REATS: LLM REASONING-BASED ENSEMBLE LEARNING FOR ADAPTIVE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in time series forecasting (TSF) have produced models with complementary design strengths, but no single model handles the diverse patterns of all samples. Existing ensembles rely on fixed rules or numerical-only black-box models, failing to exploit LLM reasoning or explain weighting decisions. We propose REATS, an intelligent LLM ensemble router that jointly processes textual temporal-pattern descriptions and numerical features to generate interpretable, sample-adaptive weights through chain-of-thought reasoning. To enable effective LLM-based ensembling, we study its key design choices and propose: (i) a structured input pipeline that converts raw series into hybrid textual–numerical representations with token cost fixed across input lengths, supports LLM reasoning and API-free rule-based chain-of-thought construction, and incorporates retrieved similar-sample priors; (ii) diverse multi-row weight supervision to enrich training signals, together with a token-efficient percentage-table format that reduces numerical complexity and LLM hallucinations; and (iii) a two-stage fine-tuning framework combining SFT for structured reasoning and GRPO with a reciprocal reward mapping that maps continuous, unbounded regression rewards to bounded signals, amplifies near-oracle sensitivity, and addresses uniform sensitivity and outlier-dominated advantage compression in mixed-quality rollout groups. Experiments on eight benchmarks show that REATS outperforms competitive ensemble baselines for both foundation-model and small-model candidate groups, while providing natural-language explanations and strong transfer learning and out-of-domain generalization to unseen candidate models. The code is available at the link https://anonymous.4open.science/r/REATS-8688.

1 INTRODUCTION

Time series forecasting (TSF) supports applications such as energy management (Uremović et al., 2022), financial analysis (Zhang et al., 2025a), and transportation planning (Wachs, 1987). Recent models span Transformers (Zhang et al., 2025a), linear architectures (Liu et al., 2022; Zhang et al., 2025b), and convolutional networks (Zhu et al., 2023), whose distinct designs target different temporal characteristics. Yet each applies one modeling strategy to all samples, although the dominant patterns and best-performing architecture can vary across samples and scenarios. Ensemble learning can address this limitation by combining complementary strengths, with the central challenge being how to assign appropriate model weights for each sample.

Traditional ensembles use fixed weights or static validation-error strategies (Chen & Luc, 2022; Gruber et al., 2015), which cannot adapt to complex, sample-specific temporal patterns. Neural routers enable dynamic weighting (Fu et al., 2022), but rely only on numerical inputs and cannot exploit textual knowledge about temporal patterns or model design. They also require retraining when candidate models change, restricting the expressiveness and flexibility of ensemble decisions. Table 1 compares these paradigms with our LLM-based approach.

The reasoning and language capabilities of LLMs offer a new way to overcome these limitations: (1) **Multimodal information fusion** combines textual temporal-pattern descriptions (e.g., trend, seasonality, and volatility) with numerical data for comprehensive reasoning; (2) **Flexible scalability** adjusts the candidate pool through prompt-level model descriptions without retraining; and (3)

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.

Table 1: Design-property comparison of ensemble-routing paradigms for TSF. Check marks indicate whether the method explicitly supports the property by design.

Method	Multimodal input	Sample-adaptive	Unseen model generalization	Human-readable reasoning
Static ensemble	✗	✗	✗	✗
NN-based ensemble	✗	✓	✗	✗
LLM ensemble (REATS)	✓	✓	✓	✓

Inherent interpretability explains how temporal characteristics relate to model suitability and weighting decisions, facilitating human inspection.

Applying LLMs to this task poses three challenges: (1) raw-series token costs scale linearly with length, requiring a fixed-budget representation that preserves temporal semantics; (2) scalable supervision must avoid prohibitively expensive API-generated CoT over massive training sets, while a single optimal weight vector provides insufficient diversity and weak feedback; and (3) typical GRPO (Shao et al., 2024) uses binary or bounded rewards unsuitable for continuous MSE optimization. The naive $r = -\delta$ has uniform sensitivity that poorly distinguishes near-optimal candidates, while its unbounded range lets one poor generation dominate group variance. This compression persists after advantage normalization because the inflated σ_r collapses reward spacing among the remaining good candidates, motivating a nonlinear bounded mapping that reshapes within-group reward geometry.

Based on the above insights, we propose REATS: LLM Reasoning-based Ensemble Learning for Adaptive Time Series Forecasting. Our contributions are as follows:

- We propose an ensemble learning framework, REATS, that leverages the reasoning capabilities of LLMs for TSF, and study the key techniques that make this paradigm effective: how to represent time series for an LLM router, how to construct reasoning and weight supervision at scale, and how to adapt GRPO to a continuous forecasting objective. The framework includes a systematic sample construction pipeline (e.g., retrieval-augmented prior knowledge, diverse weight supervision, and token-efficient weight table format) and a two-stage fine-tuning process, achieving strong performance in both accuracy and interpretability.
- We design a hybrid textual–numerical input representation with fixed token cost that activates semantic reasoning and enables rule-based CoT construction without API dependency. The rule-based CoT significantly enhances both in-domain and out-of-domain ensemble performance, demonstrating that expensive API-generated CoT is not always necessary for effective LLM fine-tuning. Besides, the designed multi-row weight supervision enriches fine-tuning signals during SFT and GRPO stages.
- We design a reciprocal reward mapping function that adapts GRPO to continuous regression tasks by addressing two limitations of the naive $r = -\delta$: uniform sensitivity that struggles to discriminate near-optimal candidates, and unbounded range that allows outlier-dominated advantage compression in mixed-quality rollout groups. The resulting bounded, nonlinear mapping enables effective late-stage policy refinement. This reward-level solution outperforms several recent GRPO algorithmic variants that target reward sparsity through optimization modifications.
- Experiments on eight benchmarks show that REATS consistently surpasses the competitive baselines while demonstrating robust OOD generalization and enhanced transfer learning.

2 Related Work

We focus on ensemble learning for time series forecasting, the literature most relevant to REATS. Appendix A.2 reviews individual forecasters. **Recent NLP and LLM ensembles** rank or fuse multiple model-generated responses to improve generation quality (Lv et al., 2024; Yun et al., 2025; Jiang et al., 2023). However, these methods operate on textual candidates and therefore do not directly address TSF ensembles, whose candidate models produce continuous numerical trajectories and require sample-specific numerical weights. **Numerical forecasting ensembles** largely follow two paradigms. Traditional methods use fixed, uniform, or performance-based weights (Bertsimas & Boussioux, 2023; Chen & Luc, 2022; Gruber et al., 2015). They are simple and efficient but cannot adapt allocations to the temporal pattern of each input. Neural and reinforcement-learning

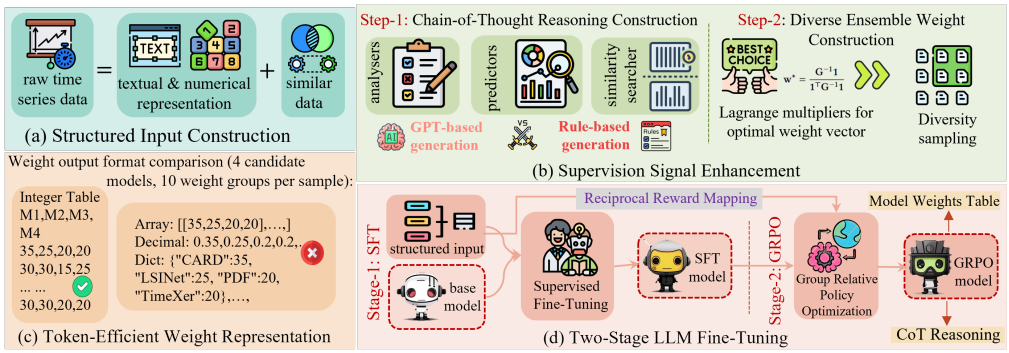


Figure 1: Overall framework of REATS.

routers (Fu et al., 2022) instead map numerical temporal features to dynamic weights learned through forecasting-oriented reward signals, gaining sample-level adaptability but remaining tied to numerical representations and a fixed candidate pool. Despite their ability to combine semantic knowledge with explicit reasoning, LLMs remain underexplored as adaptive routers for numerical TS forecasts. REATS studies the representations, supervision, and optimization needed to realize this capability.

3 LLM Reasoning-based Ensemble Learning for Adaptive Time Series Forecasting

3.1 Problem Definition

Given a univariate history $\mathbf{X} = [x_1, \ldots, x_T]$ and horizon H, each of N candidate models $\mathcal{M} = \{M_1, \ldots, M_N\}$ predicts $\hat{\mathbf{Y}}_i = M_i(\mathbf{X}) = [\hat{y}^i_{T+1}, \ldots, \hat{y}^i_{T+H}]$. Their ensemble is:

$$\hat{\mathbf{Y}} = \sum_{i=1}^{N} w_i \cdot \hat{\mathbf{Y}}_i, \quad \text{s.t.} \sum_{i=1}^{N} w_i = 1,\ w_i \geq 0 \tag{1}$$

where $\mathbf{w} = [w_1, \ldots, w_N]$. We fine-tune an LLM to infer these weights through explicit CoT from temporal-pattern features, retrieved similar series, and candidate-model descriptions, minimizing forecasting error.

3.2 Overview

Figure 1 illustrates the components of REATS: **(1) Structured Input Construction** transforms raw series into hybrid textual–numerical descriptions augmented with retrieved similar-sample priors; **(2) Supervision Signal Enhancement** constructs chain-of-thought reasoning and diverse weight supervision; **(3) Token-Efficient Weight Representation** formats weights as integer percentage tables; and **(4) Two-Stage Fine-Tuning** performs SFT for structured reasoning, followed by GRPO with a reciprocal reward mapping that adapts continuous, unbounded regression rewards for direct forecasting optimization. Component (1)–(3) are detailed in Section 3.3 and the (4) in Section 3.4.

3.3 Structured LLM finetuning data construction

Figure 2 illustrates the fine-tuning data construction pipeline, covering input construction (feature extraction and retrieval augmentation) and output construction (CoT reasoning, diverse weights, and weight table format).

Hybrid textual–numerical feature extraction. Directly tokenizing raw values as text causes token counts to grow linearly with series length (e.g., 1656→7480 from length 96 to 512) and increases hallucination risk, while encoding series through an MLP projector weakens interpretability and OOD generalization (Table 10). Instead, we extract eight groups $\Phi(\mathbf{X}) = \{\phi_{\text{sta}}, \phi_{\text{noise}}, \phi_{\text{trend}}, \phi_{\text{sea}}, \phi_{\text{ac}}, \phi_{\text{stat}}, \phi_{\text{out}}, \phi_{\text{dist}}\}$ covering stationarity, noise, trend, seasonality, autocorrelation, statistics, IQR outliers, and

Algorithm 1: Rule-Based Chain-of-Thought Generation

Input: Temporal features $\Phi(\mathbf{X})$, candidate models $\mathcal{M}$ (termed *tools* in the prompt) with model category labels, oracle weights $\mathbf{w}^*$, RAG reference $\Phi(\mathbf{X}_{\text{rag}})$, $\mathbf{w}_{\text{rag}}$
Output: CoT text

$\bar{\mathbf{w}} \leftarrow$ average of K' oracle weight rows;
// Step 1: Key patterns
$s_1 \leftarrow$ template phrases for each parsed attribute of $\Phi(\mathbf{X})$;
// Step 2: Oracle-guided tool matching
foreach *tool* t_i *sorted by* $\bar{w}_i$ *descending* **do**
    Match tool category against temporal attributes via rules;
    Select fit-level template guided by $\bar{w}_i$: high (>0.35) / mid (>0.15) / low;
end
$s_2 \leftarrow$ join tool-match sentences;
// Step 3: RAG reference comparison
$s_3 \leftarrow$ template noting agreement/divergence between $\bar{\mathbf{w}}$ and $\mathbf{w}_{\text{rag}}$;
// Step 4: Allocation conclusion
$s_4 \leftarrow$ conclusion template naming dominant/moderate/minor tools;
return $s_1 \oplus s_2 \oplus s_3 \oplus s_4$;

distribution (Appendix Table 13). Each group has a header (e.g., "[trend]") and key-value pairs (e.g., "Trend slope: -0.00454" and "Trend strength: Strong"), preserving values and semantic interpretations. Values are rounded to three significant figures. This representation has fixed token cost across input lengths, activates LLM reasoning, and provides explicit anchors for Rule-CoT construction.

Retrieval-augmented prior knowledge construction. We construct a knowledge pool $\mathcal{P} = \{(\mathbf{X}_j, \mathbf{w}_j)\}_{j=1}^{|\mathcal{P}|}$ from training set samples only, where $\mathbf{w}_j \in \mathbb{R}^N$ stores the optimal ensemble weights. Given a query series $\mathbf{X}$, we retrieve the top-K most similar samples:

$$\mathcal{R}(\mathbf{X}) = \underset{j \in \mathcal{I}(\mathbf{X})}{\arg\min\text{-K}} \, \|\mathbf{X} - \mathbf{X}_j\|_2, \quad \mathcal{I}(\mathbf{X}) = \{j : \mathbf{X}_j \neq \mathbf{X}\} \text{ during training.} \tag{2}$$

The retrieved samples provide $\bar{\mathbf{w}} = \frac{1}{K} \sum_{j \in \mathcal{R}} \mathbf{w}_j$ and are averaged at the raw-series level into one prototype, then converted to the query's hybrid representation. This captures neighborhood patterns without K-fold token expansion and grounds LLM reasoning in empirical model performance, improving reasoning and ensemble accuracy.

Chain-of-Thought reasoning construction. CoT supervision follows four steps: (1) identify dominant temporal features; (2) analyze candidate suitability; (3) interpret retrieved weights; and (4) justify the allocation. A `<think>...</think>` block precedes the weights in a "reason-then-decide" output. During data preparation, our API-free rules use training-set oracle weights to reverse-engineer explanations by matching temporal features to candidate-category strengths and instantiating template variants, avoiding costly GPT calls while achieving comparable quality (Table 6(a)). At inference, the LLM generates CoT and weights without oracle information.

Algorithm 1 summarizes the procedure. Appendix Algorithm 2 and Appendix A.9 provide detailed rules and a complete example.

Diverse ensemble weight construction. We first derive the optimal (oracle) weight vector $\mathbf{w}^*$ by minimizing ensemble MSE $= \mathbf{w}^\top \mathbf{G} \mathbf{w}$ (where $G_{ij} = \frac{1}{T} \sum_t e_{i,t} e_{j,t}$) subject to $\sum_i w_i = 1$ and $w_i \geq 0$. Without the non-negativity constraint, the closed-form solution is:

$$\mathbf{w}^* = \frac{\mathbf{G}^{-1}\mathbf{1}}{\mathbf{1}^\top \mathbf{G}^{-1}\mathbf{1}} \tag{3}$$

The full derivation is shown in Appendix A.7. In practice, we solve the full simplex-constrained quadratic program $\mathbf{w}^* = \arg\min_{\mathbf{w} \geq 0, \, \mathbf{1}^\top \mathbf{w} = 1} \mathbf{w}^\top (\mathbf{G} + \epsilon \mathbf{I}) \mathbf{w}$ to obtain the exact non-negative oracle weights. To avoid overfitting to a single target, we generate nine diverse weight vectors via Dirichlet sampling and multi-objective greedy selection. Together with the oracle vector in the first row, they form K'=10 rows (Appendix A.8). These auxiliary targets (1) make SFT CoT capture broader patterns across valid allocations and (2) give GRPO additional rewards that mitigate sparsity and encourage weight-space exploration.

