominant (36% weight).
- **Metric Evolution:** MSE → MAE → SMAPE progression refined weight optimization.

RESULT QUALITY

- ✓ **Weights align with performance-driven selection**
- ✓ **Clear convergence from naive to optimized ensemble**
- ✓ **Interpretable decisions throughout optimization process**

FINAL ASSESSMENT

Outcome: Well-grounded ensemble configuration leveraging complementary model strengths while excluding underperformers.

TSOrchestra: LLM-Guided Weight Optimization Demo

Dataset: jena_weather/10T (short)
Models: Moirai, Sundial, Toto-q-50

ITERATION PROCESS

ROUND 1

- **Weights:** [Moirai: 0.00, Sundial: 0.80, Toto-q-50: 0.20] → MSE: 1,045
- **LLM Analysis:** "The SLSQP-optimized weights assigned are highly concentrated with Sundial receiving 80.1% of the weight, whereas Moirai, despite its strengths, is assigned 0% weight which is surprising given its capabilities in seasonality."
- **Decision:** CONTINUE ✓
- **Confidence:** 65.0%

ROUND 2: PERFORMANCE-BASED ADJUSTMENT

- **Weights:** [Moirai: 0.00, Sundial: 0.30, Toto-q-50: 0.70] → MAE: 6.293
- **LLM Analysis:** "The SLSQP-optimized weights show a significant concentration on Toto-q-50 (70.2%), which corresponds well with its MAE performance being the lowest among the models."
- **Decision:** CONTINUE ✓**BEST** ✓
- **Confidence:** 80.0%

ROUND 3: REFINED OPTIMIZATION

- **Weights:** [Moirai: 0.00, Sundial: 0.70, Toto-q-50: 0.30] → SMAPE: 0.547
- **LLM Analysis:** "The distribution of weights (Moirai: 0.0, Sundial: 0.7044, Toto-q-50: 0.2956) reflects the relative performance metrics well, as both Sundial and Toto-q-50 show solid SMAPE and MAE performance compared to Moirai."
- **Decision:** STOP ✓
- **Confidence:** 78.0%

KEY INSIGHTS

- **Pattern Discovery:** System identified **Toto-q-50** as dominant (70% weight).
- **Model Exclusion:** Eliminated Moirai (0% weight).
- **Metric Evolution:** MSE → MAE → SMAPE progression refined weight optimization.

RESULT QUALITY

- ✓ **Weights align with performance-driven selection**
- ✓ **Clear convergence from naive to optimized ensemble**
- ✓ **Interpretable decisions throughout optimization process**

FINAL ASSESSMENT

Outcome: Well-grounded ensemble configuration leveraging complementary model strengths while excluding underperformers.