SKILL4TSF: CONTRASTIVE SKILL DISTILLATION FOR TIME SERIES FORECASTING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting exhibits substantial heterogeneity across datasets, forecasting horizons, and evolving temporal regimes, making it difficult for any single forecasting model or fixed pipeline to perform reliably in all settings. We propose SKILL4TSF, a skill-based forecasting agent that learns reusable, condition-specific strategies for model selection and forecast refinement. Our key component, Contrastive Skill Distillation (CSD), derives skills from matched comparisons among heterogeneous forecasters. At the model level, CSD identifies which forecaster is preferable under particular time series characteristics using statistically supported performance differences. At the strategy level, it analyzes recurrent residual patterns on hard forecasting windows, proposes targeted interventions, including adaptive preprocessing, component-wise routing, and bias correction, and retains only those validated by paired statistical tests. Each skill explicitly represents its applicability conditions, recommended strategy, empirical evidence, and explanatory rationale. At inference time, Skill4TSF combines feature-based filtering with reasoning-based applicability assessment to retrieve and compose compatible skills for each input window. Experiments on eight real-world benchmarks show that Skill4TSF achieves the lowest average MSE on seven datasets and the lowest average MAE on six, with results averaged over four forecasting horizons. Ablations support the contributions of both distillation levels, while online adaptation and cross-backbone transfer experiments further examine the utility of learned skills. These results support evidence-grounded skill learning as a practical approach to adaptive time series forecasting.

1 INTRODUCTION

Time series forecasting is a fundamental task with broad real-world applications, ranging from finance Arsenault et al. (2025) and transportation Ghosh et al. (2009) to energy systems Nogales et al. (2002) and healthcare Puri et al. (2022). Given a sequence of historical observations, the objective is to predict values over a future time horizon. Early approaches to time series forecasting primarily relied on statistical and mathematical models Shumway & Stoffer (2024). With the rapid advancement of deep learning, neural forecasting methods based on convolutional neural networks (CNNs) and Transformers have gained increasing popularity in recent years Wang et al. (2026). More recently, this trend has further led to the emergence of larger-scale forecasting models, including foundation models Das et al. (2024) and large language model (LLM)-based approaches Jin et al. (2024).

Although these models have achieved promising results on their respective evaluation datasets, extensive benchmarking studies Aksu et al. (2024); Qiu et al. (2024) have shown that no single model consistently outperforms all others across datasets and application scenarios. In some cases, heavily parameterized models may even perform worse than simple linear baselines Zeng et al. (2023). Moreover, even within the same task or deployment setting, the underlying data distribution and temporal patterns may evolve over time, consequently changing the most appropriate preprocessing strategies and forecasting models. In contrast to conventional approaches that directly fit a predictive model to temporal dependencies, agent-based methods Tao et al. (2026b); Zhao et al. (2025) formulate time series forecasting as a sequential decision-making problem. Rather than relying on a fixed forecasting architecture, an agent dynamically identifies the most suitable processing strategy for the input time series and selects an appropriate forecasting model accordingly.

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.

Recent research on agents has increasingly highlighted the role of skills Zhou et al. (2026). A skill is a structured document that encapsulates procedural steps for problem solving as well as domain-specific knowledge, thereby guiding an agent's behavior Li et al. (2026). Skills offer a learning mechanism beyond parameter updates: acquired knowledge can influence agent execution in a human-readable and editable form. Prior work has demonstrated that skills learned from datasets can substantially improve agent performance Ni et al. (2026); Yang et al. (2026). However, existing work on skill learning and evolution has largely focused on tasks in natural language processing. These approaches typically induce skills empirically from execution trajectories and aggregate performance scores alone. As a result, they are ill-suited to time series forecasting, where an agent must reason about the relative benefits of alternative strategies, their applicable conditions, and their dependencies on particular forecasting models. Moreover, such approaches provide limited guidance for making targeted, evidence-based revisions to individual strategies.

In this paper, we propose Skill4TSF, a skill-based agent specifically designed for time series forecasting. Rather than distilling general guidance from execution outcomes alone, Skill4TSF learns condition-specific forecasting strategies through Contrastive Skill Distillation (CSD). CSD operates at two complementary levels: model-level comparisons identify which forecaster to prefer under particular data conditions, while hypothesis-driven interventions diagnose recurrent forecasting errors and evaluate targeted improvements beyond model selection, such as adaptive preprocessing, component-wise routing, and bias correction. Candidate skills are evaluated across multiple windows, grounding strategy recommendations in comparative evidence rather than isolated successes. Each skill explicitly records its applicability conditions, forecasting strategy, empirical evidence, and explanatory rationale, making its scope and model dependencies available for inspection and refinement. At inference time, the agent combines statistical filtering with reasoning-based applicability assessment to retrieve relevant skills and compose compatible strategies for the current window. This design connects strategy discovery, validation, and execution, enabling the agent to adapt its forecasting workflow to observed temporal characteristics while providing an explicit basis for targeted strategy revision.

In summary, our main contributions are threefold:

- We introduce Skill4TSF, a skill-based agent tailored to time series forecasting that learns and applies reusable, condition-specific forecasting strategies.
- We propose Contrastive Skill Distillation (CSD), which grounds conditional model selection and model-aware strategy refinement in matched forecasting comparisons.
- Extensive experiments demonstrate Skill4TSF's strong performance, with ablations, robustness analyses, and case studies supporting its component effectiveness, robustness, and interpretability.

2 Related Work

2.1 Time Series Forecasting

Time series forecasting aims to model temporal patterns that link historical observations to future values. Traditional approaches primarily rely on statistical models, such as autoregressive integrated moving average (ARIMA) Shumway & Stoffer (2024) and exponential moving average (EMA) Montgomery et al. (2015). Supported by well-established mathematical principles, these methods generally offer strong interpretability. With the availability of large-scale datasets, deep learning methods Wang et al. (2026) have demonstrated substantial potential for time series forecasting. CNN Wu et al. (2023) and linear models Zeng et al. (2023); Xu et al. (2024) represent relatively simple yet effective model families for time series forecasting. Following the remarkable success of Transformers Vaswani et al. (2017) in sequence modeling, a growing body of research has adapted Transformer-based architectures to time series forecasting, investigating issues such as time series embedding Nie et al. (2023), non-stationarity Liu et al. (2022), autocorrelation Wu et al. (2021), and cross-channel dependencies Liu et al. (2024). More recently, foundation models Das et al. (2024); Shi et al. (2025) and large language model (LLM)-based approaches Jin et al. (2024); Wang et al. (2024) have been introduced, leveraging their large-scale parameterization and enhanced modeling capacity to further improve forecasting performance.

Agent-based approaches offer an alternative paradigm for time series forecasting. These methods leverage the reasoning capabilities of LLMs to make a sequence of decisions throughout the forecasting pipeline. TimeSeriesScientist Zhao et al. (2025) formulates time series forecasting as an end-to-end multi-agent workflow, in which specialized LLM-based agents collaboratively execute the entire forecasting pipeline. MemCast Tao et al. (2026a) organizes training-derived historical patterns, reasoning wisdom, and general temporal laws into a hierarchical memory to guide and iteratively refine LLM-based time series predictions. Cast-R1 Tao et al. (2026b) employs an LLM agent with dynamic memory that adaptively invokes diagnostic and forecasting tools, and is trained through supervised fine-tuning and multi-turn reinforcement learning. In contrast to these approaches, our method is the first skill-based agent framework for time series forecasting. It improves the forecasting capability of LLM agents while maintaining the acquired knowledge in an editable and extensible form, without requiring any modification to the underlying model parameters.