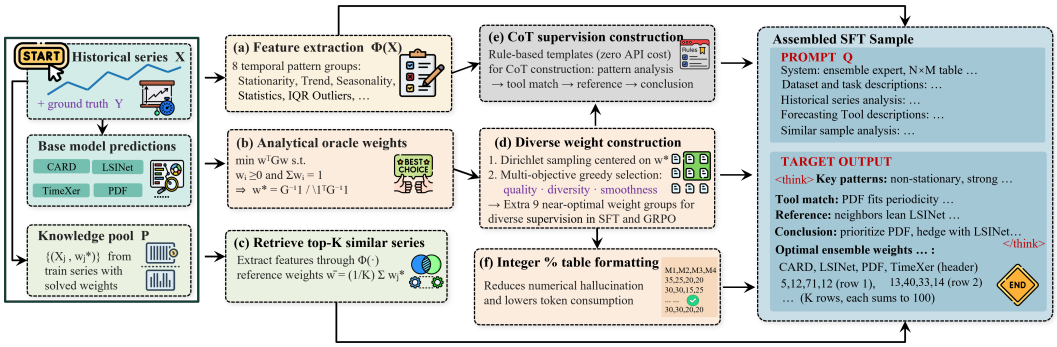


Figure 2: Structured LLM fine-tuning data construction pipeline.

Token-efficient weight representation. We represent weights as integer percentages formatted as a compact comma-separated table, e.g., `CARD,SEMixer,TimeXer,LSINet` (header) `/ 35,25,20,20` (row 1) `/ 30,30,15,25` (row 2). This reduces numerical hallucination and lowers token consumption (empirically validated in Table 3(b) and (c)).

3.4 Two-Stage Fine-Tuning with SFT and GRPO

We train a lightweight LLM (1.7B parameters) in two stages: supervised fine-tuning (SFT) followed by Group Relative Policy Optimization (GRPO).

Stage 1: Supervised Fine-Tuning. Using next-token prediction, each prompt Q targets a `<think>...</think>` CoT block followed by a K'-row table. SFT teaches the "reason-then-decide" format, pattern-aware reasoning, and coarse oracle-like allocations, but its token loss does not directly optimize forecasting MSE.

Stage 2: Group Relative Policy Optimization. GRPO (Shao et al., 2024) samples G outputs per prompt, each parsed into K' weight rows, and uses three reward terms:

- *Relative MSE Reward.* For ensemble MSE $\mathcal{L}(\mathbf{w})$, $\delta_i = \mathcal{L}(\mathbf{w}_i) - \mathcal{L}(\mathbf{w}_i^*)$ is the gap between predicted row i and its oracle.
- *Oracle Proximity Reward.* Weight-space MSE between predicted and oracle vectors supplies auxiliary gradients when forecasting-MSE signals are sparse.
- *Format Penalty.* A negative penalty applies only to negative weights or sums $\neq 100\%$.

Unlike binary or bounded rewards in standard GRPO tasks, our raw signal is a continuous MSE gap $\delta \geq 0$. The naive $r = -\delta$ has two distinct limitations: its uniform sensitivity does not prioritize fine distinctions near the oracle, and its unbounded range lets one outlier inflate group standard deviation σ_r and collapse normalized advantages $\hat{A}_i = (r_i - \mu_r)/\sigma_r$ among near-optimal candidates. DAPO (Yu et al., 2026), DrGRPO (Liu et al.), GSPO (Zheng et al., 2025), and SAPO (Gao et al., 2025) modify optimization but not the underlying reward, leaving these reward issues unresolved (Section 4.3).

The reciprocal $r = 1/(1+k\delta)$ bounds rewards in $[0, 1]$ while distinguishing "poor" from "worse" early and "good" from "better" near convergence, avoiding far-range collapse and near-oracle saturation. Its nonlinear bounded shape compresses outlier gaps and amplifies near-oracle differences, a change preserved after advantage normalization unlike linear rescaling (Section 4.3 and Appendices A.6.1–A.6.4).

The overall reward is:

$$r = \lambda_1 \cdot r_{\text{mse}} + \lambda_2 \cdot r_{\text{agg}} + \lambda_3 \cdot r_{\text{oracle}} + p_{\text{format}} \tag{4}$$

where $r_{\text{mse}} = \frac{1}{K'} \sum_{i=1}^{K'} \frac{1}{1+k\cdot\delta_i}$, $r_{\text{oracle}} = \frac{1}{K'} \sum_{i=1}^{K'} \frac{1}{1+k\cdot\text{MSE}(\mathbf{w}_i, \mathbf{w}_i^*)}$, and $r_{\text{agg}} = \frac{1}{1+k\cdot\text{MSE}(\sum_j \bar{w}_j \hat{\mathbf{y}}_j, \mathbf{y})}$ with $\bar{\mathbf{w}} = \frac{1}{K'} \sum_i \mathbf{w}_i$.

Thus, SFT initializes structured reasoning, format compliance, and coarse weights, while GRPO explores beyond imitation targets to refine weights directly for forecasting accuracy.

4 EXPERIMENTS AND RESULTS

4.1 EXPERIMENTAL SETTINGS

Datasets and metrics, we evaluate univariate forecasting on ETTh1/2, ETTm1/2, Exchange, Weather, Electricity, and Traffic using MSE and MAE, with input/prediction lengths of 96 and a 7:1:2 train/validation/test split (Appendix A.1.1). **Candidate model pool for ensemble,** our candidate pool contains 10 small specialized forecasting models and 8 time-series foundation models, which also serve as individual baselines (Appendix A.1.2). **Ensemble baselines,** we compare REATS with heuristic averaging/random weighting, train/validation-based InvMSE and OptW, neural RLMC, five zero-shot LLM ensembles, and four recent GRPO variants (Appendix A.1.3). **Implementation details,** REATS uses Qwen3-1.7B, K=3 retrieved neighbors, K'=10 weight rows, four candidates by default, and reciprocal-reward parameters k=20 and $(\lambda_1, \lambda_2, \lambda_3)$=$(0.8, 0, 0.2)$. All oracle weights are derived exclusively from training-set ground truth (Appendix A.1.4).

4.2 COMPARISON OF ENSEMBLE FORECASTING PERFORMANCE

Foundation model candidates. Table 2(a) shows that no foundation model dominates all datasets. Zero-shot Codex (0.1594) matches OptW_{val} (0.1597), whereas REATS-GRPO reaches **0.1384**, improving by 13.3% and winning all datasets. Its gain over SFT (0.1455→0.1384) confirms that MSE rewards refine weights beyond imitation.

Small model candidates. Table 2(b) is harder: extreme model variance (e.g., TimeXer at 1.6664) makes averaging (0.3173) worse than the best individual model, and zero-shot Codex (0.1709) trails OptW_{tr} (0.1352), indicating that LLMs without fine-tuning struggle with highly heterogeneous candidates. REATS-GRPO achieves **0.1080**, a 20.1% reduction, showing that fine-tuned reasoning identifies complementary strengths among unreliable candidates.

Statistical significance. The per-dataset wins in both candidate groups are statistically significant under the sign test (p<0.05).

Generalization to unseen candidate models. Trained on small candidates and evaluated on unseen foundation models, REATS reconstructs RAG pools and descriptions from training predictions without test labels. It achieves **0.1442** (Table 3), beating OptW_{val} (0.1564) by 7.8% and DeepSeek-V3.2 (0.1626) by 11.3%. Its leading mixed- and within-group OOD results (Tables 12 and 16) indicate transferable temporal reasoning rather than model-identity memorization.

Generalization to different numbers of candidate models. Across 2, 4, 6, and 8 foundation candidates, REATS generally has the lowest average MSE, with gains tending to grow with the candidate set, as shown in Figure 3.

4.3 DISCUSSION OF GRPO REWARD MAPPING

Comparison of bounded mapping functions. We compare reciprocal $r = 1/(1 + k\delta)$, exponential $r = \exp(-k\delta)$, linear $r = \max(0, 1 - k\delta)$, and unbounded $r = -\delta$. GRPO requires signal across diverse within-group δ values to avoid gradient vanishing. The reciprocal's polynomial decay preserves sensitivity over a wide range while strongly discriminating near the oracle, unlike the uniform sensitivity of $r = -\delta$ (Figure 5(a)). It also counters outlier-driven advantage compression through (1) *boundedness*, which limits an outlier's effect on σ_r, and (2) *nonlinearity*, which preserves improved reward separation after normalization, unlike affine rescaling.

The reciprocal mapping gives the lowest MSE on both datasets, while $r = -\delta$ is comparable to or worse than bounded alternatives (Figure 5(b)(c)). Appendices A.6.1–A.6.4 detail k selection, sensitivity, theory, and outlier compression.

Comparison with GRPO variants. DAPO, DrGRPO, GSPO, and SAPO address reward sparsity through optimization-level changes such as normalization, sample filtering, and soft advantage computation, but do not reshape the underlying continuous, unbounded signal. Standard GRPO with reciprocal mapping outperforms these variants using shared $r = -\delta$ (Figure 5(d)), showing the complementary value of improving reward quality directly (Appendix Table 8).

Table 2: Ensemble forecasting results (MSE ↓). Dataset abbreviations: Exch=Exchange, H1/H2=ETTh1/ETTh2, M1/M2=ETTm1/ETTm2, Wea=Weather, Elec=Electricity, Traf=Traffic. The MAE results are shown in appendix Table 14.

(a) Foundation Model Candidates

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
MOIRAI	.1675	.2161	.2413	.1273	.3164	.0269	.5908	.1489	.2294
MOMENT	.3037	.1966	.2765	.1124	.1707	.5683	.7635	.9884	.4225
TimeMoE	.2359	.1768	.1996	.0997	.1416	.0752	.3902	.0831	.1753
TimesFM	.1845	.1712	.2212	.1112	.2862	.0039	.4770	.0864	.1927
Traditional Ensemble									
Ens_{avg}	.1932	.1624	.2077	.0944	.1771	.0766	.4593	.1508	.1902
Ens_{rand}	.1990	.1680	.2132	.0979	.1875	.0939	.4786	.1857	.2030
$\text{InvMSE}_{\text{tr}}$	.1831	.1621	.2061	.0940	.1605	.0560	.4383	.0831	.1729
$\text{InvMSE}_{\text{val}}$	.1915	.1621	.2062	.0940	.1602	.0082	.4379	.0825	.1678
NN_{RLMC}	.1864	.1617	.2047	.0915	.1615	.0032	.3900	.0821	.1602
OptW_{tr}	.1703	.1613	.1988	.0929	.1429	.0433	.3932	.0821	.1606
OptW_{val}	.1967	.1618	.2013	.0928	.1429	.0069	.3930	.0823	.1597
LLM-based Ensemble									
GPT-5.2	.1617	.1491	.1936	.0874	.1579	.0237	.4095	.0942	.1596
Codex	.1605	.1471	.1924	.0860	.1490	.0424	.4064	.0914	.1594
GPT-5.5	.1650	.1529	.1973	.0890	.1644	.0745	.4191	.1036	.1707
DS-V3.2	.1535	.1448	.1919	.0852	.1506	.0392	.4086	.1084	.1603
Grok-4	.1681	.1521	.1986	.0897	.1626	.0836	.4337	.1143	.1753
REATS (Ours)									
REATS-SFT	.1466	.1403	.1864	.0831	.1381	.0050	.3839	.0806	.1455
REATS-GRPO	**.1368**	**.1336**	**.1763**	**.0806**	**.1320**	**.0030**	**.3688**	**.0760**	**.1384**

(b) Small Model Candidates

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
CARD	.1690	.1647	.1944	.0979	.1147	.4372	.4442	.0667	.2111
LSINet	.5319	.1558	.1839	.0815	.0839	1.2352	.2607	2.1674	.5875
PDF	.6344	.1515	.1825	.0748	.0729	.8325	**.1472**	2.1738	.5337
TimeXer	1.4382	1.9124	1.9351	1.8879	1.9464	.4429	1.6963	2.0720	1.6664
Traditional Ensemble									
Ens_{avg}	.3146	.2420	.2730	.1753	.1892	.1926	.3168	.8353	.3173
Ens_{rand}	.3918	.3127	.3439	.2476	.2636	.3026	.3813	.9927	.4045
$\text{InvMSE}_{\text{tr}}$	.1879	.1517	.1810	.0781	.0783	.2081	.1757	.0750	.1420
$\text{InvMSE}_{\text{val}}$	.3064	.1591	.2058	.0866	.0783	.1932	.1869	.0733	.1612
RLMC	.3875	.1572	.3029	.0970	.0838	.1666	.1549	**.0666**	.1771
OptW_{tr}	**.1561**	.1549	.1795	.0753	.0740	.2072	.1486	.0859	.1352
OptW_{val}	.2629	.1599	.1911	.0928	.0733	.1928	.1540	.0858	.1516
LLM-based Ensemble									
GPT-5.2	.2497	.1857	.2112	.1103	.1140	.1130	.2127	.2889	.1857
Codex	.2397	.1724	.2014	.0994	.1059	.1240	.2016	.2231	.1709
GPT-5.5	.2574	.1904	.2155	.1150	.1182	.1378	.2255	.3821	.2052
DS-V3.2	.2435	.1708	.2008	.1050	.1077	.1168	.1943	.2439	.1729
Grok-4	.2617	.1862	.2191	.1164	.1253	.1809	.2232	.4550	.2210
REATS (Ours)									
REATS-SFT	.1888	.1492	.1769	.0785	.0792	.0424	.1605	.0925	.1210
REATS-GRPO	.1587	**.1409**	**.1700**	**.0725**	**.0720**	**.0206**	.1476	.0819	**.1080**

Table 3: Left: Trained on small model candidates from Table 2(b), evaluated on 4 unseen foundation models (mapping in Appendix Table 11). Right: Ablation study on small model candidates. Default REATS uses CoT + RAG + integer percentage table. ID = in-distribution, OOD = out-of-distribution. MAE results are in Appendix Table 15. Other OOD results with similar conclusions are in Table 16.

(a) OOD to Foundation Models (MSE ↓)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.3037	.1966	.2765	.1124	.1707	.5683	.7635	.9884	.4225
Model 2	.2359	.1742	.2022	.0987	.1229	.4397	.3740	.0893	.2171
Model 3	.1845	.1712	.2212	.1112	.1416	.0752	.5569	.1800	.2052
Model 4	.1675	.2161	.2413	.1273	.3164	**.0039**	.5908	.0831	.2183
Traditional Ensemble									
Ens_{avg}	.1930	.1618	.2090	.0946	.1391	.1670	.4765	.1725	.2017
Ens_{rand}	.1994	.1674	.2141	.0981	.1489	.1837	.4957	.2056	.2141
$\text{InvMSE}_{\text{tr}}$	.1831	.1615	.2076	.0943	.1290	.1083	.4485	.0906	.1779
$\text{InvMSE}_{\text{val}}$	.1915	.1615	.2079	.0943	.1291	.0072	.4481	.0896	.1662
RLMC	.1697	.1619	.2106	.0902	.1228	.2903	.3740	**.0795**	.1874
OptW_{tr}	.1703	.1608	.2013	.0930	.1182	.0761	.3797	.0839	.1604
OptW_{val}	.1967	.1619	.2067	.0927	.1184	.0109	.3797	.0842	.1564
LLM-based Ensemble									
GPT-5.2	.1656	.1486	.1965	.0881	.1152	.0771	.4158	.1122	.1649
Codex	.1598	.1453	.1952	.0862	.1126	.0973	.4157	.1013	.1642
GPT-5.5	.1710	.1546	.2035	.0912	.1260	.1157	.4356	.1259	.1779
DeepSeek-V3.2	.1587	.1449	.1940	.0860	.1080	.1047	.4032	.1011	.1626
Grok-4	.1734	.1531	.2021	.0897	.1258	.1641	.4438	.1353	.1859
REATS (Ours)									
REATS-GRPO	**.1388**	**.1350**	**.1821**	**.0854**	**.1043**	.0062	**.3613**	.1409	**.1442**

(b) Component & Format Ablation (MSE ↓)

Cat.	Variant	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
CoT	REATS-SFT (ID)	**.1888**	**.1492**	**.1769**	**.0785**	**.0792**	**.0424**	.1605	**.0925**	**.1210**
	w/o CoT (ID)	.2415	.1587	.1886	.0833	.0839	.0447	**.1582**	.0929	.1315
	REATS-SFT (OOD)	**.1881**	**.2125**	**.3408**	**.1051**	**.1705**	.0653	.2001	**.0924**	**.1719**
	w/o CoT (OOD)	.2412	.2639	.3689	.2070	.7563	**.0292**	**.1809**	.0926	.2675
RAG	REATS-SFT (ID)	**.1888**	**.1492**	**.1769**	**.0785**	.0792	**.0424**	**.1605**	**.0925**	**.1210**
	w/o RAG (ID)	.2388	.1532	.1815	.0790	**.0792**	.1464	.1629	.0935	.1418
	REATS-SFT (OOD)	**.1881**	**.2125**	**.3408**	**.1051**	**.1705**	**.0653**	**.2001**	**.0924**	**.1719**
	w/o RAG (OOD)	.3499	.2378	.7690	.3823	.3524	.1413	.4445	.0990	.3470
Weight Fmt	REATS-SFT (ID)	**.1888**	.1492	**.1769**	.0785	.0792	.0424	.1605	**.0925**	**.1210**
	Decimal (ID)	.2182	.1511	.1776	**.0769**	**.0779**	.0429	.1597	.0931	.1247
	Dict (ID)	.2060	**.1487**	.1775	.0781	.0786	.0567	.1613	.0933	.1250
	Array (ID)	.2236	.1500	.1788	.0779	.0785	**.0309**	**.1575**	.0926	.1237
	REATS-SFT (OOD)	**.1881**	.2125	.3408	.1051	.1705	.0653	**.2001**	**.0924**	.1719
	Decimal (OOD)	.1950	.2164	**.3043**	**.1034**	.1682	.0720	.2069	.0979	**.1705**
	Dict (OOD)	.1905	.2223	.3293	.1165	**.1481**	.0737	.2105	.0947	.1732
	Array (OOD)	.2797	**.1966**	.3595	.2233	.2627	**.0511**	.2061	.0925	.2089

(c) Token Efficiency & Inference Speed (s/sample)

Format	Example Row	Tokens	vs Table	Infer	vs Table
Table	`40,30,20,10`	**103**	—	**1.10**	—
Array	`[40, 30, 20, 10],...,`	132	+28%	1.21	+10%
Decimal	`0.40,0.30,0.20,0.10,...,`	182	+77%	1.42	+29%
Dict	`{"CARD":40, "LSINet":30,...,`	292	+183%	2.05	+86%

4.4 Ablation Study

We evaluate REATS design choices on small candidate models.

Chain-of-Thought (CoT). By analyzing patterns and candidate suitability before weighting, CoT improves ID MSE from 0.1315 to 0.1210 (8.7%). Its OOD effect is larger: removing it raises MSE from 0.1719 to 0.2675 (55.6%), because unseen candidates require reasoning over series features, retrieved references, and candidate designs rather than memorized input-weight mappings.

Retrieval-Augmented Generation (RAG). Removing similar-case reference anchors degrades both ID and especially OOD performance, confirming that RAG grounds reasoning in unfamiliar scenarios.

Weight output format. As shown in Table 3(b), among integer tables, decimals, dictionaries, and arrays, the table gives the best ID MSE (0.1210 vs. 0.1247 for decimals), near-best OOD MSE

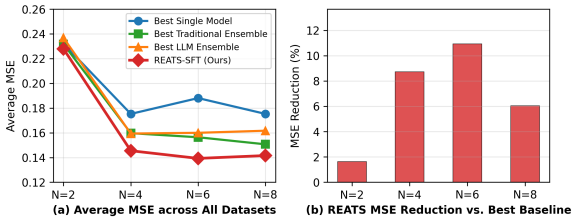


Figure 3: Scalability to different numbers of candidate models. N=4 results correspond to Table 2(a). Full results for other N values are in Appendix Table 18 (MSE) and Table 19 (MAE).



Figure 4: Transfer learning comparison (zero-shot model group, MSE ↓). Each source dataset (X-axis) is used for training, and the average MSE across the other 7 target datasets (Y-axis) is reported. Baselines are the best method from each category.

Table 4: Effectiveness of the diverse weight table, where K' denotes the number of weight rows.

	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
K'=1	.1753	.1410	.1785	.0740	.0739	.0745	.1491	.0855	.1190
K'=10	**.1587**	**.1409**	**.1700**	**.0725**	**.0720**	**.0206**	**.1476**	**.0819**	**.1080**

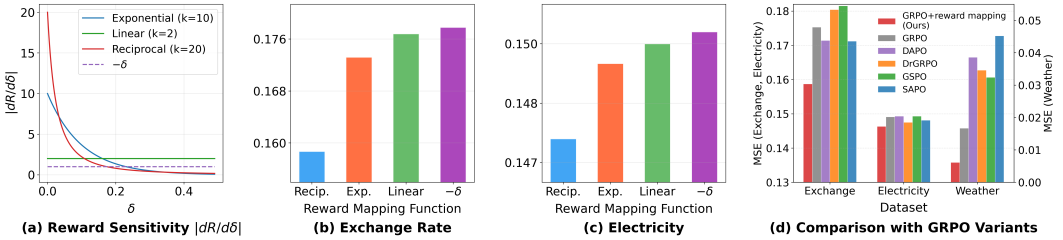


Figure 5: Reward mapping analysis. (a) Raw reward sensitivity $|dr/d\delta|$ of four mappings. (b)(c) Empirical MSE on Exchange and Electricity under different reward mappings. (d) Comparison with recent GRPO variants (all using naive $r = -\delta$). Full results in Appendix Table 8.

(0.1719 vs. 0.1705), the fewest tokens (103 vs. 292 for dictionaries), and fastest inference (1.10 vs. 2.05 s/sample), showing the table format's advantages in accuracy, efficiency, and parsing simplicity.

Number of supervision weight rows K'. Reducing K' from 10 to the oracle row alone raises average MSE from 0.1080 to 0.1190 (10.2%) on all datasets (Table 4). Evaluating each rollout on K' allocations reduces reward sparsity and separates group rewards. These rows span a near-optimal weight-space neighborhood, so reward reflects a simplex region and its gradient provides richer directional guidance than one-row comparison.

Input representation. Holding all other settings fixed, hybrid textual–numerical and MLP inputs tie for best ID accuracy and both outperform raw-series text, but the former generalizes best OOD by eliciting semantic rather than position-dependent numerical reasoning, as shown in the Appendix Table 10. It also keeps 1085 tokens across lengths, versus raw text's 1656→7480 (Appendix Table 9), and uniquely supports Rule-CoT construction (Section 3.3).

4.5 Discussion on CoT

Rule CoT VS LLM CoT. Both Rule-CoT and GPT-CoT are constructed using the same temporal features, candidate-model information, RAG references, and explicitly provided oracle weights. They differ only in how the reasoning text is generated. Rule-CoT beats GPT-CoT on 6/8 small-model datasets (0.1210 vs. 0.1260) and nearly ties it for TSFMs (0.1455 vs. 0.1456, as shown in Table 6(a)).Rule-CoT's structural consistency facilitates SFT, while GPT-CoT may hallucinate or vary across similar inputs. Its comparable or better performance at zero API cost shows advantages.

Impact of CoT on GRPO. CoT improves GRPO on all datasets, reducing average MSE from 0.1151 to 0.1080 (6.2% improvement shown in Table 6(b)). Alternative reasoning paths emphasize different patterns and diversify weights, enriching exploration and reward signals. Without CoT, direct weight generation lacks diversity, resulting in sparse reward variance.

Table 5: RAG top-k analysis (MSE ↓).

	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
k=25	.2132	.1508	.1805	**.0780**	**.0791**	.0535	.1612	.0935	.1262
k=3	**.1888**	**.1492**	**.1769**	.0785	.0792	**.0424**	**.1605**	**.0925**	**.1210**

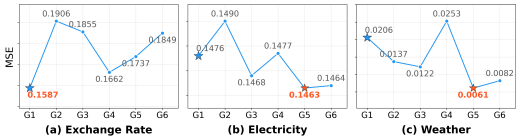


Figure 6: GRPO reward coefficient sensitivity across datasets.

Table 6: Further analysis on CoT (MSE ↓). The settings are same in Table 2.

(a) Rule-CoT vs GPT-CoT on SFT.

Group	Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Small	GPT-CoT	.2152	.1591	.1822	.0793	.0815	**.0294**	**.1589**	.1025	.1260
	Rule-CoT	**.1888**	**.1492**	**.1769**	**.0785**	**.0792**	.0424	.1605	**.0925**	**.1210**
TSFM	GPT-CoT	.1486	.1406	.1866	**.0828**	**.1359**	**.0045**	.3851	**.0805**	.1456
	Rule-CoT	**.1466**	**.1403**	**.1864**	.0831	.1381	.0050	**.3839**	.0806	**.1455**

(b) Effectiveness of CoT on GRPO

Group	Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Small	w/o CoT	.1936	.1424	.1704	.0725	.0735	.0359	.1491	.0835	.1151
	w/ CoT	**.1587**	**.1409**	**.1700**	**.0725**	**.0720**	**.0206**	**.1476**	**.0819**	**.1080**

Interpretability and GRPO reasoning improvement. REATS explains weights through pattern analysis → tool matching → RAG reference → allocation, allowing users to inspect why particular tools receive higher weights. Beyond improving accuracy, GRPO also improves reasoning quality: for an autocorrelation-dominated Weather series, it replaces SFT's periodicity-based PDF preference with the oracle-aligned attention model CARD (Appendix A.10).

4.6 Transfer Learning Comparison

Trained on one dataset and tested on the other seven, REATS-SFT generally achieves the lowest average MSE (Figure 4), with similar small-model results in Appendix Figure 7. Numerical baselines learn weight mappings tightly coupled to source-domain forecasts and statistics and are thus prone to overfitting. REATS instead combines textual–numerical inputs with CoT over semantic temporal properties (e.g., trend and seasonality), reducing dependence on dataset-specific distributions and yielding more transferable mappings.

4.7 Hyperparameters and Efficiency Analysis

Number of neighbors in Retrieval-Augmented Generation (RAG). Using k=3 rather than 25 neighbors lowers MSE from 0.1262 to 0.1210, indicating that a few close references anchor reasoning more precisely than many noisier ones (Table 5, using small model group).

GRPO reward coefficients. In Figure 6, G1–G6 correspond to $(\lambda_1, \lambda_2, \lambda_3)$ = (0.8,0,0.2), (0.5,0,0.5), (0,0.8,0.2), (0.3,0.3,0.4), (1,0,0), (0,1,0). Figure 6 yields three findings: **(1)** independent per-row r_{mse} supplies finer-grained implicit reward augmentation and beats r_{agg}, which aggregates all rows before computing one reward (G5 vs. G6), on every dataset; **(2)** r_{oracle} stabilizes small datasets (G1 beats G5 on Exchange), whereas pure r_{mse} excels on larger Electricity and Weather, and excessive oracle weight (G2) hurts; and **(3)** combining all rewards (G4) inherits the weaker objectives. Thus, G1 (λ_1=0.8, λ_3=0.2) balances MSE optimization and oracle guidance across dataset sizes.

Efficiency analysis. Compared with general-purpose LLMs such as GPT, Grok, and DeepSeek, REATS uses a fully fine-tunable 1.7B model deployable on a single lightweight GPU. Rule-CoT reduces data construction and training costs without sacrificing accuracy, while integer tables reduce tokens and inference latency (Tables 6(a) and 3(c)). Though slower than traditional ensembles, REATS offers a favorable trade-off between efficiency, accuracy, and interpretability.

5 Conclusion

REATS repurposes the LLM as a sample-adaptive, interpretable TSF ensemble router through hybrid textual–numerical inputs augmented with retrieval priors, scalable CoT and multi-row weight supervision, token-efficient weight outputs, and SFT-to-GRPO training. REATS outperforms competitive ensemble baselines across foundation or small models. Experiments further show that (1) reciprocal-reward GRPO refines weights beyond SFT imitation and outperforms optimization-level GRPO variants, while multi-row supervision provides denser reward signals; (2) API-free Rule-CoT matches or exceeds API-generated CoT and enables reasoning over unseen candidates; and (3) semantic reasoning improves OOD generalization and cross-dataset transfer over numerical-only baselines.

AI USE DISCLOSURE

Generative AI tools were used to improve the readability, conciseness, and grammatical quality of the manuscript. They were not used to formulate research hypotheses, develop the methodology or theoretical claims, design or conduct experiments, implement the methods, process data, interpret results, or generate reported findings. All AI-assisted language edits were reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work involves no human-subject studies or new data collection and uses only publicly available time-series benchmarks and released model checkpoints. REATS is intended for research on time-series forecasting and ensemble learning. Inaccurate forecasts or inappropriate model weighting may cause harm if deployed without validation in high-stakes domains such as energy, finance, and transportation. Real-world deployment should therefore include domain-specific evaluation, continuous monitoring, and appropriate human oversight.

REPRODUCIBILITY STATEMENT

The anonymous source code and configurations are available at https://anonymous.4open.science/r/REATS-8688. Appendix A.1 documents the datasets, candidate-model pools, baselines, implementation details, and training hyperparameters, while Appendix A.4 lists the pretrained model sources. Algorithms 1 and 2 specify Rule-CoT generation; Appendices A.7 and A.8 detail oracle-weight computation and diverse-weight construction. Reward design, parameter selection, theoretical analysis, and complete supplementary results are provided in the corresponding appendix sections.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

Dimitris Bertsimas and Leonard Boussioux. Ensemble modeling for time series forecasting: an adaptive robust optimization approach. *arXiv preprint arXiv:2304.04308*, 2023.

Shikun Chen and Nguyen Manh Luc. Rrmse voting regressor: A weighting function based improvement to ensemble regression. *arXiv preprint arXiv:2207.04837*, 2022.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *The twelfth international conference on learning representations*, 2024.

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*, pp. 10148–10167. PMLR, 2024.

Yuwei Fu, Di Wu, and Benoit Boulet. Reinforcement learning based dynamic model combination for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6639–6647, 2022.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: a family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 16115–16152, 2024.

Susan Gruber, Roger W Logan, Inmaculada Jarrín, Susana Monge, and Miguel A Hernán. Ensemble learning of inverse probability weights for marginal structural modeling in large observational datasets. *Statistics in medicine*, 34(1):106–117, 2015.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32369–32399, 2024b.

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*, 2025.

Yuntong Liu, Chunna Zhao, and Yaqun Huang. A combined model for multivariate time series forecasting based on mlp-feedforward attention-lstm. *IEEE Access*, 10:88644–88654, 2022.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The twelfth international conference on learning representations*, pp. 1–43, 2024.

Bo Lv, Nayu Liu, Chen Tang, Xin Liu, Yue Yu, and Ping Luo. Specfuse: Ensembling large language models via next-segment prediction. *arXiv preprint arXiv:2412.07380*, 2024.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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

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.

Niko Uremović, Marko Bizjak, Primož Sukič, Gorazd Štumberger, Borut Žalik, and Niko Lukač. A new framework for multivariate time series forecasting in energy management system. *IEEE Transactions on Smart Grid*, 14(4):2934–2947, 2022.

Martin Wachs. Forecasts in urban transportation planning: Uses, methods, and dilemmas. *Climatic Change*, 11(1):61–80, 1987.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Heecheol Yun, Kwangmin Ki, Junghyun Lee, and Eunho Yang. When to ensemble: Identifying token-level points for stable and fast llm ensembling. *arXiv preprint arXiv:2510.15346*, 2025.

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.

Xu Zhang, Zhengang Huang, Yunzhi Wu, Xun Lu, Erpeng Qi, Yunkai Chen, Zhongya Xue, Qitong Wang, Peng Wang, and Wei Wang. Multi-period learning for financial time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2848–2859, 2025a.

Xu Zhang, Qitong Wang, Peng Wang, and Wei Wang. A lightweight sparse interaction network for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13304–13312, 2025b.

Xu Zhang, Qitong Wang, Peng Wang, and Wei Wang. Semixer: Semantics enhanced mlp-mixer for multiscale mixing and long-term time series forecasting. In *Proceedings of the ACM Web Conference 2026*, pp. 5636–5647, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, Rong Jin, et al. Card: Channel aligned robust blend transformer for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11245–11283, 2024.

Yuzhen Zhu, Shaojie Luo, Di Huang, Weiyan Zheng, Fang Su, and Beiping Hou. Drcnn: decomposing residual convolutional neural networks for time series forecasting. *Scientific Reports*, 13(1):15901, 2023.