2.2 Agent Skills

Agent skills provide agents with external, natural-language artifacts that guide their problem-solving behavior in challenging tasks and environments Li et al. (2026). Recently, a growing body of work has investigated methods for learning and self-evolving such skills Zhou et al. (2026). CoEvoSkills Zhang et al. (2026) couples a skill generator with an independent surrogate verifier to iteratively evolve reusable, multi-file skill packages. SkillOpt Yang et al. (2026) formulates a natural-language skill document as a trainable external state and employs a separate optimizer model to refine it iteratively. Trace2Skill Ni et al. (2026) distills a large collection of both successful and unsuccessful agent trajectories into a transferable skill repository, hierarchically consolidating recurrent and non-conflicting patterns into unified procedural guidance. SkillClaw Ma et al. (2026) facilitates collective skill evolution by aggregating multi-user agent interaction trajectories into shared, skill-specific evidence, which an autonomous evolver uses to refine existing skills or generate new ones. Unlike general-purpose skill-learning methods that primarily distill procedural guidance from aggregate execution outcomes, our method grounds skill acquisition in contrastive forecasting evidence, explicitly characterizing when a strategy is effective, which forecasting models it depends on, and how it can be revised to address recurring errors.

3 Method

Figure 1 provides an overview of Skill4TSF. The framework consists of a skill acquisition stage and a skill-guided forecasting stage. During skill acquisition, Contrastive Skill Distillation (CSD) extracts reusable forecasting skills from historical windows and comparative evidence across a pool of heterogeneous base forecasters. The resulting skill pool is then frozen and used at test time to guide model selection, strategy composition, and forecast refinement for unseen forecasting windows.

3.1 Skill Representation

A Skill4TSF skill specifies the data conditions under which a forecasting strategy is applicable and the strategy to apply. Each skill ω is represented as a quadruple:

$$\omega = (\mathcal{C}_\omega, \mathcal{S}_\omega, \mathcal{E}_\omega, \mathcal{R}_\omega), \tag{1}$$

where $\mathcal{C}_\omega$ characterizes the applicable data conditions, $\mathcal{S}_\omega$ specifies the forecasting strategy, $\mathcal{E}_\omega$ records empirical evidence, and $\mathcal{R}_\omega$ provides the rationale underlying the strategy.

Condition signature $\mathcal{C}_\omega$. The condition signature combines a symbolic and a semantic view of applicability. Its symbolic component consists of threshold-based predicates over interpretable window-level features, such as trend strength and the ADF test p-value, enabling efficient coarse retrieval. Its semantic component is an LLM-generated description of the scenario, enabling fine-grained reasoning when several skills satisfy similar statistical conditions.

Strategy $\mathcal{S}_\omega$. The strategy space is not limited to a fixed set of model-selection rules. Depending on the diagnosed failure mode, a strategy may specify model preferences, adaptive preprocessing, component-wise routing, or systematic bias correction. Candidate strategies are proposed and distilled through the CSD procedure in Section 3.2.

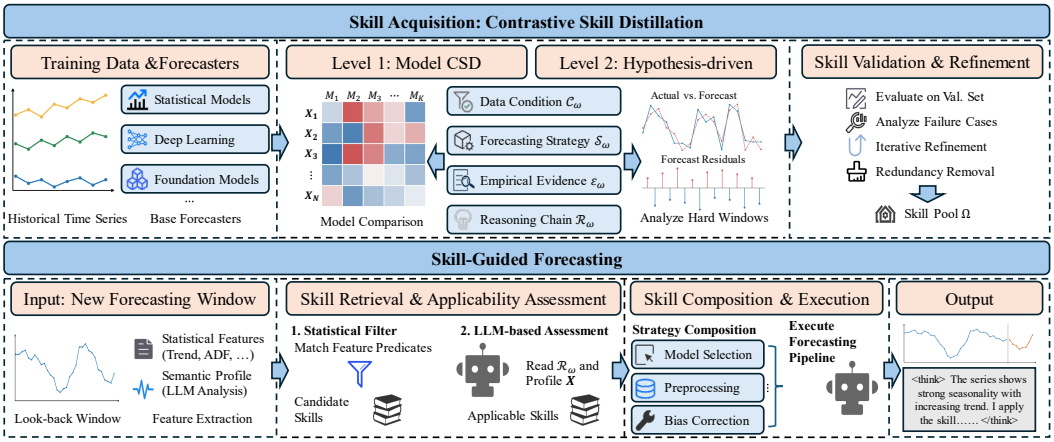


Figure 1: Overview of Skill4TSF. Level 1 CSD distills model-selection skills from comparative forecasting results, while Level 2 distills intervention skills from recurrent failure patterns. At inference time, applicable skills are retrieved, composed, and executed for a new forecasting window.

Evidence record $\mathcal{E}_\omega$. Rather than storing raw application cases, the evidence record maintains a compact three-tier summary. Aggregate statistics (application count, success count, and mean error reduction Δe_ω) are updated for every application and occupy constant space. A bounded buffer retains the most recent failure analyses produced by the LLM. These are used during iterative refinement to diagnose whether condition boundaries should be tightened or reasoning chains revised. Older failure records are periodically summarized and archived by the LLM. The confidence score $c_\omega \in [0, 1]$ is computed directly from the aggregate statistics.

Reasoning chain $\mathcal{R}_\omega$. The reasoning chain is a structured natural language rationale generated during CSD. It explains why the conditions in $\mathcal{C}_\omega$ are associated with the effectiveness of the strategy in $\mathcal{S}_\omega$. By making this rationale explicit, the reasoning chain supports the auditability, falsifiability, and refinability of each skill. When a skill fails, its reasoning can be examined to identify a potentially invalid assumption or inference step. The chain can also indicate situations where the strategy may no longer be appropriate, allowing flawed reasoning to be revised without relearning the entire skill.

3.2 Contrastive Skill Distillation

CSD builds a skill pool in two sequential stages: Level 1 distills model-selection skills, and Level 2 distills interventions that improve forecasts beyond model selection. Within each chronological training partition, the first 80% of windows form the discovery set $\mathcal{D}_{\text{disc}}$, and the remaining 20% form the confirmation set $\mathcal{D}_{\text{conf}}$. Discovery windows are used to identify performance contrasts and specify candidate conditions and strategies. These are frozen before confirmation outcomes are examined, then tested for paired improvement on eligible confirmation windows. The separate validation partition $\mathcal{D}_{\text{val}}$ is used to assess, refine, and select confirmed skills, but not for the initial significance tests. The resulting skill pool is frozen before test evaluation. The two levels proceed sequentially: Level 1 first establishes a model-selection policy, which is fixed before Level 2 intervention hypotheses are formulated and evaluated. Because the policy may select different forecasters on different windows, Level 2 interventions need not be tied to a single forecaster. At inference, Level 1 selects a forecaster before applicable Level 2 interventions are considered.

Level 1: Model CSD. Given training windows $\{\mathbf{X}_i\}_{i=1}^N$ and a pool of base forecasters $\mathbb{M} = \{M_1, \ldots, M_K\}$ spanning statistical models, neural forecasting models, and time series foundation models, each model forecasts every window, forming a competition matrix:

$$e_{i,j} = \ell\big(M_j(\mathbf{X}_i), \mathbf{Y}_i\big), \tag{2}$$

where ℓ is an evaluation metric such as MSE. This one-time offline procedure requires $O(NK)$ evaluations and is fully parallelizable across models and windows.

For each window $\mathbf{X}_i$, we compute an interpretable feature vector $\mathbf{f}_i$, including STL-based trend and seasonality strength and the ADF test p-value. To identify informative contrasts, we examine the observed loss differences between the globally selected fixed base forecaster and alternative forecasters on each training window. We prioritize cross-family comparisons and windows where an alternative forecaster outperforms the fixed base, as these cases may reveal condition-dependent model strengths. These window-level comparisons are used only to generate candidate skills. For each selected pair, the LLM receives the feature diagnostics, forecast visualizations, and model summaries, and produces a structured contrastive analysis of the observed performance difference.

A single contrastive analysis is merely an anecdote. On $\mathcal{D}_{\text{disc}}$, CSD groups analyses with similar feature profiles and explanatory narratives. Formally, two analyses from discovery windows i and j are candidates for the same group when

$$\text{sim}(\mathbf{f}_i, \mathbf{f}_j) > \tau_f \ \wedge \ \cos\big(\text{embed}(\mathcal{A}_i), \text{embed}(\mathcal{A}_j)\big) > \tau_a, \tag{3}$$

where $\mathcal{A}_i$ is the LLM-generated analysis of window i. For each discovery group $\mathcal{G}_\omega^{\text{disc}}$, the LLM proposes an applicability condition $\mathcal{C}_\omega$ and a model-selection strategy $\mathcal{S}_\omega$. Both are fixed before independent confirmation.

Let $\mathcal{I}_\omega^{\text{conf}}$ denote the indices of confirmation windows satisfying the candidate's frozen applicability condition, assessed without access to their future observations. Let $e_{\text{global},i}$ be the error of a fixed forecaster M_{global}, selected by aggregate performance on $\mathcal{D}_{\text{disc}}$, and let $e_{\omega,i}$ be the error obtained by applying the candidate strategy to the same window. We retain the candidate if

$$\left|\mathcal{I}_\omega^{\text{conf}}\right| \geq s_{\min} \quad \wedge \quad p_{\text{W}}^{+}\left(\{e_{\text{global},i} - e_{\omega,i}\}_{i \in \mathcal{I}_\omega^{\text{conf}}}\right) < \alpha, \qquad \alpha = 0.05, \tag{4}$$

where p_{W}^{+} is the one-sided paired Wilcoxon signed-rank p-value for an improvement in error relative to M_{global}. Thus, discovery windows determine what to test, whereas confirmation windows provide the evidence used to retain the candidate. The LLM synthesizes the skill rationale from the discovery analyses and records the confirmation statistics in $\mathcal{E}_\omega$.

Level 2: Hypothesis-driven strategy distillation. While Level 1 determines *which* forecaster to use, Level 2 investigates *how* to improve forecasts beyond model selection. It focuses on hard windows, where even the best-performing forecaster incurs substantial error. By jointly analyzing these windows, the LLM identifies recurrent residual patterns and summarizes them as shared failure modes. For example, under strong seasonality, systematic underestimation near seasonal peaks and overestimation near troughs may indicate that the best forecaster compresses the seasonal amplitude. The best-performing forecaster on a historical window may be used to diagnose difficult cases, but it does not define the reference forecast for hypothesis verification. Instead, each hypothesis is evaluated against the forecast produced by the fixed Level 1 policy on its eligible windows.

Based on the diagnosed failure mode, the LLM proposes a small set of targeted intervention hypotheses, such as adjusting the lookback window near changepoints, applying variance-stabilizing transformations, routing decomposed components to different forecasters, or correcting systematic forecast biases. Each hypothesis h, including its applicability condition and any fitted parameters, is specified using $\mathcal{D}_{\text{disc}}$ and frozen before confirmation. Let $\mathcal{I}_h^{\text{conf}}$ denote the eligible confirmation-window indices. For each $i \in \mathcal{I}_h^{\text{conf}}$, $e_{\text{L1},i}$ denotes the error of the fixed Level 1 policy, and $e_{h,i}$ denotes the error obtained by applying h within that policy. We retain h if

$$\left|\mathcal{I}_h^{\text{conf}}\right| \geq s_{\min} \quad \wedge \quad p_{\text{W}}^{+}\left(\{e_{\text{L1},i} - e_{h,i}\}_{i \in \mathcal{I}_h^{\text{conf}}}\right) < \alpha, \qquad \alpha = 0.05. \tag{5}$$

An intervention is distilled into a skill only if it passes this confirmation test. Its rationale connects the failure pattern observed on discovery windows, the proposed intervention, and its independently measured improvement.

Iterative refinement. After confirmation, we evaluate the retained skills on the separate validation partition. Let $\mathcal{I}_\omega^{\text{val}}$ be the indices of validation windows satisfying skill ω's current applicability condition, assessed using information available at forecast time. Let $e_i^{(\omega)}$ denote the error of applying the skill under the prescribed forecasting workflow. We define its validation success rate as

$$c_\omega^{\text{val}} = \frac{1}{|\mathcal{I}_\omega^{\text{val}}|} \sum_{i \in \mathcal{I}_\omega^{\text{val}}} \mathbb{I}\left[e_i^{(\omega)} < e_{\omega,i}^{\text{ref}}\right], \qquad e_{\omega,i}^{\text{ref}} = \begin{cases} e_{\text{global},i}, & \omega \text{ is a Level 1 skill,} \\ e_{\text{L1},i}, & \omega \text{ is a Level 2 skill,} \end{cases} \tag{6}$$

where M_{global} is selected on $\mathcal{D}_{\text{disc}}$ and the Level 1 policy is fixed before Level 2 skills are validated. Each skill is thus validated against the same reference used in its confirmation test, so that c_ω^{val} isolates the contribution of the skill itself.

We use c_ω^{val} as a development score for skill selection and refinement, not as an independent test-set estimate of skill success. Skills below a prespecified validation threshold are flagged for revision. The LLM examines their validation failures to identify overly broad conditions or unsupported premises in their rationales. Skills that fail two revision rounds are discarded. We then remove redundant skills and resolve conflicts, repeating this process until the validation stopping criterion is met. The resulting skill pool and all selection thresholds are frozen before test evaluation.

3.3 Skill-Guided Forecasting

At test time, the frozen skill pool is used only for inference. For each forecasting window $\mathbf{X}$, the agent retrieves applicable skills, composes compatible strategies, and checks the resulting forecast for consistency with the diagnosed data characteristics.

Diagnosis and skill retrieval. The agent extracts the feature vector $\mathbf{f_X}$ used during skill acquisition and prompts the LLM to generate a semantic profile of the current window. Retrieval proceeds in two stages. First, coarse filtering matches the statistical predicates of each skill against $\mathbf{f_X}$:

$$\Omega_{\text{cand}} = \left\{\omega \in \Omega : \mathbf{f_X} \text{ satisfies } \mathcal{C}_\omega^{\text{pred}}\right\}. \tag{7}$$

The LLM then reads each candidate's reasoning chain $\mathcal{R}_\omega$ alongside the semantic profile and judges whether the causal premises hold for the current window. It assigns an applicability score:

$$a_\omega = \text{LLM}\big(\mathcal{R}_\omega, \text{profile}(\mathbf{X})\big), \qquad a_\omega \in \{1, \ldots, 5\}, \tag{8}$$

where higher scores indicate stronger alignment between the skill's rationale and the observed window characteristics. We retain skills with $a_\omega \geq 3$. This procedure combines efficient predicate-based filtering with reasoning-based assessment of logical applicability: the LLM judges whether a skill's causal argument is valid for $\mathbf{X}$, rather than merely checking similarity between $\mathbf{f_X}$ and $\mathcal{C}_\omega$.

Skill composition and execution. When multiple skills are retained, the LLM checks whether their strategies address complementary aspects of the forecasting task and do not conflict. Compatible strategies are composed and executed to produce a preliminary forecast. Depending on the selected skills, execution may involve choosing a base forecaster, modifying the lookback window, preprocessing the input, routing series components, or correcting forecast biases.