A Appendices

A.1 Experimental Details

A.1.1 Datasets and metrics

We evaluate on eight benchmarks (Zhang et al., 2026; Nie et al., 2022; Zhang et al., 2025b): ETTh1/h2, ETTm1/m2, Exchange, Weather, Electricity, and Traffic, covering energy, finance, meteorology, and transportation domains. All experiments are conducted on standard univariate time series forecasting, and both input and prediction length are 96, with train/val/test split ratio of 7:1:2. We report MSE and MAE. Dataset statistics are summarized in Appendix Table 7.

A.1.2 Candidate model pool for ensemble

We construct a diverse pool of forecasting models spanning two categories: (1) *Small specialized models*: TimeXer (Wang et al.), LSINet (Zhang et al., 2025b), CARD (Zhou et al., 2024), TimeMixer (Wang et al., 2024), ModernTCN (Luo & Wang, 2024), SEMixer (Zhang et al., 2026), DLinear (Zeng et al., 2023), PDF (Dai et al., 2024), PatchTST (Nie et al., 2022), and MLF (Zhang et al., 2025a); (2) *Foundation models*: MOMENT (Goswami et al., 2024), Sundial (Liu et al., 2025), Timer (Liu et al., 2024b), TIME-MOE (Shi et al., 2025), TimesFM (Das et al., 2024), MOIRAI (Woo et al., 2024), TimerXL (Liu et al., 2024a) and Chronos (Ansari et al., 2024). These models also serve as individual baselines. The corresponding pre-trained weight sources are provided in Appendix A.4.

A.1.3 ENSEMBLE BASELINES

We compare against four categories of ensemble methods: (1) *Heuristic*: uniform averaging and random weighting; (2) *Error-based*: inverse MSE weighting and optimal fixed weighting (OptW), where OptW is obtained by solving the same non-negative simplex-constrained quadratic program used for REATS oracle construction. The subscripts tr/val indicate whether the fixed weights are estimated on the training or validation set and then applied to all test samples; (3) *Neural network-based*: RLMC (Fu et al., 2022), which follows its original setup using raw historical series while sharing the same data splits, candidate forecasts, and oracle supervision as REATS; (4) *Prompt-based*: LLM zero-shot ensemble without fine-tuning (GPT-5.2/5.5, Codex, DeepSeek-V3.2, Grok-4), using the same prompt components (temporal features, tool descriptions, and RAG references) and reason-then-decide output format as REATS for fair comparison. Additionally, to validate our reward mapping design, we compare against four recent GRPO algorithmic variants (DAPO (Yu et al., 2026), DrGRPO (Liu et al.), GSPO (Zheng et al., 2025), SAPO (Gao et al., 2025)) that address reward sparsity through optimization-level modifications (Section 4.3).

A.1.4 IMPLEMENTATION DETAILS

We use Qwen3-1.7B as the base LLM. Retrieval uses $K{=}3$ similar samples, with $K'{=}10$ weight rows per sample (1 oracle + 9 diverse). Default ensemble size is 4 candidates (scalability to 2/6/8 also evaluated). For the reciprocal reward mapping, we set $k{=}20$ (chosen to jointly ensure non-vanishing far-range rewards and strong near-oracle discrimination, see Appendix A.6.1), $(\lambda_1, \lambda_2, \lambda_3){=}(0.8, 0, 0.2)$, and format penalty -0.5. The learning rate for SFT and GRPO are 1×10^{-5} and 5×10^{-6}, respectively. At inference, only the first output row is used as the final ensemble weights, since it is trained against the QP oracle and receives the strongest optimization pressure. Training is on NVIDIA A100 40G GPUs. All oracle weights used for SFT supervision and GRPO reward computation are derived from training set ground truth only.

A.2 RELATED WORK: TIME SERIES FORECASTING

In recent years, time series forecasting has shifted from traditional statistical methods to deep learning–based approaches, which can be broadly categorized into Transformer-based models, convolutional networks, and lightweight linear models.

Transformers have been widely adopted due to their strong sequence modeling capabilities. Early works directly applied them along the temporal dimension, while later studies introduced task-specific designs. For example, PatchTST (Nie et al., 2022) segments time series into patches and models channels independently. TimeXer (Wang et al.) adopts a decoder-only architecture with causal masking to capture temporal dependencies, while CARD (Zhou et al., 2024) focuses on cross-channel correlations via channel-aligned attention. TimeMixer (Wang et al., 2024) and SEMixer (Zhang et al., 2026) model temporal dynamics across multiple scales via decomposition and mixing strategies. MLF (Zhang et al., 2025a) extends this idea by introducing multi-period modeling with adaptive patching and weighted fusion. In addition, efficient alternatives have been explored. ModernTCN (Luo & Wang, 2024) uses large-kernel convolutions to model dependencies without attention, while DLinear (Zeng et al., 2023) and LSINet (Zhang et al., 2025b) demonstrate that simple linear architectures can achieve competitive performance with high efficiency.

Finally, inspired by large language models, time series foundation models learn general temporal representations via large-scale pretraining for cross-task generalization. Encoder-based methods such as MOMENT (Goswami et al., 2024) and MOIRAI (Woo et al., 2024) adopt masked modeling with patch tokenization to handle diverse and multivariate data, while decoder-based approaches including Timer (Liu et al., 2024b), TimesFM (Das et al., 2024), Sundial (Liu et al., 2025), and TIME-MOE (Shi et al., 2025) unify forecasting and related tasks under an autoregressive or generative paradigm, incorporating designs such as flexible patching, probabilistic modeling, and mixture-of-experts to improve scalability and generalization.

Algorithm 2: Rule-Based Chain-of-Thought Generation (Detailed)

Input: Temporal features $\Phi(\mathbf{X})$, tool set $\mathcal{T} = \{t_1, ..., t_N\}$ each with category $c_i \in \{\text{linear, transformer, FFT, decomp., CNN, foundation}\}$, oracle weights $\mathbf{w}^* \in \mathbb{R}^{K \times N}$, RAG features $\Phi(\mathbf{X}_{\text{rag}})$ and weights $\mathbf{w}_{\text{rag}} \in \mathbb{R}^{K' \times N}$

Output: Chain-of-thought text *CoT*

$\bar{\mathbf{w}} \leftarrow \frac{1}{K'} \sum_{k=1}^{K'} \mathbf{w}_k^*$; `// Average oracle weights across` K' `rows`

`// Step 1: Summarize key temporal patterns`

Parse $\Phi(\mathbf{X})$ into: trend strength/direction, noise level, ACF value, seasonality (yes/no), stationarity, outliers, distribution type;

foreach *attribute* a **do**

  Sample phrase from predefined template pool $\mathcal{P}_a$; `// e.g., (strong, upward)` $\rightarrow$ `''pronounced upward drift''`

end

$s_1 \leftarrow$ `''Key patterns: ''` + concatenate all sampled phrases;

`// Step 2: Oracle-guided tool-to-pattern matching`

foreach *tool* t_i *in descending order of* $\bar{w}_i$ **do**

  Retrieve category c_i and its strength descriptors;

  `// Category-specific rule matching`

  if $c_i =$ *linear* ***and*** *trend is strong* **then**

    reason $\leftarrow$ "strong trend favors linear decomposition";

  else if $c_i =$ *FFT* ***and*** *seasonality detected* **then**

    reason $\leftarrow$ "FFT decomposition captures periodic structure";

  else if $c_i =$ *transformer* ***and*** *ACF is high* **then**

    reason $\leftarrow$ "attention can exploit autocorrelation structure";

  end

  `// ... (other category-attribute rules)`

  if $\bar{w}_i > 0.35$ **then**

    $\text{sent}_i \leftarrow$ sample from high-fit templates with reason ; `// ''{tool} fits best because ...''`

  else if $\bar{w}_i > 0.15$ **then**

    $\text{sent}_i \leftarrow$ sample from mid-fit templates ; `// ''{tool} adds complementary value via ...''`

  else

    $\text{sent}_i \leftarrow$ sample from low-fit templates with mismatch reason ; `// ''{tool} has limited relevance ...''`

  end

end

$s_2 \leftarrow$ `''Tool match: ''` + join all sent_i;

`// Step 3: Compare with RAG reference`

$\bar{\mathbf{w}}_{\text{rag}} \leftarrow \frac{1}{K} \sum_k \mathbf{w}_{\text{rag},k}$;

$t_{\text{dom}} \leftarrow \arg\max_i \bar{w}_i$; $t_{\text{rag}} \leftarrow \arg\max_i \bar{w}_{\text{rag},i}$;

if $t_{dom} = t_{rag}$ **then**

  $s_3 \leftarrow$ "consistent with the current allocation direction";

else

  Compare $\Phi(\mathbf{X})$ vs $\Phi(\mathbf{X}_{\text{rag}})$ to identify feature differences;

  $s_3 \leftarrow$ "similar-series ensembles lean toward t_{rag}, but current series differs in {differences}, shifting preference toward t_{dom}";

end

$s_3 \leftarrow$ `''Reference: ''` + s_3;

`// Step 4: State allocation conclusion`

dom $\leftarrow \{t_i : \bar{w}_i > 0.35\}$; mid $\leftarrow \{t_i : 0.12 < \bar{w}_i \leq 0.35\}$; minor $\leftarrow$ rest;

$s_4 \leftarrow$ `''Conclusion: ''` + sample from conclusion templates naming dom/mid/minor;

return $CoT = s_1 \oplus s_2 \oplus s_3 \oplus s_4$;

A.3 DATASETS

The 8 public datasets used in this paper are extensively used for long-term TSF algorithm evaluation, covering multiple fields including industry (4 ETT datasets), climate (Weather), energy (Electricity), transportation (Traffic), and economy (Exchange). The detailed descriptions are as follows:

1. Electricity dataset[1] collects the electricity consumption (kWh) every 15 minutes of 321 clients from 2012 to 2014.
2. ETT datasets[2] comprises two sub-datasets, ETT1 and ETT2, collected from two separate counties. Each sub-dataset offers two versions with varying sampling resolutions (15 minutes and 1 hour). ETT dataset includes multiple time series of electrical loads and a single time sequence of oil temperature.
3. Weather dataset[3] contains 21 meteorological indicators, such as air temperature, humidity, etc, recorded every 10 minutes for the entirety of 2020.
4. Exchange dataset[4] contains the current exchange of eight countries.
5. Traffic records hourly road occupancy rates measured by 862 sensors of the San Francisco Bay area freeways in 2 years.

Table 7: Statistics of the eight benchmark datasets. *Time points* denotes the total number of observations. *Split* denotes train/validation/test sizes. *Frequency* denotes the sampling interval.

Datasets	Variable	Time points	Split (train/val/test)	Frequency
ETTh1,ETTh2	7	17,420	(8545, 2881, 2881)	Hourly
ETTm1,ETTm2	7	69,680	(34465, 11521, 11521)	15min
Weather	21	52,696	(36792, 5271, 10540)	10min
Exchange rate	8	7,588	(5120, 665, 1422)	Daily
Electricity	321	26,304	(18317, 2633, 5261)	Hourly
Traffic	862	17,544	(12185, 1757, 3509)	Hourly

A.4 FORECASTING MODEL DESCRIPTIONS

We use two groups of forecasting models as ensemble candidates. All models operate in a univariate setting with input length 96 and prediction length 96.

A.4.1 FOUNDATION MODEL WEIGHT SOURCES

- **MOIRAI**: A universal time series forecasting transformer pre-trained on large-scale data with masked token prediction. Weights: `Salesforce/moirai-1.1-R-small`[5].
- **MOMENT**: A family of foundation models for time series, pre-trained with masked reconstruction on diverse time series corpora. Weights: `AutonLab/MOMENT-1-large`[6].
- **TimeMoE**: A decoder-only Mixture-of-Experts model for time series, using sparse expert routing for multi-horizon forecasting. Weights: `Maple728/TimeMoE-50M`[7].
- **TimesFM**: A decoder-only foundation model from Google, pre-trained on 100B real-world time points with patched input tokenization. Weights: `google/timesfm-2.5-200m-pytorch`[8].

[1] https://archive.ics.uci.edu/dataset/321/electricity
[2] https://github.com/zhouhaoyi/Informer2020
[3] https://www.bgc-jena.mpg.de/wetter/
[4] https://github.com/laiguokun/multivariate-time-series-data
[5] https://huggingface.co/Salesforce/moirai-1.1-R-small
[6] https://huggingface.co/AutonLab/MOMENT-1-large
[7] https://huggingface.co/Maple728/TimeMoE-50M
[8] https://huggingface.co/google/timesfm-2.5-200m-pytorch

Table 8: Comparison of GRPO algorithm variants on three datasets (MSE ↓). All variants use the naive reward ($-\delta$). Visualized in Figure 5(d).

Dataset	GRPO with Reward Mapping	GRPO	DAPO	DrGRPO	GSPO	SAPO
Exchange	**0.1587**	0.1753	0.1714	0.1804	0.1815	0.1712
Electricity	**0.1463**	0.1503	0.1493	0.1475	0.1493	0.1481
Weather	**0.0061**	0.0167	0.0386	0.0346	0.0324	0.0452

Table 9: Average token count of the human input across different data representations.

Representation	Input=96	Input=512
Raw TS (time series as text)	1656	7480
Textual–Num (structured analysis)	1085	1085

- **Timer**: A generative pre-trained Transformer for time series with unified next-token prediction across forecasting tasks. Weights: `Timer_forecast_1.0.ckpt`[9].
- **TimerXL**: An extended version of Timer with larger model capacity. Weights: `TimerXL_forecast.pth`[10].
- **Sundial**: A causal language model for time series that discretizes continuous values via adaptive binning. Weights: `thuml/sundial-base-128m`[11].
- **Chronos**: A language modeling framework for time series using quantization-based tokenization, built on T5 architecture. Weights: `amazon/chronos-t5-base`[12].

A.5 Supplementary Experimental Results

A.5.1 Comparison with GRPO Algorithm Variants

A.5.2 OOD Model Mapping

A.5.3 Supplementary OOD Experiments (Table 16 and Table 17)

A.5.4 Supplementary Results about Scalability to Varying Numbers of Candidate Models (Table 18 and Table 19)

A.6 Discussion of Reward Mapping Function Designs

A.6.1 Selection of Reward Scaling Factor k

We evaluate the three bounded reward mapping functions that involve a scaling factor k (the naive $r=-\delta$ has no such parameter), where $\delta = \mathcal{L}(\mathbf{w}) - \mathcal{L}(\mathbf{w}^*)$ denotes the forecasting MSE gap between the predicted ensemble weights and the oracle weights. We define a unified scoring framework with three criteria:

1. **Near-oracle sensitivity**: $r(0) - r(0.01) \in [0.05, 0.20]$, ensuring the function can distinguish near-optimal candidates without over-sensitivity.
2. **Mid-range signal**: $r(0.1) \in [0.20, 0.50]$, maintaining meaningful learning signal while sufficiently penalizing suboptimal generations.
3. **Far-range signal**: $r(0.3) > 0.05$, ensuring that poorly-performing generations still receive non-zero reward differences for policy optimization.

Figure 8 shows the reward curves under varying k. By checking which k values simultaneously satisfy all three criteria, we identify the valid range for each function and select k near the center: k=20 for reciprocal (valid range $[10, 25]$, widest among all three), k=10 for exponential (valid range $[7, 12]$, narrow), and k=2 for linear (to maximize coverage before the dead zone at $\delta \geq 1/k$).

[9] https://github.com/thuml/OpenLTM
[10] https://github.com/thuml/OpenLTM
[11] https://huggingface.co/thuml/sundial-base-128m
[12] https://huggingface.co/amazon/chronos-t5-base

Table 10: Input ablation study (REATS-SFT, MSE ↓). (a) In-distribution evaluation on small model candidates. (b) OOD evaluation: trained on small models, tested on foundation models. Best in **bold**.

(a) Small Model Candidates (ID)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Textual–Num (**Ours**)	**.1888**	.1492	**.1769**	.0785	.0792	**.0424**	.1605	.0925	**.1210**
Raw TS (text)	.1988	.1487	.1787	.0770	.0790	.0475	.1595	.0935	.1228
Raw TS (tsenc)	.1968	**.1451**	.1778	**.0756**	**.0760**	.0480	**.1562**	**.0922**	**.1210**

(b) Small → Zeroshot (OOD)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Textual–Num	.1579	**.1442**	**.1925**	.0892	.1165	.0098	.3972	**.3640**	**.1839**
Raw TS (text)	.1583	.1450	.1937	.0894	.1183	.0211	**.3967**	.3677	.1863
Raw TS (tsenc)	**.1501**	.1448	.1945	**.0874**	**.1118**	**.0089**	.4042	.6306	.2165

Table 11: Model mapping for OOD generalization experiments (Table 3). Model 1–4 are unseen foundation models used for each dataset, with varying candidate sets to cover a broader range of foundation models. **Bold** indicates models not seen during training.

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	**MOMENT**	**TimeMoE**	**TimesFM**	**MOIRAI**
ETTh1	**MOMENT**	**Sundial**	**TimesFM**	**MOIRAI**
ETTh2	**MOMENT**	**Sundial**	**TimesFM**	**MOIRAI**
ETTm1	**MOMENT**	**Sundial**	**TimesFM**	**MOIRAI**
ETTm2	**MOMENT**	**Timer**	**TimeMoE**	**MOIRAI**
Weather	**MOMENT**	**Timer**	**TimeMoE**	**TimesFM**
Electricity	**MOMENT**	**Sundial**	**Timer**	**MOIRAI**
Traffic	**MOMENT**	**Sundial**	**Timer**	**TimeMoE**

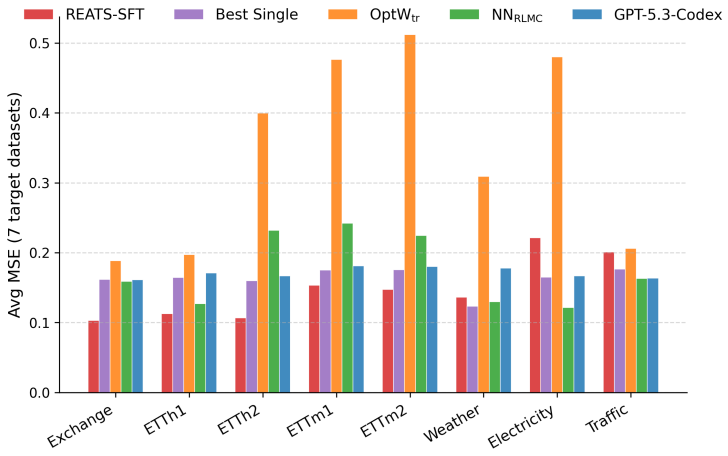


Figure 7: Transfer learning comparison (small model group, MSE ↓). Each source dataset (X-axis) is used for training, and the average MSE across the other 7 target datasets (Y-axis) is reported. Baselines are the best method from each category.

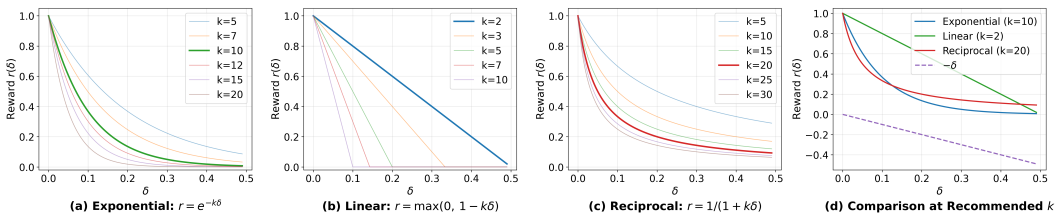


Figure 8: Reward curves under varying k for (a) exponential, (b) linear, and (c) reciprocal mapping functions. Bold lines indicate the selected k values. (d) Comparison of all four mappings at their recommended k.

Table 12: OOD generalization to mixed models (MSE ↓). Trained on small model candidates (CARD, LSINet, PDF, TimeXer), evaluated on mixed candidates (2–3 unseen per dataset). **Bold** in mapping table indicates unseen models. The MAE results are shown in appendix Table 15.

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.1690	**.1515**	1.8930	**.0815**	1.9402	.4372	1.7326	**.0667**	.8090
Model 2	.1675	1.9515	**.2022**	.1124	**.0729**	.7114	.7635	.9884	.6212
Model 3	.3037	1.9124	2.0073	2.0260	1.9540	.5683	**.1472**	2.1670	1.3857
Model 4	1.5737	.1746	1.9351	1.8879	.2862	.4397	1.7626	2.1059	1.2707
Traditional Ensemble									
Ens_{avg}	.2534	.4671	.8089	.4161	.4371	.2702	.6257	.7310	.5012
Ens_{rand}	.3149	.5819	.9487	.5392	.5611	.3393	.7218	.8503	.6071
$\text{InvMSE}_{\text{tr}}$	.1688	.1611	.2467	.0860	.0840	.2739	.1914	.0786	.1613
$\text{InvMSE}_{\text{val}}$	.2427	.1922	.3573	.1034	.0829	.3331	.2199	.0765	.2010
RLMC	.2562	.1739	.2023	.1087	.0736	.2617	.1513	.0667	.1583
OptW_{tr}	.1594	.1632	.2094	.0848	.0792	.2598	.1562	.0827	.1493
OptW_{val}	.1937	.1732	.2103	.1103	.0792	.3172	.1586	.0825	.1656
LLM-based Ensemble									
GPT-5.2	.1760	.3551	.5593	.2701	.2608	.2747	.3541	.2109	.3076
Codex	.1671	.3235	.5333	.2039	.2263	.1939	.3156	.1843	.2685
GPT-5.5	.1964	.3982	.5998	.2758	.3103	.3162	.3746	.2945	.3457
DeepSeek-V3.2	.1724	.3556	.5755	.1904	.2783	.1588	.3074	.2001	.2798
Grok-4	.1828	.3646	.5754	.2452	.3072	.2771	.4228	.2986	.3342
REATS (Ours)									
REATS-GRPO	**.1370**	.1798	.2568	.0816	.0761	**.0218**	.1522	.0791	**.1231**

Model Mapping

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	CARD	**MOIRAI**	**MOMENT**	**ModernTCN**
ETTh1	PDF	**TimeMixer**	TimeXer	**Timer**
ETTh2	**PatchTST**	**Sundial**	**TimeMixer**	TimeXer
ETTm1	LSINet	**MOMENT**	**PatchTST**	TimeXer
ETTm2	**ModernTCN**	PDF	**PatchTST**	**TimesFM**
Weather	CARD	**MLF**	**MOMENT**	**Timer**
Electricity	**DLinear**	**MOMENT**	PDF	**TimeMixer**
Traffic	CARD	**MOMENT**	**SEMixer**	**TimeMixer**

A.6.2 Reward Sensitivity Analysis

To formalize the discriminative power of each reward function, we analyze their derivatives with respect to δ. By the first-order Taylor expansion:

$$r(\delta + \Delta\delta) - r(\delta) = r'(\delta)\,\Delta\delta + o(\Delta\delta), \quad \Delta\delta \to 0 \tag{5}$$

For two nearby points δ_1, δ_2, the mean value theorem guarantees the existence of $\xi \in (\delta_1, \delta_2)$ such that

$$r(\delta_1) - r(\delta_2) = r'(\xi)\,(\delta_1 - \delta_2) \tag{6}$$

Therefore, the local reward difference can be approximated as:

$$|r(\delta_1) - r(\delta_2)| \approx \left|\frac{dr}{d\delta}\right| \cdot |\delta_1 - \delta_2| \tag{7}$$

This shows that $|dr/d\delta|$ directly quantifies the *local reward sensitivity*: given the same performance difference $|\Delta\delta|$ between two candidates, a larger derivative produces a larger raw reward gap, contributing to better discrimination among candidates in that δ region.

The derivatives of the three reward functions are:

$$\text{Reciprocal:} \quad \left|\frac{dr}{d\delta}\right| = \frac{k}{(1+k\delta)^2} \tag{8}$$

$$\text{Exponential:} \quad \left|\frac{dr}{d\delta}\right| = k \cdot e^{-k\delta} \tag{9}$$

$$\text{Linear:} \quad \left|\frac{dr}{d\delta}\right| = \begin{cases} k & \delta < 1/k \\ 0 & \delta \geq 1/k \end{cases} \tag{10}$$

At their recommended k values, the three functions exhibit distinct sensitivity profiles:

- **Reciprocal** (k=20): Maintains effective sensitivity across a wide δ range (polynomial decay ensures no dead zones even at moderate δ). **Under GRPO's constraint that the reward mapping should provide signal across the full group diversity, the reciprocal also achieves the highest near-oracle sensitivity** ($\delta < 0.03$), preserving higher raw reward resolution near the oracle as weights approach optimality. In contrast, other functions produce diminished or zero reward signals at moderate δ, limiting their near-oracle k choices.
- **Exponential** (k=10): Provides stronger discrimination in the mid-range ($\delta \in [0.034, 0.34]$), but its signal vanishes rapidly beyond this range, and weaker near-oracle sensitivity limits the final refinement precision.
- **Linear** (k=2): Maintains constant sensitivity within its narrow active region $[0, 1/k]$, but offers zero signal beyond the cutoff, severely restricting its effective range.

A.6.3 Theoretical analysis

The empirical analysis above demonstrates that the reciprocal mapping outperforms alternatives. In this section, we provide theoretical justification by analyzing how the reward mapping function affects the GRPO learning signal. Since GRPO continues from an SFT-initialized model, the policy starts from a coarse initialization that GRPO must further refine.

GRPO signal and reward variance. In GRPO, the policy gradient for generation i within a prompt group is scaled by the normalized advantage:

$$\hat{A}_i = \frac{r_i - \mu_r}{\sigma_r}, \quad \text{where } \mu_r = \frac{1}{G}\sum_j r_j, \;\; \sigma_r = \sqrt{\frac{1}{G}\sum_j (r_j - \mu_r)^2} \tag{11}$$

where G is the group size. A necessary condition for reward-driven learning is $\sigma_r > 0$: when all rewards are identical ($\sigma_r = 0$), no reward-driven policy-gradient update arises from that group. To understand how the choice of reward mapping affects σ_r, we apply a first-order Taylor expansion of $r(\delta)$ around the group mean $\bar{\delta}$:

$$\sigma_r \approx |r'(\bar{\delta})| \cdot \sigma_\delta \tag{12}$$

where σ_δ is the standard deviation of δ values within the generation group (controlled by sampling temperature). This decomposes the GRPO learning signal into two independent factors:

- $|r'(\bar{\delta})|$: the reward function's **local sensitivity** at the current training regime
- σ_δ: the **generation diversity** (controlled by sampling hyperparameters)

Under the first-order approximation, $|r'(\bar{\delta})| > 0$ and $\sigma_\delta > 0$ yield non-zero local reward dispersion. Moreover, while advantage normalization prevents the raw derivative magnitude from proportionally scaling the learning signal, a mapping with stronger near-oracle sensitivity and bounded range produces better-separated normalized advantages in mixed-quality groups (see Appendix A.6.4 for quantitative analysis).

Connection to GRPO's within-group advantage separation. The normalized advantage gap between two candidates i, j in a GRPO group is:

$$\hat{A}_i - \hat{A}_j = \frac{f(\delta_i) - f(\delta_j)}{\sigma_{f(\delta)}} \tag{13}$$

For any affine mapping $f(\delta) = -a\delta + b$ ($a > 0$), the scaling factor a cancels between numerator and denominator, leaving $\hat{A}_i - \hat{A}_j = (\delta_j - \delta_i)/\sigma_\delta$ regardless of a. Thus, affine reward transformations

do not alter the normalized advantage gaps within a group. However, for a nonlinear bounded mapping such as $f(\delta) = 1/(1 + k\delta)$, both the numerator and $\sigma_{f(\delta)}$ depend nonlinearly on the entire group's δ distribution, which changes the advantage separation among candidates. In particular, in mixed-quality groups where some $\delta_{\text{bad}} \gg \delta_i, \delta_j$, the unbounded $f = -\delta$ allows $\sigma_{f(\delta)}$ to grow without limit, driving $|\hat{A}_i - \hat{A}_j| \to 0$; the bounded reciprocal prevents this compression (see Appendix A.6.4).

Since GRPO generates multiple candidate outputs per prompt via sampling, the δ values within each group can span a wide range. The reward mapping must maintain adequate sensitivity across this range. If the mapping's derivative vanishes in certain δ regions ("dead zones"), two failure cases arise: (1) if *all* group members fall in the dead zone, all rewards become identical, $\sigma_r = 0$, and no reward-driven learning signal; (2) if *some* group members fall in the dead zone, their rewards collapse to indistinguishable values, and those generations provide no reward-driven learning signal.

Derivative comparison. We compare the sensitivity $|r'(\delta)|$ of each mapping:

$$|r'_{\text{rec}}(\delta)| = \frac{k}{(1 + k\delta)^2} = O(1/\delta^2) \quad \text{(polynomial decay)} \tag{14}$$

$$|r'_{\text{exp}}(\delta)| = k \cdot e^{-k\delta} = O(e^{-k\delta}) \quad \text{(exponential decay)} \tag{15}$$

$$|r'_{\text{lin}}(\delta)| = \begin{cases} k & \text{if } \delta < 1/k \\ 0 & \text{if } \delta \geq 1/k \end{cases} \quad \text{(hard cutoff)} \tag{16}$$

The three mappings exhibit fundamentally different decay behaviors. The reciprocal decays polynomially, meaning its sensitivity decreases slowly and all samples in the group remain distinguishable in raw reward space regardless of their δ value. The exponential decays much faster, causing samples with moderate-to-large δ to receive near-identical rewards and become indistinguishable. The linear has a hard cutoff at $\delta = 1/k$, beyond which sensitivity is exactly zero and all such samples are mapped to identical rewards.

Absence of a hard reward dead zone. We formally verify that the reciprocal mapping maintains distinct rewards for any pair of non-identical δ values. For $r(\delta) = \frac{1}{1+k\delta}$ with $k > 0$, since $(1 + k\delta)^2 > 0$ for all $\delta \geq 0$, **the derivative $|r'_{\text{rec}}(\delta)| = k/(1 + k\delta)^2 > 0$ for all $\delta \geq 0$.** Because $r(\delta)$ is strictly monotonically decreasing, $\delta_i \neq \delta_j \implies r(\delta_i) \neq r(\delta_j)$, so any generation group with $\sigma_\delta > 0$ yields $\sigma_r > 0$ regardless of $\bar{\delta}$.

Positive lower bound on reward dispersion. Beyond the qualitative guarantee above, we derive a quantitative lower bound on σ_r. Let $D = \max_{i,j} |r(\delta_i) - r(\delta_j)|$ be the maximum reward difference in the group. Since μ_r lies between $\min_i r_i$ and $\max_i r_i$, at least one term satisfies $(r_i - \mu_r)^2 \geq D^2/4$, giving:

$$\sigma_r^2 = \frac{1}{G}\sum_i (r_i - \mu_r)^2 \geq \frac{D^2}{4G} \implies \sigma_r \geq \frac{D}{2\sqrt{G}} \tag{17}$$

To bound D from below, we apply the mean value theorem: for any δ_i, δ_j, there exists $\xi \in [\delta_i, \delta_j]$ such that:

$$|r(\delta_i) - r(\delta_j)| = |r'(\xi)| \cdot |\delta_i - \delta_j| \tag{18}$$

Since $|r'_{\text{rec}}(\delta)| = k/(1 + k\delta)^2$ is monotonically decreasing and $\xi \leq \delta_{\max}$:

$$|r'(\xi)| \geq |r'(\delta_{\max})| = \frac{k}{(1 + k\delta_{\max})^2} \tag{19}$$

Combining the above:

$$\sigma_r \geq \frac{k}{2\sqrt{G}\,(1 + k\delta_{\max})^2} \cdot \max_{i \neq j} |\delta_i - \delta_j| \tag{20}$$

This lower bound confirms that **the reciprocal mapping maintains adequate sensitivity at both small δ (where $|r'| \approx k$) and moderate δ (where $|r'| = k/(1 + k\delta)^2$ remains positive)**, providing effective discrimination across the full δ range encountered during GRPO training.