Reflective plausibility check and correction. The LLM evaluates whether the preliminary forecast is consistent with the diagnosed structural patterns of $\mathbf{X}$. If it identifies a mismatch, it applies one targeted correction, such as revising component weights, adjusting postprocessing, or selecting an alternative compatible strategy. The final output contains the forecast and a natural language explanation of the selected skills and any applied correction.

4 Experiments

4.1 Experimental Settings

Datasets. We conduct extensive experiments on eight real-world datasets to comprehensively compare our method against a broad range of time series forecasting approaches. The datasets include ETTh1, ETTh2, ETTm1, ETTm2 Zhou et al. (2021), ECL, Traffic, Weather Wu et al. (2021), and Solar-Energy Lai et al. (2018), covering diverse application domains such as electricity, transportation, and meteorology. They further vary substantially in the number of variables, temporal resolution, and dataset scale, thereby enabling a comprehensive evaluation across heterogeneous forecasting settings. Detailed descriptions of these datasets are provided in the Appendix A.

Baselines. We select a diverse set of competitive baselines spanning several methodological families, thereby establishing a comprehensive evaluation benchmark. For deep learning-based models, we include PatchTST Nie et al. (2023) and iTransformer Liu et al. (2024), each trained separately on the training set under each experimental configuration. For foundation models, we consider Aurora Wu et al. (2026), Sundial Liu et al. (2025), and Moirai Woo et al. (2024), and evaluate them

using their officially released pretrained checkpoints. For LLM-based approaches, we include OFA Zhou et al. (2023), Time-LLM Jin et al. (2024), and MemCast Tao et al. (2026a), which leverage the reasoning capabilities of large language models to support time series forecasting.

Implementation Details. For each dataset, we evaluate all methods under four forecasting horizons {96, 192, 336, 720}, using mean squared error (MSE) and mean absolute error (MAE) as evaluation metrics. For each dataset, the look-back window is selected from {96, 336, 512} based on validation performance. Our method uses Qwen3.5-27B Qwen Team (2026) as the LLM backbone, with the default sampling configuration of temperature=0.7 and top-p=0.8. For baselines that require training, we follow their official experimental configurations and optimize the models using Adam.

4.2 Main Results

Table 1 reports the averaged forecasting performance across four prediction horizons. Skill4TSF achieves the best MSE on seven of the eight datasets and the best MAE on six datasets, demonstrating strong and consistent performance across diverse domains and temporal granularities. In particular, it yields clear improvements on the ETTm1, ETTm2, ECL, Traffic, and Weather datasets, where it outperforms both specialized deep forecasting models and recent foundation- and LLM-based approaches on both metrics. For example, on ETTm1, Skill4TSF achieves an MSE/MAE of 0.317/0.366, improving over the strongest baseline Time-LLM by 0.012 in MSE and 0.006 in MAE. On ETTm2, it obtains 0.242/0.307, which is 0.009 lower in MSE and 0.006 lower in MAE than Time-LLM. These results suggest that the proposed skill-guided decision process can effectively exploit condition-dependent model strengths rather than relying on a single fixed forecaster. On ETTh1, Skill4TSF attains the best MAE and remains competitive in MSE, while Aurora achieves the lowest MSE. Similarly, although Moirai obtains a marginally lower MAE on ETTh2 and Sundial achieves the best MAE on Solar, Skill4TSF still delivers the lowest MSE on both datasets. Overall, the results validate that combining contrastively distilled model-selection and intervention skills enables robust forecasting improvements across heterogeneous time series characteristics.

Table 1: Overall forecasting performance on different datasets. Lower values indicate better performance. Results are averaged from all prediction lengths {96, 192, 336, 720}. For each dataset, the best result is highlighted in red. Full results are listed in Appendix C.

Models	Skill4TSF		MemCast		Time-LLM		OFA		Aurora		Sundial		Moirai		PatchTST		iTransformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.317	0.366	0.359	0.392	0.329	0.372	0.352	0.383	0.346	0.383	0.336	0.377	0.390	0.389	0.387	0.400	0.407	0.410
ETTm2	0.242	0.307	0.265	0.322	0.251	0.313	0.266	0.326	0.256	0.324	0.258	0.320	0.276	0.320	0.281	0.326	0.288	0.332
ETTh1	0.396	0.414	0.439	0.441	0.408	0.423	0.427	0.426	0.386	0.415	0.411	0.434	0.510	0.469	0.469	0.454	0.454	0.447
ETTh2	0.326	0.385	0.364	0.398	0.334	0.383	0.354	0.394	0.337	0.384	0.333	0.387	0.354	0.377	0.387	0.407	0.383	0.407
ECL	0.154	0.247	0.165	0.261	0.158	0.252	0.167	0.263	0.178	0.275	0.169	0.265	0.188	0.273	0.205	0.290	0.178	0.270
Traffic	0.381	0.259	0.426	0.293	0.388	0.264	0.414	0.294	0.524	0.352	-	-	-	-	0.481	0.304	0.428	0.282
Weather	0.214	0.250	0.242	0.266	0.225	0.257	0.237	0.270	0.230	0.267	0.234	0.270	0.260	0.275	0.259	0.281	0.258	0.278
Solar	0.199	0.278	0.210	0.285	0.206	0.286	0.215	0.296	0.203	0.289	0.221	0.252	0.714	0.704	0.270	0.307	0.233	0.262

4.3 Ablation Studies

4.3.1 Impact of Each Module

In this subsection, we evaluate the contributions of the framework's key components. Skill distillation consists of two stages: Level 1 (Model CSD) and Level 2 (Hypothesis-Driven Strategy Distillation). To assess Level 1, we disable Model CSD and always select the forecaster with the best overall training-set performance. To assess Level 2, we disable hypothesis-driven strategy distillation, retaining model selection but constraining the framework to a single fixed workflow. To evaluate iterative refinement, we prevent the initially distilled, significance-filtered skills from being revised in response to validation-set failure cases. Finally, to assess reflection, we remove the final reflection stage, thereby ending forecasting without reflection-driven refinement. We conduct ablation studies on four representative datasets; results are shown in Figure 2. Level 1 and Level 2 have the largest effects on performance: disabling either increases the MSE on ETTm1 by 0.009 and 0.011, respectively, demonstrating the contribution of the two-stage distillation framework. Iterative refinement also provides substantial gains, with its removal increasing the MSE by 0.006 on both ETTm1 and Weather. Reflection yields consistent, albeit smaller, improvements across all datasets,

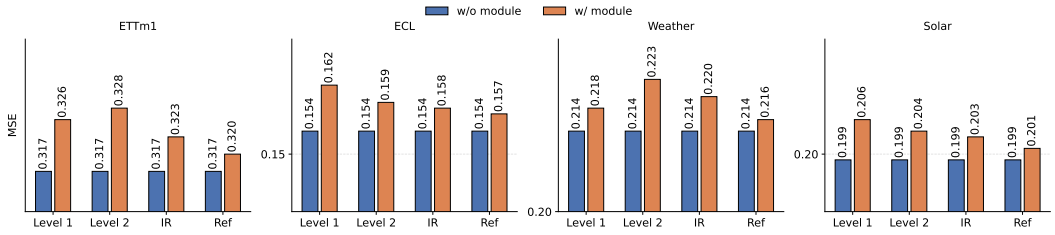


Figure 2: Ablation study of the key components in our framework. Level 1 denotes Model CSD, Level 2 denotes Hypothesis-Driven Strategy Distillation, IR denotes Iterative Refinement, and Ref denotes the Reflection Mechanism. Results are reported in terms of MSE, where lower values indicate better forecasting performance.

with its removal increasing the MSE by 0.002–0.003. Thus, although reflection is not the dominant contributor, it further improves forecasting performance.

4.3.2 Impact of LLM Backbones

Table 2: Effect of LLM backbone scale on the forecasting performance of our framework. Results are reported in terms of MSE, where lower values indicate better performance. For each dataset, the best result is highlighted in red.

Datasets	ETTm1	ETTm2	ETTh1	ETTh2	ECL	Traffic	Weather	Solar
Qwen3.5-27B	0.317	0.242	0.396	0.326	0.154	0.381	0.214	0.199
Qwen3.5-9B	0.322	0.250	0.402	0.331	0.158	0.391	0.220	0.203
Qwen3.5-4B	0.325	0.253	0.425	0.339	0.160	0.396	0.224	0.204
Qwen3.5-2B	0.327	0.253	0.424	0.343	0.166	0.397	0.223	0.208

In this subsection, we investigate the effect of LLM backbone scale on our framework. To control for model-family differences, we exclusively use models from the Qwen3.5 family, with parameter scales of 27B, 9B, 4B, and 2B. The results are reported in Table 2. As the model size decreases, its reasoning ability and general knowledge capacity tend to deteriorate, leading to a corresponding decline in the performance of our framework. For example, on ETTh1, replacing the Qwen3.5-27B backbone with Qwen3.5-2B increases the MSE by 0.028. Similarly, on Traffic, replacing Qwen3.5-27B with Qwen3.5-9B increases the MSE from 0.381 to 0.391. These results indicate that the capability of the LLM backbone has a substantial impact on our method. Stronger backbone models enable more effective decision-making and reasoning throughout the forecasting process. More importantly, the observed scaling trend suggests that our framework can effectively exploit the reasoning capabilities of LLMs for time series forecasting.

4.3.3 Online Updating

Because our framework is skill-based, its learning process does not require updating the parameters of the underlying LLM. It can therefore be incrementally updated with newly acquired experience at substantially lower computational cost than reinforcement learning-based approaches. In this subsection, we evaluate our method in an online learning setting to demonstrate its capacity for continual adaptation. Specifically, on four representative datasets, we update the agent's skills using newly observed data as additional training evidence after every S time steps. We vary the update interval S to investigate how update frequency affects the final forecasting performance. The results are presented in Figure 3. The horizontal axis denotes the skill-update frequency, with the update interval S selected from $\{100, 200, 400, 800\}$. Overall, the best performance is typically achieved with relatively small update intervals. More frequent updates allow the agent to respond more promptly to evolving environments and distribution shifts, thereby enabling the forecasting process to adapt more quickly. However, updating too frequently is not necessarily optimal. Skill refinement requires aggregating multiple forecasting trajectories to distill generalizable knowledge

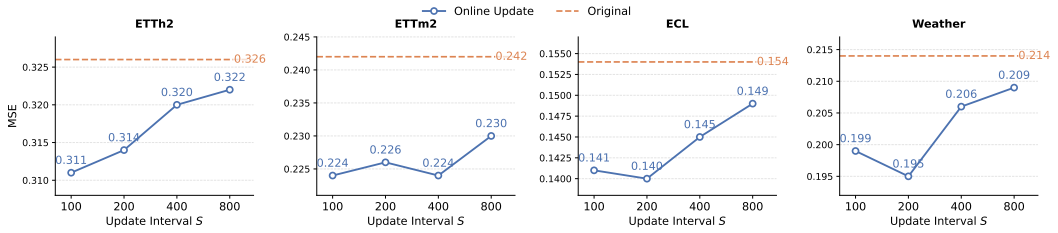


Figure 3: Effect of the skill-update interval S on the online forecasting performance of our framework across four representative datasets. Skills are incrementally refined using newly observed forecasting trajectories every S time steps. Lower MSE indicates better performance.

and robust decision-making strategies. When S is excessively small, the agent may not accumulate sufficient trajectory evidence between updates, making it difficult to derive effective skills.

4.3.4 Transfer Across Models

Table 3: Cross-model transferability of distilled skills across Qwen3.5 backbones. Source-only, target-only, and source+target denote the use of skills distilled by the source model, the target model, and both models, respectively. Results are reported in terms of MSE (lower is better).

Source Model	Target Model	ETTm1			ECL		
		Source only	Target only	Source+Target	Source only	Target only	Source+Target
Qwen3.5-27B	Qwen3.5-9B	0.325	0.322	0.320	0.160	0.158	0.155
	Qwen3.5-4B	0.324	0.325	0.321	0.163	0.160	0.159
	Qwen3.5-2B	0.328	0.327	0.324	0.165	0.166	0.161
Qwen3.5-9B	Qwen3.5-27B	0.324	0.317	0.315	0.161	0.154	0.153
Qwen3.5-4B		0.327		0.315	0.159		0.152
Qwen3.5-2B		0.327		0.319	0.168		0.158

As external textual artifacts, skills can be readily transferred across agents and LLM backbones. In this subsection, we investigate the cross-model transferability of the skills learned by our framework. Specifically, source-only denotes that the target model uses only the skills learned by the source model. Target-only indicates that the target model uses only its own learned skills. Source+target denotes that the target model jointly uses skills learned by both the source and target models. The results are presented in Table 3. We observe that transferring skills learned by a stronger model to a weaker target model does not necessarily outperform the weaker model's own skills when used in isolation. However, combining the transferred skills with the target model's native skills often yields a substantial improvement. Conversely, skills learned by a weaker model generally cannot enable a stronger target model to reach the performance achieved by its own skills. Nevertheless, integrating these transferred skills with the target model's native skill set can occasionally provide benefits.

5 Conclusion

We present Skill4TSF, a forecasting agent that adapts its workflow through explicit, reusable skills rather than updates to the underlying language model. Its central mechanism, Contrastive Skill Distillation, combines conditional model selection with hypothesis-driven strategy refinement, grounding both in matched forecasting comparisons across multiple windows. By recording applicability conditions, comparative evidence, and explanatory rationales, Skill4TSF produces inspectable skills that support forecasting decisions and targeted revision. Experiments across eight benchmarks demonstrate strong forecasting performance, while ablations highlight the complementary contributions of model-selection and intervention skills. Further studies examine how these textual artifacts can facilitate online adaptation and transfer across language-model backbones. Future work should develop more efficient skill acquisition and retrieval, strengthen safeguards for strategy composition, and systematically evaluate performance under distribution shifts.

AI USE STATEMENT

We used generative AI tools to improve the grammar and readability of the manuscript. In addition, LLMs are components of the proposed method. These methodological uses are described in Section 3. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics [1]. It studies the application of agent-based methods to time series forecasting using publicly available benchmark datasets and publicly released or publicly accessible forecasting models. The study does not involve human participants or the collection of private information.

REPRODUCIBILITY STATEMENT

The datasets and models used in this work are publicly available and are cited in the manuscript. The proposed method is described in Section 3 and illustrated in Figure 1. Experimental settings and implementation details are provided in Section 4.1 and Appendix B. We will release the code upon acceptance of the paper.

REFERENCES

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

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

Pierre-Daniel Arsenault, Shengrui Wang, and Jean-Marc Patenaude. A survey of explainable artificial intelligence (xai) in financial time series forecasting. *ACM Computing Surveys*, 57(10):1–37, 2025.

Vassilis Assimakopoulos and Konstantinos Nikolopoulos. The theta model: a decomposition approach to forecasting. *International journal of forecasting*, 16(4):521–530, 2000.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE Transactions on Intelligent Transportation Systems*, 10 (2):246–254, 2009. doi: 10.1109/TITS.2009.2021448.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

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

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026.

[1] https://iclr.cc/public/CodeOfEthics

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 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. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=LO7ciRpjI5.

Ziyu Ma, Shidong Yang, Yuxiang Ji, Xucong Wang, Yong Wang, Yiming Hu, Tongwen Huang, and Xiangxiang Chu. Skillclaw: Let skills evolve collectively with agentic evolver. *arXiv preprint arXiv:2604.08377*, 2026.

Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Francisco Javier Nogales, Javier Contreras, Antonio J Conejo, and Rosario Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17(2): 342–348, 2002.

Chetanya Puri, Gerben Kooijman, Bart Vanrumste, and Stijn Luca. Forecasting time series in healthcare with gaussian processes and dynamic time warping based subset selection. *IEEE Journal of Biomedical and Health Informatics*, 26:6126–6137, 10 2022. doi: 10.1109/jbhi.2022.3214343.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9): 2363–2377, May 2024. ISSN 2150-8097. doi: 10.14778/3665844.3665863. URL https://doi.org/10.14778/3665844.3665863.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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*, volume 2025, pp. 34635–34667, 2025.

Robert H Shumway and David S Stoffer. Arima models. In *Time series analysis and its applications: with R examples*, pp. 85–175. Springer, 2024.

Xiaoyu Tao, Mingyue Cheng, Ze Guo, Shuo Yu, Yaguo Liu, Qi Liu, and Shijin Wang. Memcast: Memory-driven time series forecasting with experience-conditioned reasoning. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=IoY0A6opfl.

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

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Yd8eHMY1wz.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Xingjian Wu, Jianxin Jin, Wanghui Qiu, Peng Chen, Yang Shu, Bin Yang, and Chenjuan Guo. Aurora: Towards universal generative multimodal time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VVJ6Ck9JBl.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with \$10k\$ parameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bWcnvZ3qMb.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026.

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

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, Xue Liu, Xiaoxiao Li, and Philip S. Yu. Coevoskills: Self-evolving agent skills via co-evolutionary verification, 2026. URL https://arxiv.org/abs/2604.01687.

Haokun Zhao, Xiang Zhang, Jiaqi Wei, Yiwei Xu, Yuting He, Siqi Sun, and Chenyu You. Timeseriesscientist: A general-purpose ai agent for time series analysis. *arXiv preprint arXiv:2510.01538*, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Yingli Zhou, Wang Shu, Yaodong Su, Wenchuan Du, Yixiang Fang, and Xuemin Lin. A comprehensive survey on agent skills: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2605.07358*, 2026.

A Dataset Description

We evaluate our method on eight public time-series forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, ECL, Traffic, Weather, and Solar-Energy.

(1) **ETT** Zhou et al. (2021) comprises multivariate measurements from electricity transformers, including load and oil temperature. The `ETTh1` and `ETTh2` subsets are sampled hourly, whereas `ETTm1` and `ETTm2` are sampled every 15 minutes.

(2) **ECL** Wu et al. (2021) contains hourly electricity consumption records for 321 clients from 2012 to 2014.

(3) **Traffic** Wu et al. (2021) contains hourly road-occupancy measurements collected by 862 sensors on freeways in the San Francisco Bay Area from 2015 to 2016.

(4) **Weather** Wu et al. (2021) comprises 21 meteorological variables recorded every 10 minutes in 2020 at a weather station operated by the Max Planck Institute for Biogeochemistry.

(5) **Solar-Energy** Lai et al. (2018) contains solar-power production records from 137 photovoltaic plants, sampled every 10 minutes in 2006.

We follow the data preprocessing procedures and train/validation/test split protocols adopted in prior work Wu et al. (2023); Liu et al. (2024). All datasets are split chronologically to prevent temporal information leakage. Further details and dataset statistics are provided in Table 4.

Table 4: Dataset descriptions and summary statistics. Channels is the number of variables per time step, Timesteps is the total number of observations, Frequency is the sampling interval, and Domain is the application area. The datasets span real-world forecasting tasks in energy, transportation, and meteorology.

Dataset	Channels	Timesteps	Frequency	Domain
ETTh1	7	14400	Hourly	Electricity
ETTh2	7	14400	Hourly	Electricity
ETTm1	7	57600	15 minutes	Electricity
ETTm2	7	57600	15 minutes	Electricity
ECL	321	26304	Hourly	Electricity
Traffic	862	17544	Hourly	Transportation
Weather	21	52696	10 minutes	Meteorology
Solar	137	52560	10 minutes	Energy

B Implementation Details

For skill acquisition, we construct forecasting windows with a stride of 8 for ETTh1, ETTh2, ECL, and Traffic, and a stride of 16 for the other datasets. A unit stride would produce highly overlapping windows and many near-duplicate samples. Using larger strides reduces the number of windows to process, and thus the associated computational cost, while remaining small relative to the look-back window length, preserving dense temporal coverage with limited loss of temporal information. All diagnostic features are computed solely from the observed look-back window. The windows in each training partition are ordered chronologically: the first 80% are used to discover candidate skills, while the remaining 20% are reserved for their independent confirmation. We retain candidates supported by at least 30 confirmation windows and assess their improvement over a reference forecaster selected without access to the current window's future observations. The validation partition is used to select applicability thresholds and refine skill conditions. The resulting skill pool and all selection thresholds are frozen before test evaluation. Ground-truth future values are available for residual analysis during offline skill acquisition, but are never provided to the agent when forecasting a test window.

Each skill is stored as a structured record containing its feature predicates, semantic condition, required forecaster, strategy parameters, evidence summary, and rationale. At inference time, skills satisfying the feature predicates are assigned an applicability score from 1 to 5 by the LLM. We

execute at most three mutually compatible skills with scores of at least 3. When a required forecaster is unavailable or an intervention cannot be applied, we fall back to the selected base forecast. We constrain LLM responses to a predefined JSON schema and allow up to two retries for invalid outputs. The final reflection step permits at most one correction and has access only to the input window, diagnostic features, selected skills, and preliminary forecast.

We use a fixed pool of eleven forecasters spanning three model families: statistical methods (Prophet Taylor & Letham (2018), ARIMA Shumway & Stoffer (2024), and Theta Assimakopoulos & Nikolopoulos (2000)), deep-learning models (PatchTST Nie et al. (2023), iTransformer Liu et al. (2024), DLinear Zeng et al. (2023), and TimesNet Wu et al. (2023)), and pretrained time-series foundation models (TimesFM Das et al. (2024), MOIRAI Woo et al. (2024), Chronos Ansari et al. (2024), and Time-MoE Shi et al. (2025)). Level 1 model-selection skills choose among these forecasters, while Level 2 intervention skills operate on the selected forecast. We deliberately use this diverse, fixed pool rather than continually incorporating newer models, such as Aurora. This design allows us to examine whether improvements arise from the agent's data-dependent model selection and forecasting interventions, rather than relying solely on the strength of a newly advanced model.

We evaluate forecasting performance using two widely adopted metrics: mean squared error (MSE) and mean absolute error (MAE). Given a ground-truth sequence $\boldsymbol{y} = (y_1, \ldots, y_n)$ and its corresponding predictions $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_n)$, the metrics are defined as follows:

$$\mathrm{MSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{9}$$

$$\mathrm{MAE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{10}$$

Table 5: Full forecasting results on eight datasets at prediction lengths of 96, 192, 336, and 720. Lower MSE and MAE indicate better performance. "Avg" denotes the unweighted mean across the four prediction lengths for each dataset. Red indicates the lowest value for each dataset, prediction length, and metric. Ties are highlighted jointly. A dash indicates that a result is unavailable.