Failure modes of alternative mappings. Applying the same mean value theorem-based derivation to the alternative mappings yields analogous bounds that reveal their limitations. For the **exponential**

mapping $r_{\text{exp}}(\delta) = e^{-k\delta}$, since $|r'_{\text{exp}}(\xi)| = ke^{-k\xi}$ is monotonically decreasing, $|r'_{\text{exp}}(\xi)| \geq k \cdot e^{-k\delta_{\max}}$ for $\xi \leq \delta_{\max}$, the analogous bound is:

$$\sigma_r \geq \frac{k \cdot e^{-k\delta_{\max}}}{2\sqrt{G}} \cdot \max_{i \neq j} |\delta_i - \delta_j| \tag{21}$$

The factor $e^{-k\delta_{\max}}$ decays exponentially, making this bound vanishingly small even at moderate $\delta_{\max}$ (e.g., with $k\delta_{\max} = 5$, the factor is ≈ 0.007). Practically, effective raw-reward saturation ($|r'| < 0.01$) begins at $\delta \approx \ln(100k)/k$, beyond which all samples become effectively indistinguishable.

For the **linear** mapping $r_{\text{lin}}(\delta) = \max(0, 1 - k\delta)$, when $\delta_{\max} > 1/k$, the derivative $|r'_{\text{lin}}(\xi)| = 0$ for $\xi > 1/k$, so the same bound degenerates to:

$$\sigma_r \geq 0 \tag{22}$$

which is trivially true and provides no guarantee. When $\min_i \delta_i > 1/k$, all rewards are exactly zero and $\sigma_r = 0$.

In contrast, the reciprocal's bound has only polynomial decay $(1 + k\delta_{\max})^2$ in the denominator, ensuring a strictly positive lower bound for any finite $\delta_{\max}$. Among the three mappings considered and under the rollout ranges studied, the reciprocal is the most robust: it decays polynomially rather than exponentially, avoiding both the hard dead zone of the linear mapping and the rapid effective saturation of the exponential mapping.

A.6.4 Why naive reward $r = -\delta$ fails in practice

While the naive mapping $r(\delta) = -\delta$ does not suffer from dead-zone issues, it introduces two problems under GRPO's advantage normalization $\hat{A}_i = (r_i - \mu_r)/\sigma_r$: (1) as a linear function, it produces uniform advantage spacing regardless of candidate quality, whereas the convex reciprocal concentrates discrimination among near-optimal candidates; (2) its unbounded range allows a single outlier to dominate σ_r, compressing all near-optimal advantages to near-identical values.

	$r = -\delta$	Reciprocal $r = 1/(1 + 20\delta)$
Top-2 gap (δ=0.001 vs 0.01)	0.009	0.147
Bottom-2 gap (δ=0.5 vs 2.0)	1.500	0.067
Normalized $\hat{A}_1 - \hat{A}_2$	0.012	0.379

In a realistic mixed-quality group ($\delta \in \{0.001, 0.01, 0.1, 0.5, 2.0\}$), the reciprocal achieves $32\times$ larger normalized advantage separation between the best two candidates, because it simultaneously bounds outlier rewards near zero (preventing σ_r inflation) and nonlinearly stretches near-oracle differences. This effect cannot be replicated by any affine transformation of $r = -\delta$.

A.7 Derivation of Closed-Form Oracle Weights

Given N base forecasters, let $\hat{\mathbf{y}}_i \in \mathbb{R}^T$ denote the prediction of model i and $\mathbf{y} \in \mathbb{R}^T$ the ground truth, where T is the prediction horizon. We define the per-model error as $\mathbf{e}_i = \hat{\mathbf{y}}_i - \mathbf{y}$.

The ensemble prediction under weights $\mathbf{w}$ is $\hat{\mathbf{y}} = \sum_{i=1}^{M} w_i \hat{\mathbf{y}}_i$. Since $\sum_{i=1}^{N} w_i = 1$, we can write $\mathbf{y} = \sum_{i=1}^{M} w_i \mathbf{y}$, and the ensemble error becomes:

$$\mathbf{e}_{\text{ens}} = \hat{\mathbf{y}} - \mathbf{y} = \sum_{i=1}^{M} w_i \hat{\mathbf{y}}_i - \sum_{i=1}^{M} w_i \mathbf{y} = \sum_{i=1}^{M} w_i (\hat{\mathbf{y}}_i - \mathbf{y}) = \sum_{i=1}^{M} w_i \mathbf{e}_i \tag{23}$$

Table 13: Summary of the time-series characteristic analysis tools.

Feature Group	Key features	Description
Stationarity ϕ_{sta}	ADF test result	Tests whether the series contains a unit root, i.e., non-stationary time series.
	KPSS test result	Tests whether a series is stationary around its "trend" or "mean".
Noise ϕ_{noise}	Coefficient of variation	Captures relative variability compared to the mean level.
	Noise proportion	Estimates how much variance is attributed to irregular components.
	Noise intensity	Reflects predictability difficulty due to random fluctuations.
Trend ϕ_{trend}	Trend slope	Measures long-term directional movement over time.
	R^2 (Goodness of fit)	Indicates how well a linear trend explains the series.
	Trend type	Identifies whether a clear increasing or decreasing pattern exists.
	Trend strength	Quantifies the dominance of trend relative to fluctuations.
Seasonality ϕ_{sea}	Period length	Indicates repeating temporal cycle length.
	Seasonal variance ratio	Measures contribution of periodic components.
	Seasonality strength	Assesses whether periodic patterns are dominant.
Autocorrelation ϕ_{ac}	Significant ACF/PACF lags	The number of lags outside the confidence interval in the ACF (Autocorrelation Function) and Partial ACF.
	ACF/PACF coefficients	Autocorrelation coefficients at each lag.
	Autocorrelation judgment	Evaluates overall temporal dependency strength.
Statistics ϕ_{stat}	Mean / Median	Describe central tendency and distribution symmetry.
	Standard deviation	Measures overall dispersion of values.
	Minimum / Maximum	Define the range of observed values.
	Quantiles (Q1, Q2, Q3)	Describe distribution spread and skewness.
	Percentiles	Capture tail behavior of the distribution.
	Central tendency consistency	Assesses symmetry between mean and median.
IQR Outliers ϕ_{out}	Lower threshold	Defines lower bound for outlier detection.
	Upper threshold	Defines upper bound for outlier detection.
	Lower outlier count	Counts extreme low values.
	Upper outlier count	Counts extreme high values.
	Outlier ratio	Reflects prevalence of rare extreme events.
Distribution ϕ_{dist}	Normality test	Assesses deviation from Gaussian assumptions.
	Skewness	Measures asymmetry of the distribution.
	Kurtosis	Captures tail heaviness and peak sharpness.
	Distribution type	Summarizes overall distribution shape.

The ensemble MSE is:

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{M} w_i\, e_{i,t}\right)^2 \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{M} w_i\, w_j\, e_{i,t}\, e_{j,t} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{M} w_i\, w_j \underbrace{\frac{1}{T}\sum_{t=1}^{T} e_{i,t}\, e_{j,t}}_{G_{ij}} \\ &= \mathbf{w}^\top \mathbf{G}\mathbf{w} \end{aligned} \tag{24}$$

where $\mathbf{G} \in \mathbb{R}^{M\times M}$ with $G_{ij} = \frac{1}{T}\sum_{t=1}^{T} e_{i,t}\, e_{j,t}$. Thus, minimizing the ensemble MSE is equivalent to minimizing the quadratic form $\mathbf{w}^\top \mathbf{G}\mathbf{w}$.

We minimize $\mathbf{w}^\top \mathbf{G}\mathbf{w}$ subject to the constraint $\sum_{i=1}^{M} w_i = 1$. Introducing a Lagrange multiplier λ:

$$\mathcal{J}(\mathbf{w}, \lambda) = \mathbf{w}^\top \mathbf{G}\mathbf{w} - \lambda(\mathbf{1}^\top \mathbf{w} - 1) \tag{25}$$

Taking the derivative with respect to $\mathbf{w}$ and setting it to zero:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{w}} = 2\mathbf{G}\mathbf{w} - \lambda \mathbf{1} = \mathbf{0} \quad \Longrightarrow \quad \mathbf{w} = \frac{\lambda}{2}\mathbf{G}^{-1}\mathbf{1} \tag{26}$$

Substituting into the constraint $\mathbf{1}^\top \mathbf{w} = 1$:

$$\mathbf{1}^\top \left(\frac{\lambda}{2}\mathbf{G}^{-1}\mathbf{1}\right) = 1 \quad \Longrightarrow \quad \frac{\lambda}{2} = \frac{1}{\mathbf{1}^\top \mathbf{G}^{-1}\mathbf{1}} \tag{27}$$

Substituting back:

$$\mathbf{w}^* = \frac{\mathbf{G}^{-1}\mathbf{1}}{\mathbf{1}^\top \mathbf{G}^{-1}\mathbf{1}} \tag{28}$$

This is computed independently for each sample (each sample has its own $\mathbf{G}$).

Since per-sample $\mathbf{G}$ may be ill-conditioned, we regularize $\mathbf{G} \leftarrow \mathbf{G} + \epsilon \mathbf{I}$ (with $\epsilon = 10^{-10}$). We then solve the simplex-constrained QP $\mathbf{w}^* = \arg\min_{\mathbf{w} \geq 0,\, \mathbf{1}^\top \mathbf{w} = 1} \mathbf{w}^\top \mathbf{G}\mathbf{w}$ to obtain the exact non-negative optimal weights. The closed-form solution above serves as the initial point.

A.8 Diverse Weight Selection Procedure

As discussed in the main text, since relying solely on the single optimum $\mathbf{w}^*$ as supervision may provide insufficient supervision signals and degrade fine-tuning performance, we generate $K'-1=9$ diverse yet high-quality weight vectors per sample. We sample candidates via a Dirichlet distribution centered on $\mathbf{w}^*$ and select K vectors through a multi-objective greedy procedure that minimizes:

$$s(\mathbf{w}) = \frac{\mathcal{L}(\mathbf{w}) - \mathcal{L}^*}{\mathcal{L}^*} - d_{\min}(\mathbf{w}, \mathcal{S}) - H(\mathbf{w}) \tag{29}$$

where $\mathcal{L}(\mathbf{w})$ is the ensemble MSE under weight $\mathbf{w}$, $\mathcal{L}^*$ is the optimal MSE, $d_{\min}(\mathbf{w}, \mathcal{S}) = \min_{\mathbf{w}' \in \mathcal{S}} \|\mathbf{w} - \mathbf{w}'\|_2$ is the minimum ℓ_2 distance to the already-selected set $\mathcal{S}$, and $H(\mathbf{w}) = -\sum_i w_i \log w_i$ encourages smooth non-sparse distributions.

This diverse supervision enables the LLM to learn a distribution of reasonable weight configurations rather than overfitting to a single solution, improving generalization to unseen time series patterns. For SFT, it exposes the model to multiple valid solutions per sample, preventing memorization of a single weight pattern. For GRPO, it provides row-wise oracle references (Section 3.4) that enable fine-grained per-row reward computation, yielding richer and more informative gradient signals than a single-row comparison.

A.9 SFT Training Data Example

We provide a representative SFT training sample below to illustrate the input–output format. Each sample consists of a system prompt, a structured user input, and the model's chain-of-thought response with ensemble weights.

System prompt:

```
You are a time series ensemble expert.  Generate a weight
table (K' rows × N candidate models (tools)) for ensemble.
Each row:  all weights non-negative and sum to 1.
```

User input (structured prompt with dataset context, series analysis, tool descriptions, and RAG-retrieved similar samples):

```
Assign ensemble weights to the forecasting tools based on
the information below.
```

```
Dataset: Exchange (daily exchange rates of 8 countries).
Task: predict the next 96 values from the historical data.
Historical series analysis:
[stationarity] ADF p-value: 0.69, ADF stationary: False,
KPSS stationary: False, Overall judgment: Non-stationary
[trend] Trend slope: 0.0113, R^2: 93%, Trend type: Upward
trend, Trend strength: Strong
[noise] Noise ratio: 18%, Noise strength: Low noise
[autocorrelation] Dominant ACF lag: 1, Dominant ACF value:
0.972, Judgment: Significant autocorrelation exists
[distribution] Normal: False, Skewness: -0.089, Kurtosis:
-1.53, Type: Left-skewed (non-normal)
[seasonality] ...   [statistics] ...   [outliers] ...
Available forecasting tools:
TimeXer: decoder-only Transformer with patch tokenization
and causal masking.
PDF: FFT-based periodicity decoupling with parallel
Transformer and conv branches.
CARD: channel-dependent Transformer with cross-channel
attention and token blending.
LSINet: lightweight linear model with sparse temporal
connections via Bernoulli interaction.
Similar sample analysis: [stationarity] Non-stationary,
[trend] Strong upward, [noise] Low ...
Optimal ensemble weights(%) on the most similar time series:
CARD,LSINet,PDF,TimeXer
4,33,35,28 | 6,43,25,26 | 5,39,25,31 | ...
```

Model output (chain-of-thought reasoning followed by weight predictions):

```
<think>
Key patterns: non-stationary series, pronounced
upward drift, clean signal with little noise, extremely
strong short-lag persistence, no outliers, left-skewed
distribution.
Tool match: PDF aligns strongly since FFT periodicity
decoupling fits the series; LSINet contributes through
strong trend favoring linear decomposition; CARD and TimeXer
have limited relevance.
Reference: similar-series ensembles lean toward LSINet with
moderate PDF, but current characteristics better suit PDF.
Conclusion: prioritize PDF, hedge with LSINet, token
allocation to CARD and TimeXer.
</think>
Optimal ensemble weights(%) on the historical time series:
CARD,LSINet,PDF,TimeXer
5,12,71,12 | 13,40,33,14 | 19,18,48,15 | 15,29,42,14 | ...
```

A.10 More discussions about Interpretability Analysis and GRPO Reasoning Improvement

A.10.1 Chain-of-thought interpretability

Both REATS-SFT and REATS-GRPO generate structured chain-of-thought reasoning before producing ensemble weights. The reasoning follows a consistent four-step structure: (1) summarize key series patterns, (2) match patterns to tool capabilities, (3) reference similar historical cases from RAG, and (4) state the allocation conclusion. This structure enables users to understand *why* specific tools receive higher weights, making the ensemble decision transparent and auditable.

Table 14: Ensemble forecasting results (MAE ↓). Dataset abbreviations: Exch=Exchange, H1/H2=ETTh1/ETTh2, M1/M2=ETTm1/ETTm2, Wea=Weather, Elec=Electricity, Traf=Traffic.

(a) Foundation Model Candidates

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
MOIRAI	.2810	.3476	.3588	.2662	.3957	.0584	.5800	.2695	.3196
MOMENT	.3845	.3428	.3990	.2516	.3140	.0922	.7105	.8640	.4198
TimeMoE	.3341	.3199	.3238	.2311	.2731	.0586	.4489	.1894	.2724
TimesFM	.2984	.3117	.3423	.2506	.3575	.0431	.5040	.1932	.2876
Traditional Ensemble									
Ens_{avg}	.3026	.3060	.3338	.2300	.3046	.0602	.5145	.3160	.2960
Ens_{rand}	.3071	.3114	.3386	.2341	.3112	.0608	.5248	.3295	.3022
$\text{InvMSE}_{\text{tr}}$	.2950	.3057	.3320	.2292	.2930	.0569	.4961	.1973	.2756
$\text{InvMSE}_{\text{val}}$	.3016	.3057	.3323	.2292	.2928	.0479	.4958	.1958	.2751
RLMC	.2979	.3045	.3272	.2229	.3016	.0409	.4488	.1885	.2666
OptW_{tr}	.2833	.3042	.3240	.2250	.2748	.0540	.4544	.1967	.2645
OptW_{val}	.3071	.3049	.3262	.2249	.2747	.0455	.4541	.1968	.2668
LLM-based Ensemble									
GPT-5.2	.2728	.2915	.3193	.2194	.2850	.0492	.4741	.2244	.2670
Codex	.2701	.2894	.3189	.2167	.2783	.0514	.4728	.2217	.2649
GPT-5.5	.2756	.2954	.3234	.2220	.2916	.0566	.4822	.2420	.2736
DeepSeek-V3.2	.2631	.2868	.3182	.2155	.2783	.0498	.4739	.2408	.2658
Grok-4	.2780	.2950	.3249	.2224	.2909	.0592	.4941	.2605	.2781
REATS (Ours)									
REATS-SFT	.2545	.2815	.3125	.2116	.2681	.0391	.4521	.1968	.2520
REATS-GRPO	**.2464**	**.2751**	**.3031**	**.2071**	**.2597**	**.0367**	**.4397**	**.1839**	**.2440**

(b) Small Model Candidates

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
CARD	.2849	.3087	.3278	.2289	.2535	.0608	.4929	.1708	.2660
LSINet	.4962	.2992	.3137	.2084	.1993	.1165	.3730	1.1956	.4002
PDF	.5316	.2958	.3183	.2022	.1877	.0919	.2810	1.1898	.3873
TimeXer	.9606	1.0635	1.1316	1.0863	1.1301	.0827	1.0080	1.1635	.9533
Traditional Ensemble									
Ens_{avg}	.4306	.3842	.4135	.3330	.3456	.0821	.4291	.7684	.3983
Ens_{rand}	.4637	.4125	.4423	.3570	.3695	.0840	.4544	.8007	.4230
$\text{InvMSE}_{\text{tr}}$	.3143	.2955	.3164	.2050	.1961	.0830	.3061	.1965	.2391
$\text{InvMSE}_{\text{val}}$	.4237	.3053	.3493	.2169	.1961	.0811	.3169	.1923	.2602
RLMC	.4439	.3005	.4371	.2277	.1993	.0765	.2888	**.1707**	.2681
OptW_{tr}	.2824	.2996	.3163	.2055	.1930	.0821	.2812	.2197	.2350
OptW_{val}	.3658	.3050	.3280	.2258	.1912	.0810	.2860	.2195	.2503
LLM-based Ensemble									
GPT-5.2	.3745	.3286	.3518	.2568	.2563	.0634	.3439	.4405	.3020
Codex	.3572	.3146	.3422	.2417	.2438	.0636	.3332	.3826	.2849
GPT-5.5	.3839	.3347	.3570	.2635	.2620	.0686	.3551	.5058	.3163
DeepSeek-V3.2	.3581	.3109	.3371	.2395	.2386	.0615	.3214	.3742	.2802
Grok-4	.3837	.3307	.3604	.2653	.2679	.0750	.3529	.5550	.3239
REATS (Ours)									
REATS-SFT	.3028	.2916	.3144	.2087	.2019	.0472	.2925	.2316	.2363
REATS-GRPO	**.2802**	**.2823**	**.3071**	**.1964**	**.1865**	**.0406**	**.2800**	.2117	**.2231**

We illustrate with a Weather test sample (non-stationary, downward trend, low noise, ACF lag-1 = 0.987):

REATS-SFT:

```
Key patterns: non-stationary, marked downward trajectory,
very little noise, extremely strong short-lag persistence.
Tool match: PDF fits best because FFT periodicity
decoupling aligns with series characteristics; CARD can
help modestly through attention exploiting autocorrelation.
Conclusion: allocate heavily to PDF, keep moderate weight
on CARD.
Weights: CARD=32%, LSINet=5%, PDF=56%, TimeXer=7%
```

REATS-GRPO:

```
Key patterns: non-stationary, marked downward trajectory,
very little noise, extremely strong short-lag persistence.
Tool match: CARD is well-suited here as attention can
exploit autocorrelation structure; PDF adds complementary
value via FFT periodicity decoupling.
Conclusion: prioritize CARD given its fit, hedge with PDF,
LSINet.
Weights: CARD=53%, LSINet=18%, PDF=24%, TimeXer=5%
```

Both outputs provide clear, interpretable reasoning that links observed series properties to specific tool capabilities, demonstrating the inherent interpretability of the REATS framework regardless of the training stage.

A.10.2 GRPO REASONING IMPROVEMENT OVER SFT

Comparing the two outputs above reveals that GRPO refines the feature-to-tool mapping through RL exploration. The oracle weights for this sample are: CARD=49%, PDF=38%, TimeXer=7%, LSINet=6%.

The dominant characteristic of this series is its extremely strong autocorrelation (0.987), which favors attention-based models that can directly exploit lag dependencies. GRPO correctly identifies this and allocates CARD as the primary tool (53% vs. oracle 49%). SFT instead defaults to PDF (56%), which targets periodicity, a less relevant property for this autocorrelation-dominated series. This shows that GRPO, through reward-driven exploration, learns more accurate reasoning pathways that better connect series characteristics with appropriate tool capabilities.

Table 15: Generalization study across tool groups (MAE ↓). The model is trained on small model candidates (CARD, LSINet, PDF, TimeXer). (a) Evaluated on **foundation model** candidates (all 4 unseen). (b) Evaluated on **mixed model** candidates (2–3 unseen per dataset). **Bold** in mapping tables indicates unseen models not present during training. Dataset abbreviations: Exch=Exchange, H1/H2=ETTh1/ETTh2, M1/M2=ETTm1/ETTm2, Wea=Weather, Elec=Electricity, Traf=Traffic.

(a) OOD to Foundation Models

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.3845	.3428	.3990	.2516	.3140	.0922	.7105	.8640	.4198
Model 2	.3341	.3169	.3283	.2304	.2461	.0786	**.4383**	.1943	.2709
Model 3	.2984	.3117	.3423	.2506	.2731	.0586	.5764	.3245	.3045
Model 4	.2810	.3476	.3588	.2662	.3957	.0431	.5800	.1894	.3077
Traditional Ensemble									
Ens_{avg}	.3025	.3053	.3349	.2305	.2734	.0652	.5310	.3408	.2980
Ens_{rand}	.3073	.3106	.3395	.2343	.2806	.0656	.5409	.3521	.3039
$\text{InvMSE}_{\text{tr}}$	.2950	.3050	.3332	.2298	.2614	.0603	.5089	.2109	.2756
$\text{InvMSE}_{\text{val}}$	.3016	.3051	.3337	.2298	.2615	.0452	.5086	.2087	.2743
RLMC	.2859	.3055	.3368	.2228	.2461	.0730	**.4383**	**.1839**	.2615
OptW_{tr}	.2833	.3036	.3271	.2257	.2441	.0579	.4466	.1980	.2608
OptW_{val}	.3071	.3052	.3312	.2259	.2447	.0466	.4466	.1988	.2633
LLM-based Ensemble									
GPT-5.2	.2744	.2908	.3220	.2204	.2436	.0535	.4840	.2507	.2674
Codex	.2695	.2873	.3212	.2171	.2409	.0560	.4839	.2365	.2641
GPT-5.5	.2815	.2976	.3287	.2255	.2566	.0585	.4985	.2712	.2773
DeepSeek-V3.2	.2667	.2863	.3193	.2165	.2336	.0539	.4739	.2290	.2599
Grok-4	.2827	.2955	.3280	.2229	.2565	.0637	.5066	.2899	.2807
REATS (Ours)									
REATS-GRPO	**.2482**	**.2760**	**.3103**	**.2139**	**.2280**	**.0382**	.4390	.2711	**.2531**

(b) OOD to Mixed Models

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.2849	**.2958**	1.1093	**.2084**	1.1312	.0608	1.0157	**.1708**	.5346
Model 2	.2810	1.0929	**.3283**	.2516	**.1877**	.1017	.7105	.8640	.4772
Model 3	.3845	1.0635	1.1564	1.1205	1.1374	.0922	**.2810**	1.1882	.8030
Model 4	.9946	.3169	1.1316	1.0863	.3575	.0786	1.0292	1.1745	.7711
Traditional Ensemble									
Ens_{avg}	.3813	.5373	.7387	.5149	.5370	.0775	.6200	.7260	.5166
Ens_{rand}	.4070	.5713	.7783	.5486	.5732	.0797	.6494	.7494	.5446
$\text{InvMSE}_{\text{tr}}$	.2851	.3066	.3910	.2172	.2104	.0774	.3284	.2068	.2529
$\text{InvMSE}_{\text{val}}$	.3716	.3394	.4822	.2424	.2088	.0756	.3560	.2019	.2847
RLMC	.3846	.3161	.3285	.2467	.1893	.0741	.2869	**.1708**	.2446
OptW_{tr}	.2784	.3087	.3494	.2193	.2074	.0803	.2911	.2150	.2437
OptW_{val}	.3070	.3186	.3506	.2514	.2073	.0726	.2937	.2146	.2520
LLM-based Ensemble									
GPT-5.2	.3038	.4679	.6045	.4102	.4087	.0654	.4573	.3766	.3868
Codex	.2931	.4431	.5938	.3572	.3796	.0637	.4308	.3502	.3639
GPT-5.5	.3263	.4952	.6270	.4163	.4429	.0702	.4732	.4430	.4117
DeepSeek-V3.2	.2952	.4528	.5989	.3299	.3961	.0542	.4101	.3426	.3600
Grok-4	.3101	.4725	.6149	.3899	.4458	.0679	.4998	.4525	.4067
REATS (Ours)									
REATS-GRPO	**.2524**	.3095	.3844	.2085	.1948	**.0394**	.2866	.2074	**.2354**

Model Mapping for (a)

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	**MOMENT**	**TimeMoE**	**TimesFM**	**MOIRAI**
ETTh1	**MOMENT**	**Sundial**	**TimesFM**	**MOIRAI**
ETTh2	**MOMENT**	**Sundial**	**TimesFM**	**MOIRAI**
ETTm1	**MOMENT**	**Sundial**	**TimesFM**	**MOIRAI**
ETTm2	**MOMENT**	**Timer**	**TimeMoE**	**MOIRAI**
Weather	**MOMENT**	**Timer**	**TimeMoE**	**TimesFM**
Electricity	**MOMENT**	**Sundial**	**Timer**	**MOIRAI**
Traffic	**MOMENT**	**Sundial**	**Timer**	**TimeMoE**

Model Mapping for (b)

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	CARD	**MOIRAI**	**MOMENT**	**ModernTCN**
ETTh1	PDF	**TimeMixer**	TimeXer	**Timer**
ETTh2	**PatchTST**	**Sundial**	**TimeMixer**	TimeXer
ETTm1	LSINet	**MOMENT**	**PatchTST**	TimeXer
ETTm2	**ModernTCN**	PDF	**PatchTST**	**TimesFM**
Weather	CARD	**MLF**	**MOMENT**	**Timer**
Electricity	**DLinear**	**MOMENT**	PDF	**TimeMixer**
Traffic	CARD	**MOMENT**	**SEMixer**	**TimeMixer**

Table 16: Generalization study within tool groups (MSE ↓). (a) Trained on **foundation model** candidates (MOIRAI, MOMENT, TimeMoE, TimesFM), 1–2 unseen per dataset. (b) Trained on **small model** candidates (CARD, LSINet, PDF, TimeXer), 1–3 unseen per dataset. **Bold** in mapping tables indicates unseen models not present during training. The MAE results are shown in Table 17.

(a) OOD within Foundation Models

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.3037	.1966	.2765	.1124	.1472	.5683	.7635	.9884	.4196
Model 2	.2355	.1742	.2022	.0987	.1229	.4397	.3740	.1800	.2284
Model 3	.2502	.1712	.2102	.1112	.2862	.0752	.5569	.0831	.2180
Model 4	.1675	.2161	.2413	.1273	.3164	**.0039**	.5908	.1489	.2265
Traditional Ensemble									
Ens_{avg}	.2123	.1618	.2070	.0946	.1585	.1670	.4765	.1835	.2076
Ens_{rand}	.2177	.1674	.2119	.0981	.1707	.1837	.4957	.2175	.2203
$\text{InvMSE}_{\text{tr}}$	.1989	.1615	.2052	.0943	.1339	.1083	.4485	.0996	.1813
$\text{InvMSE}_{\text{val}}$	.2129	.1615	.2057	.0943	.1337	.0072	.4481	.0979	.1702
RLMC	.1898	.1619	.2011	.0902	.1209	.2903	.3740	.0827	.1888
OptW_{tr}	.1707	.1608	.2006	.0930	.1183	.0761	.3797	.0864	.1607
OptW_{val}	.1989	.1619	.2007	.0927	.1191	.0109	.3797	.0863	.1563
LLM-based Ensemble									
GPT-5.2	.1760	.1493	.1924	.0884	.1281	.0473	.4160	.1114	.1636
Codex	.1696	.1451	.1928	.0861	.1215	.1146	.4168	.1052	.1689
GPT-5.5	.1940	.1553	.2003	.0912	.1390	.0435	.4351	.1369	.1744
DeepSeek-V3.2	.1690	.1464	.1904	.0861	.1173	.0538	.4033	.1051	.1589
Grok-4	.1929	.1531	.2005	.0902	.1351	.1915	.4443	.1465	.1943
REATS (Ours)									
REATS-GRPO	**.1447**	**.1341**	**.1803**	**.0827**	**.1059**	.0030	**.3648**	**.0810**	**.1371**

(b) OOD within Small Models

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.1690	.1647	2.0318	.0815	.1147	.4372	1.7633	**.0667**	.6036
Model 2	1.5364	**.1558**	**.1825**	1.9358	1.9402	1.2352	1.7247	2.1738	1.3605
Model 3	.5319	2.0278	1.8930	**.0748**	1.9540	.1663	.1472	2.1670	1.1202
Model 4	.5978	1.9515	1.9351	2.0260	**.0791**	.4429	1.7951	2.1059	1.3667
Traditional Ensemble									
Ens_{avg}	.3268	.4752	.8083	.4193	.4189	.1445	.7556	.8227	.5214
Ens_{rand}	.4027	.5965	.9526	.5429	.5359	.2355	.8761	.9823	.6406
$\text{InvMSE}_{\text{tr}}$	.1903	.1608	.2229	.0763	.0843	.1697	.1818	.0742	.1450
$\text{InvMSE}_{\text{val}}$	.2975	.1907	.8023	.1208	.0842	.1857	.2085	.0728	.2453
RLMC	.2547	.1653	.8372	.0814	.1129	.1713	.1473	.0667	.2253
OptW_{tr}	**.1586**	.1634	.1921	.0832	.0836	.1727	.1578	.0851	.1371
OptW_{val}	.4002	.1672	.8066	.0857	.0840	.1857	.1585	.0850	.2466
LLM-based Ensemble									
GPT-5.2	.2719	.2756	.5583	.2252	.3101	.0651	.4118	.2476	.2957
Codex	.2625	.2736	.4768	.1837	.2936	.0737	.3771	.2212	.2703
GPT-5.5	.2771	.3144	.6262	.2890	.3151	.0787	.4311	.2994	.3289
DeepSeek-V3.2	.2659	.4610	.5092	.2402	.5704	.1260	.4059	.2403	.3524
Grok-4	.2763	.3221	.5934	.2237	.3041	.1214	.4528	.3253	.3274
REATS (Ours)									
REATS-GRPO	.1606	**.1524**	.2324	.0755	.0981	**.0446**	.1555	.0812	**.1250**

Model Mapping for (a)

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	MOMENT	**Sundial**	**Timer**	MOIRAI
ETTh1	MOMENT	**Sundial**	TimesFM	MOIRAI
ETTh2	MOMENT	**Sundial**	**Timer**	MOIRAI
ETTm1	MOMENT	**Sundial**	TimesFM	MOIRAI
ETTm2	**Sundial**	**Timer**	TimesFM	MOIRAI
Weather	MOMENT	**Timer**	TimeMoE	TimesFM
Electricity	MOMENT	**Sundial**	**Timer**	MOIRAI
Traffic	MOMENT	**Timer**	TimeMoE	MOIRAI

Model Mapping for (b)

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	CARD	**DLinear**	LSINet	**SEMixer**
ETTh1	CARD	LSINet	**MLF**	**TimeMixer**
ETTh2	**ModernTCN**	PDF	**PatchTST**	TimeXer
ETTm1	LSINet	**MLF**	PDF	**PatchTST**
ETTm2	CARD	**ModernTCN**	**PatchTST**	**SEMixer**
Weather	CARD	LSINet	**PatchTST**	TimeXer
Electricity	**MLF**	**ModernTCN**	PDF	**PatchTST**
Traffic	CARD	PDF	**SEMixer**	**TimeMixer**

Table 17: Generalization study within tool groups (MAE ↓). (a) Trained on **foundation model** candidates (MOIRAI, MOMENT, TimeMoE, TimesFM), 1–2 unseen per dataset. (b) Trained on **small model** candidates (CARD, LSINet, PDF, TimeXer), 1–3 unseen per dataset. **Bold** in mapping tables indicates unseen models not present during training. Dataset abbreviations: Exch=Exchange, H1/H2=ETTh1/ETTh2, M1/M2=ETTm1/ETTm2, Wea=Weather, Elec=Electricity, Traf=Traffic.