Models		Skill4TSF		MemCast		Time-LLM		OFA		Aurora		Sundial		Moirai		PatchTST		iTransformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.269	0.328	0.306	0.361	0.272	0.334	0.292	0.346	0.294	0.351	0.280	0.334	0.353	0.363	0.329	0.367	0.334	0.368
	192	0.301	0.355	0.341	0.382	0.310	0.358	0.332	0.372	0.331	0.374	0.321	0.366	0.376	0.380	0.367	0.385	0.377	0.391
	336	0.330	0.380	0.371	0.400	0.352	0.384	0.366	0.394	0.359	0.391	0.350	0.389	0.399	0.395	0.399	0.410	0.426	0.420
	720	0.368	0.401	0.418	0.425	0.383	0.411	0.417	0.421	0.398	0.414	0.394	0.418	0.432	0.417	0.454	0.439	0.491	0.459
	Avg	0.317	0.366	0.359	0.392	0.329	0.372	0.352	0.383	0.346	0.383	0.336	0.377	0.390	0.389	0.387	0.400	0.407	0.410
ETTm2	96	0.153	0.246	0.169	0.263	0.161	0.253	0.173	0.262	0.179	0.270	0.170	0.256	0.189	0.260	0.175	0.259	0.180	0.264
	192	0.211	0.288	0.229	0.305	0.219	0.293	0.229	0.301	0.232	0.307	0.229	0.300	0.247	0.300	0.241	0.302	0.250	0.309
	336	0.270	0.325	0.298	0.344	0.271	0.329	0.286	0.341	0.275	0.337	0.281	0.337	0.295	0.334	0.305	0.343	0.311	0.348
	720	0.334	0.369	0.364	0.376	0.352	0.379	0.378	0.401	0.338	0.380	0.351	0.387	0.372	0.386	0.402	0.400	0.412	0.407
	Avg	0.242	0.307	0.265	0.322	0.251	0.313	0.266	0.326	0.256	0.324	0.258	0.320	0.276	0.320	0.281	0.326	0.288	0.332
ETTh1	96	0.335	0.375	0.375	0.397	0.362	0.392	0.376	0.397	0.340	0.381	0.348	0.385	0.380	0.398	0.414	0.419	0.386	0.405
	192	0.379	0.403	0.422	0.426	0.398	0.418	0.416	0.418	0.377	0.405	0.393	0.418	0.440	0.434	0.460	0.445	0.441	0.436
	336	0.419	0.429	0.465	0.457	0.430	0.427	0.442	0.433	0.399	0.422	0.422	0.440	0.514	0.474	0.501	0.466	0.487	0.458
	720	0.451	0.449	0.494	0.484	0.442	0.457	0.477	0.456	0.428	0.450	0.481	0.493	0.705	0.568	0.500	0.488	0.503	0.491
	Avg	0.396	0.414	0.439	0.441	0.408	0.423	0.427	0.426	0.386	0.415	0.411	0.434	0.510	0.469	0.469	0.454	0.454	0.447
ETTh2	96	0.251	0.319	0.280	0.342	0.268	0.328	0.285	0.342	0.259	0.325	0.271	0.333	0.287	0.325	0.302	0.348	0.297	0.349
	192	0.314	0.369	0.353	0.388	0.329	0.375	0.354	0.389	0.324	0.370	0.327	0.376	0.347	0.367	0.388	0.400	0.380	0.400
	336	0.353	0.409	0.393	0.421	0.368	0.409	0.373	0.407	0.360	0.401	0.354	0.402	0.377	0.393	0.426	0.433	0.428	0.432
	720	0.386	0.443	0.430	0.441	0.372	0.420	0.406	0.441	0.403	0.441	0.381	0.435	0.404	0.421	0.431	0.446	0.427	0.445
	Avg	0.326	0.385	0.364	0.398	0.334	0.383	0.354	0.394	0.337	0.384	0.333	0.387	0.354	0.377	0.387	0.407	0.383	0.407
ECL	96	0.125	0.216	0.136	0.233	0.131	0.224	0.139	0.238	0.134	0.234	0.132	0.229	0.152	0.242	0.181	0.270	0.148	0.240
	192	0.143	0.235	0.156	0.251	0.152	0.241	0.153	0.251	0.161	0.258	0.152	0.250	0.171	0.259	0.188	0.274	0.162	0.253
	336	0.158	0.251	0.170	0.270	0.160	0.248	0.169	0.266	0.193	0.287	0.173	0.271	0.192	0.278	0.204	0.293	0.178	0.269
	720	0.190	0.286	0.198	0.290	0.192	0.298	0.206	0.297	0.224	0.320	0.218	0.311	0.236	0.313	0.246	0.324	0.225	0.317
	Avg	0.154	0.247	0.165	0.261	0.158	0.252	0.167	0.263	0.178	0.275	0.169	0.265	0.188	0.273	0.205	0.290	0.178	0.270
Traffic	96	0.350	0.238	0.391	0.270	0.362	0.248	0.388	0.282	0.435	0.314	-	-	-	-	0.462	0.295	0.395	0.268
	192	0.365	0.247	0.411	0.281	0.374	0.247	0.407	0.290	0.465	0.328	-	-	-	-	0.466	0.296	0.417	0.276
	336	0.384	0.263	0.434	0.300	0.385	0.271	0.412	0.294	0.525	0.355	-	-	-	-	0.482	0.304	0.433	0.283
	720	0.425	0.288	0.468	0.321	0.430	0.288	0.450	0.312	0.670	0.411	-	-	-	-	0.514	0.322	0.467	0.302
	Avg	0.381	0.259	0.426	0.293	0.388	0.264	0.414	0.294	0.524	0.352	-	-	-	-	0.481	0.304	0.428	0.282
Weather	96	0.140	0.194	0.159	0.211	0.147	0.201	0.162	0.212	0.160	0.207	0.157	0.205	0.177	0.208	0.177	0.218	0.174	0.214
	192	0.184	0.232	0.204	0.250	0.189	0.234	0.204	0.248	0.202	0.247	0.205	0.251	0.219	0.249	0.225	0.259	0.221	0.254
	336	0.236	0.268	0.263	0.283	0.262	0.279	0.254	0.286	0.252	0.288	0.253	0.289	0.277	0.292	0.278	0.297	0.278	0.296
	720	0.296	0.306	0.342	0.320	0.304	0.316	0.326	0.337	0.307	0.327	0.320	0.336	0.365	0.350	0.354	0.348	0.358	0.347
	Avg	0.214	0.250	0.242	0.266	0.225	0.257	0.237	0.270	0.230	0.267	0.234	0.270	0.260	0.275	0.259	0.281	0.258	0.278
Solar	96	0.178	0.255	0.188	0.262	0.181	0.267	0.193	0.274	0.185	0.272	0.204	0.230	0.682	0.688	0.234	0.286	0.203	0.237
	192	0.192	0.270	0.202	0.277	0.198	0.278	0.208	0.288	0.198	0.282	0.221	0.248	0.694	0.695	0.267	0.310	0.233	0.261
	336	0.207	0.286	0.217	0.293	0.215	0.293	0.221	0.302	0.211	0.294	0.225	0.260	0.719	0.706	0.290	0.315	0.248	0.273
	720	0.219	0.301	0.233	0.308	0.230	0.306	0.238	0.320	0.218	0.307	0.233	0.272	0.759	0.725	0.289	0.317	0.249	0.275
	Avg	0.199	0.278	0.210	0.285	0.206	0.286	0.215	0.296	0.203	0.289	0.221	0.252	0.714	0.704	0.270	0.307	0.233	0.262

C Full Experimental Results

Due to space constraints, the main text reports results averaged across the forecasting horizons. We provide the complete MSE and MAE results for every method and forecasting horizon in Table 5. It reports the MSE and MAE of all methods on each dataset at prediction lengths of 96, 192, 336, and 720. The "Avg" row for each dataset is the arithmetic mean of the four horizon-specific results and corresponds to the aggregated results discussed in the main text.

Skill4TSF achieves the lowest MSE in 27 of the 32 dataset-horizon settings and the lowest MAE in 23 settings, counting ties. It obtains the lowest MSE at every evaluated horizon on ETTm1, ETTm2, ECL, Traffic, and Weather. The horizon-specific results also reveal exceptions that are less apparent from dataset-level averages. On ETTh1, Aurora achieves a lower MSE at horizons 192, 336, and 720, whereas Skill4TSF achieves the lowest MAE at three of the four horizons. On ETTh2, Skill4TSF has the lowest MSE at horizons 96, 192, and 336, but does not achieve the lowest MAE on average. On Solar, Sundial attains the lowest MAE at every horizon, while Aurora slightly outperforms Skill4TSF in MSE at horizon 720. These results indicate that Skill4TSF's aggregate improvements extend to most individual forecasting settings, while its relative performance remains dependent on the dataset, horizon, and evaluation metric.

D Additional Analyses

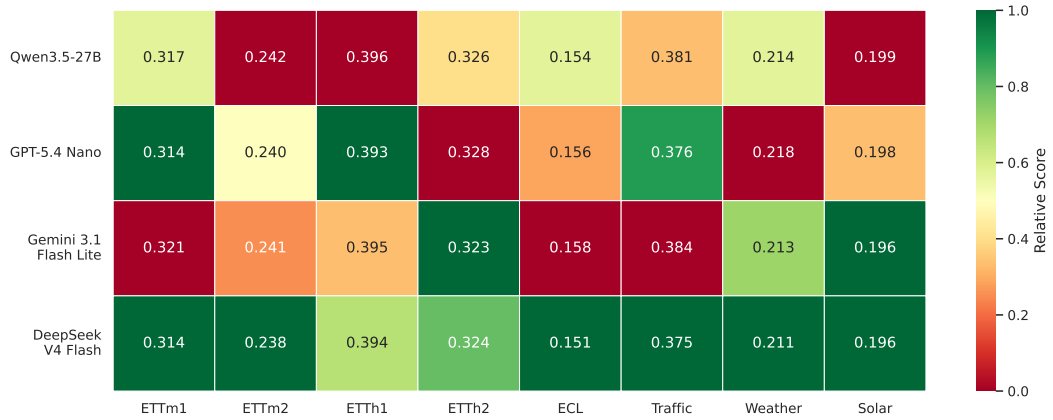


Figure 4: MSE across LLM backbones and datasets. Each cell reports the MSE achieved by the corresponding backbone on the corresponding dataset. Cell colors indicate relative performance within each dataset, ranging from red (highest MSE) to green (lowest MSE).

D.1 Impact of different LLM backbones

In this subsection, we extend our evaluation to more LLM backbones from different model families to assess the applicability of Skill4TSF across diverse model capabilities. Specifically, we include Gemini 3.1 Flash Lite [2], DeepSeek V4 Flash Xu et al. (2026), and GPT-5.4 Nano [3], which differ in their design priorities and capabilities. The results are presented in Figure 4. DeepSeek V4 Flash achieves the best or near-best performance in most cases, suggesting that its capabilities are well suited to the reasoning and decision-making tasks in our framework. Gemini 3.1 Flash Lite and GPT-5.4 Nano also achieve leading results on Solar and ETTh1, respectively. These findings suggest that Skill4TSF can leverage the distinct strengths of different LLM backbones, rather than depending on a single model family.

[2] https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/3-1-flash-lite

[3] https://openai.com/index/introducing-gpt-5-4-mini-and-nano

D.2 Illustrative Skill Records

To illustrate the structure of the distilled skills, we present two skill records from the ETTm1 forecasting task to illustrate the outputs of the two distillation levels. The first specifies a condition-dependent model preference, whereas the second specifies an intervention applied after Level 1 model selection. Both records follow the representation $\omega = (\mathcal{C}_\omega, \mathcal{S}_\omega, \mathcal{E}_\omega, \mathcal{R}_\omega)$ defined in Section 3.1.

Level 1: Model-selection skill for pronounced daily seasonality

Condition $\mathcal{C}_{\omega_1}$. *Symbolic predicates:* ETTm1 windows with STL seasonality strength greater than 0.65 and trend strength below 0.40, where the seasonal period is 96 observations (one day at 15-minute resolution). *Semantic condition:* The recent history exhibits a relatively stable daily pattern without a dominant long-term trend or an evident changepoint.

Strategy $\mathcal{S}_{\omega_1}$. Select PatchTST instead of the globally selected fixed forecaster M_{global} for windows satisfying the condition. If PatchTST is unavailable, retain the forecast from M_{global}.

Evidence $\mathcal{E}_{\omega_1}$. On an independent confirmation subset, the condition matches 48 windows. PatchTST achieves lower MSE than M_{global} on 36 of them, with a mean paired MSE reduction of 0.019. The one-sided paired Wilcoxon signed-rank test yields $p = 0.004$. On held-out validation windows, the skill improves upon the baseline in 31 of 45 applicable cases, giving $c_{\omega_1} = 31/45 \approx 0.689$.

Rationale $\mathcal{R}_{\omega_1}$. When a daily seasonal pattern dominates the observed window and the trend is comparatively weak, representing repeated temporal motifs as patches may be advantageous relative to the fixed forecaster's treatment of the same input. The confirmation comparison supports this *conditional preference*, not a claim that PatchTST is preferable for ETTm1 in general. The skill should not be applied when the daily pattern becomes unstable or a pronounced trend or changepoint invalidates its premises.

Level 2: Intervention skill for seasonal-amplitude compression

Condition $\mathcal{C}_{\omega_2}$. *Symbolic predicates:* ETTm1 windows with STL seasonality strength greater than 0.75 and a detectable daily component with period 96. *Semantic condition:* Seasonal peaks and troughs in the look-back window are persistent enough to support extrapolation of their phase and approximate shape. This intervention is not tied to a single base forecaster: its reference forecast is produced by the fixed Level 1 model-selection policy.

Strategy $\mathcal{S}_{\omega_2}$. Estimate a daily seasonal component from the observed look-back window and extrapolate it to the prediction horizon. Apply a pre-estimated amplitude correction to the Level 1 forecast:

$$\widehat{\mathbf{Y}}^{\,\text{corr}} = \widehat{\mathbf{Y}}^{\,\text{L1}} + \gamma\, \widehat{\mathbf{s}}^{\,\text{future}}, \qquad \gamma = 0.12,$$

where $\widehat{\mathbf{Y}}^{\,\text{L1}}$ is the forecast from the fixed Level 1 policy and $\widehat{\mathbf{s}}^{\,\text{future}}$ is obtained solely from the input window. The coefficient γ is fitted on historical skill-discovery windows and fixed before independent confirmation. It is never estimated from the future values of the window being forecast.

Evidence $\mathcal{E}_{\omega_2}$. Residual analysis on historical hard windows suggested a recurring underestimation of seasonal peaks and overestimation of troughs. On 42 independent confirmation windows satisfying $\mathcal{C}_{\omega_2}$, the corrected forecast improves upon the fixed Level 1 policy on 30 windows, with a mean paired MSE reduction of 0.012 and a one-sided paired Wilcoxon signed-rank p-value of 0.009. On held-out validation windows, the skill outperforms the baseline in 30 of 44 applicable cases, yielding $c_{\omega_2} = 30/44 \approx 0.682$.

Rationale $\mathcal{R}_{\omega_2}$. The historical residual pattern is consistent with forecasts whose daily variation has insufficient amplitude under strong, stable seasonality. Adding a calibrated fraction of the look-back-derived seasonal component targets this failure mode without changing the forecaster selected by Level 1. The intervention should be withheld when the seasonal pattern is weak or unstable, because a phase or amplitude error in the extrapolated component could worsen the forecast. At inference time, applicability is assessed from the input window and the skill record. Historical residuals motivate the skill but are not available for the current forecast.