(a) OOD within Foundation Models

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.3845	.3428	.3990	.2516	.2756	.0922	.7105	.8640	.4150
Model 2	.3351	.3169	.3283	.2304	.2461	.0786	**.4383**	.3245	.2873
Model 3	.3433	.3117	.3392	.2506	.3575	.0586	.5764	**.1894**	.3033
Model 4	.2810	.3476	.3588	.2662	.3957	.0431	.5800	.2695	.3177
Traditional Ensemble									
Ens_{avg}	.3162	.3053	.3349	.2305	.2843	.0652	.5310	.3530	.3026
Ens_{rand}	.3204	.3106	.3393	.2343	.2915	.0656	.5409	.3657	.3085
$\text{InvMSE}_{\text{tr}}$	.3057	.3050	.3331	.2298	.2621	.0603	.5089	.2294	.2793
$\text{InvMSE}_{\text{val}}$	.3167	.3051	.3337	.2298	.2620	.0452	.5086	.2263	.2784
RLMC	.3005	.3055	.3281	.2228	.2445	.0730	**.4383**	.1895	.2628
OptW_{tr}	.2836	.3036	.3274	.2257	.2444	.0579	.4466	.2045	.2617
OptW_{val}	.3060	.3052	.3279	.2259	.2459	.0466	.4466	.2039	.2635
LLM-based Ensemble									
GPT-5.2	.2836	.2916	.3207	.2210	.2540	.0503	.4846	.2518	.2697
Codex	.2763	.2871	.3208	.2170	.2463	.0579	.4848	.2445	.2668
GPT-5.5	.2982	.2979	.3282	.2252	.2643	.0497	.4986	.2859	.2810
DeepSeek-V3.2	.2734	.2870	.3181	.2161	.2418	.0491	.4733	.2362	.2619
Grok-4	.2972	.2956	.3287	.2235	.2616	.0653	.5066	.3043	.2853
REATS (Ours)									
REATS-GRPO	**.2529**	**.2742**	**.3094**	**.2109**	**.2292**	**.0365**	.4415	.1940	**.2436**

Model Mapping for (a)

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	MOMENT	**Sundial**	**Timer**	MOIRAI
ETTh1	MOMENT	**Sundial**	TimesFM	MOIRAI
ETTh2	MOMENT	**Sundial**	**Timer**	MOIRAI
ETTm1	MOMENT	**Sundial**	TimesFM	MOIRAI
ETTm2	**Sundial**	**Timer**	TimesFM	MOIRAI
Weather	MOMENT	**Timer**	TimeMoE	TimesFM
Electricity	MOMENT	**Sundial**	**Timer**	MOIRAI
Traffic	MOMENT	**Timer**	TimeMoE	MOIRAI

(b) OOD within Small Models

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Model 1	.2849	.3087	1.1625	.2084	.2535	.0608	1.0294	**.1708**	.4349
Model 2	.9776	.2992	**.3183**	1.1029	1.1312	.1165	1.0166	1.1898	.7690
Model 3	.4962	1.1200	1.1093	**.2022**	1.1374	.0615	**.2810**	1.1882	.6995
Model 4	.5167	1.0929	1.1316	1.1205	**.1929**	.0827	1.0367	1.1745	.7936
Traditional Ensemble									
Ens_{avg}	.4416	.5454	.7421	.5210	.5272	.0742	.6723	.7608	.5356
Ens_{rand}	.4714	.5817	.7822	.5522	.5589	.0764	.7085	.7948	.5658
$\text{InvMSE}_{\text{tr}}$	.3167	.3059	.3744	.2045	.2079	.0764	.3173	.1952	.2498
$\text{InvMSE}_{\text{val}}$	.4134	.3378	.7392	.2742	.2077	.0773	.3426	.1915	.3230
RLMC	.3750	.3087	.7556	.2083	.2512	.0759	.2812	.1708	.2960
OptW_{tr}	**.2831**	.3089	.3395	.2198	.2110	.0759	.2925	.2189	.2437
OptW_{val}	.4428	.3132	.7408	.2228	.2118	.0772	.2931	.2188	.3151
LLM-based Ensemble									
GPT-5.2	.3987	.4037	.6049	.3758	.4470	.0549	.4900	.4051	.3975
Codex	.3884	.4035	.5614	.3396	.4304	.0548	.4664	.3817	.3783
GPT-5.5	.4070	.4350	.6479	.4200	.4503	.0588	.4986	.4449	.4203
DeepSeek-V3.2	.3879	.5183	.5636	.3739	.5916	.0626	.4660	.3716	.4169
Grok-4	.4042	.4438	.6304	.3741	.4388	.0645	.5119	.4652	.4166
REATS (Ours)									
REATS-GRPO	.2847	**.2952**	.3669	.2025	.2167	**.0453**	.2900	.2111	**.2391**

Model Mapping for (b)

Dataset	Model 1	Model 2	Model 3	Model 4
Exchange	CARD	**DLinear**	LSINet	**SEMixer**
ETTh1	CARD	LSINet	**MLF**	**TimeMixer**
ETTh2	**ModernTCN**	PDF	**PatchTST**	TimeXer
ETTm1	LSINet	**MLF**	PDF	**PatchTST**
ETTm2	CARD	**ModernTCN**	**PatchTST**	**SEMixer**
Weather	CARD	LSINet	**PatchTST**	TimeXer
Electricity	**MLF**	**ModernTCN**	PDF	**PatchTST**
Traffic	CARD	PDF	**SEMixer**	**TimeMixer**

Table 18: Scalability analysis across different numbers of candidate models (MSE ↓). All candidates are foundation models. (a) N=6, (b) N=8, (c) N=2. Dataset abbreviations: Exch=Exchange, H1/H2=ETTh1/ETTh2, M1/M2=ETTm1/ETTm2, Wea=Weather, Elec=Electricity, Traf=Traffic.

(a) N=6 (Chronos, MOIRAI, MOMENT, Sundial, Timer, TimesFM)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Chronos	.1772	.1796	.2299	.1053	.1925	.0162	.6863	.2638	.2314
MOIRAI	.1675	.2161	.2413	.1273	.3164	.0269	.5908	.1489	.2294
MOMENT	.3037	.1966	.2765	.1124	.1707	.5683	.7635	.9884	.4225
Sundial	.2355	.1742	.2022	.0987	.1472	.1833	**.3740**	.0893	.1881
Timer	.2502	.1746	.2102	.1039	.1229	.4397	.5569	.1800	.2548
TimesFM	.1845	.1712	.2212	.1112	.2862	**.0039**	.4770	.0864	.1927
Traditional Ensemble									
Ens_{avg}	.1919	.1595	.2027	.0920	.1473	.0972	.4765	.1501	.1896
Ens_{rand}	.1957	.1630	.2066	.0947	.1558	.1104	.4905	.1703	.1984
$\text{InvMSE}_{\text{val}}$	.1926	.1594	.2020	.0919	.1326	.0074	.4539	.0987	.1673
RLMC	.1710	.1598	.2032	.0908	.1155	.0247	.4609	**.0860**	.1640
$\text{InvMSE}_{\text{tr}}$	.1829	.1594	.2017	.0918	.1326	.0708	.4541	.0997	.1741
OptW_{tr}	.1649	.1584	.1971	.0909	.1162	.0777	.3828	.0886	.1596
OptW_{val}	.1939	.1595	.2025	.0908	.1176	.0164	.3823	.0884	.1564
LLM-based Ensemble									
GPT-5.2	.1643	.1468	.1905	.0867	.1340	.0288	.4231	.1059	.1600
Codex	.1654	.1468	.1908	.0858	.1260	.0579	.4343	.1151	.1653
GPT-5.5	.1676	.1489	.1922	.0876	.1345	.0350	.4246	.1067	.1621
DeepSeek-V3.2	.1660	.1465	.1900	.0850	.1283	.0377	.4285	.1160	.1622
Grok-4	.1665	.1483	.1919	.0864	.1279	.0562	.4368	.1163	.1663
REATS (Ours)									
REATS-SFT	**.1468**	**.1346**	**.1751**	**.0809**	**.1077**	.0065	.3769	.0862	**.1393**

(b) N=8 (Chronos, MOIRAI, MOMENT, Sundial, TimeMoE, Timer, TimerXL, TimesFM)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Chronos	.1772	.1796	.2299	.1053	.1925	.0162	.6863	.2638	.2314
MOIRAI	.1675	.2161	.2413	.1273	.3164	.0269	.5908	.1489	.2294
MOMENT	.3037	.1966	.2765	.1124	.1707	.5683	.7635	.9884	.4225
Sundial	.2355	.1742	.2022	.0987	.1472	.1833	.3740	.0893	.1881
TimeMoE	.2359	.1768	.1996	.0997	.1416	.0752	.3902	.0831	.1753
Timer	.2502	.1746	.2102	.1039	.1229	.4397	.5569	.1800	.2548
TimerXL	.2744	.1782	.2138	.1028	.1121	.4343	.4104	.1305	.2321
TimesFM	.1845	.1712	.2212	.1112	.2862	**.0039**	.4770	.0864	.1927
Traditional Ensemble									
Ens_{avg}	.2040	.1607	.1995	.0919	.1324	.1205	.4373	.1257	.1840
Ens_{rand}	.2067	.1629	.2023	.0936	.1384	.1306	.4475	.1389	.1901
$\text{InvMSE}_{\text{val}}$	.2046	.1611	.1990	.0921	.1213	.0082	.4181	.0916	.1620
RLMC	.1686	.1612	.1997	.0922	.1079	.0257	.3727	**.0778**	.1507
$\text{InvMSE}_{\text{tr}}$	.1940	.1610	.1989	.0921	.1214	.0921	.4184	.0926	.1713
OptW_{tr}	.1843	.1590	.1954	.0915	.1075	.0716	.3810	.0838	.1593
OptW_{val}	.2033	.1603	.1976	.0915	.1082	.0200	.3807	.0836	.1556
LLM-based Ensemble									
GPT-5.2	.1830	.1512	.1892	.0874	.1183	.0960	.3977	.0957	.1648
Codex	.1844	.1514	.1901	.0875	.1176	.0922	.4067	.1036	.1667
GPT-5.5	.1859	.1541	.1924	.0893	.1220	.1011	.4045	.1001	.1687
DeepSeek-V3.2	.1785	.1487	.1864	.0864	.1142	.0874	.3954	.0996	.1621
Grok-4	.1814	.1515	.1908	.0869	.1183	.0592	.4046	.1008	.1617
REATS (Ours)									
REATS-SFT	**.1622**	**.1390**	**.1767**	**.0832**	**.1041**	.0139	**.3706**	.0832	**.1416**

(c) N=2 (MOMENT, TimerXL)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
MOMENT	.3037	.1966	.2765	.1124	.1707	.5683	.7635	.9884	.4225
TimerXL	.2744	.1782	.2138	.1028	.1121	**.4343**	.4104	.1305	.2321
Traditional Ensemble									
Ens_{avg}	.2870	.1827	.2226	.1062	.1223	.4809	.4887	.3302	.2776
Ens_{rand}	.2877	.1842	.2301	.1066	.1286	.4883	.5215	.4061	.2941
$\text{InvMSE}_{\text{val}}$	.2866	.1823	.2203	.1060	.1171	.4735	.4454	.1357	.2459
RLMC	.2745	.1791	.2114	.1028	.1116	**.4343**	.4104	.1305	.2318
$\text{InvMSE}_{\text{tr}}$	.2863	.1823	.2192	.1060	.1173	.4766	.4463	.1363	.2463
OptW_{tr}	.2749	.1782	.2117	.1029	.1111	.4393	.4094	.1298	.2322
OptW_{val}	.2752	.1783	.2125	.1029	.1111	.4363	.4094	.1298	.2319
LLM-based Ensemble									
GPT-5.2	.2780	.1772	.2068	.1038	.1109	.4431	.4193	.1558	.2369
Codex	.2807	.1791	.2120	.1046	.1125	.4585	.4287	.1568	.2416
GPT-5.5	.2812	.1795	.2155	.1049	.1155	.4487	.4452	.2125	.2504
DeepSeek-V3.2	.2786	.1776	.2072	.1039	.1097	.4467	.4128	.1399	.2346
Grok-4	.2796	.1788	.2119	.1046	.1142	.4402	.4326	.1801	.2428
REATS (Ours)									
REATS-SFT	**.2735**	**.1746**	**.2018**	**.1027**	**.1091**	.4356	**.3990**	**.1275**	**.2280**

Table 19: Scalability study with foundation model candidates (MAE ↓). We evaluate REATS with varying numbers of candidate models: (a) N=6, (b) N=8, and (c) N=2. Dataset abbreviations: Exch=Exchange, H1/H2=ETTh1/ETTh2, M1/M2=ETTm1/ETTm2, Wea=Weather, Elec=Electricity, Traf=Traffic.

(a) N=6 (Chronos, MOIRAI, MOMENT, Sundial, Timer, TimesFM)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Chronos	.2986	.3165	.3541	.2391	.3095	.0413	.6375	.3693	.3207
MOIRAI	.2810	.3476	.3588	.2662	.3957	.0584	.5800	.2695	.3196
MOMENT	.3845	.3428	.3990	.2516	.3140	.0922	.7105	.8640	.4198
Sundial	.3351	.3169	.3283	.2304	.2756	.0682	**.4383**	.1943	.2734
Timer	.3433	.3169	.3392	.2378	.2461	.0786	.5764	.3245	.3078
TimesFM	.2984	.3117	.3423	.2506	.3575	.0431	.5040	**.1932**	.2876
Traditional Ensemble									
Ens_{avg}	.3022	.3021	.3303	.2260	.2776	.0597	.5269	.3087	.2917
Ens_{rand}	.3052	.3056	.3339	.2292	.2838	.0601	.5345	.3185	.2963
$\text{InvMSE}_{\text{val}}$	.3026	.3020	.3295	.2256	.2636	.0435	.5092	.2209	.2746
RLMC	.2893	.3026	.3308	.2236	.2405	.0503	.5057	.1950	.2672
$\text{InvMSE}_{\text{tr}}$	.2956	.3019	.3292	.2255	.2637	.0563	.5093	.2227	.2755
OptW_{tr}	.2818	.3004	.3247	.2227	.2424	.0571	.4498	.2032	.2603
OptW_{val}	.3050	.3022	.3292	.2228	.2444	.0440	.4491	.2029	.2625
LLM-based Ensemble									
GPT-5.2	.2757	.2879	.3182	.2182	.2610	.0480	.4875	.2408	.2672
Codex	.2762	.2879	.3188	.2166	.2538	.0519	.4966	.2571	.2699
GPT-5.5	.2780	.2905	.3198	.2198	.2622	.0494	.4879	.2424	.2687
DeepSeek-V3.2	.2764	.2872	.3185	.2154	.2548	.0479	.4905	.2507	.2677
Grok-4	.2780	.2897	.3199	.2175	.2560	.0515	.4981	.2573	.2710
REATS (Ours)									
REATS-SFT	**.2572**	**.2739**	**.3041**	**.2088**	**.2331**	**.0373**	.4491	.2013	**.2456**

(b) N=8 (Chronos, MOIRAI, MOMENT, Sundial, TimeMoE, Timer, TimerXL, TimesFM)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
Chronos	.2986	.3165	.3541	.2391	.3095	**.0413**	.6375	.3693	.3207
MOIRAI	.2810	.3476	.3588	.2662	.3957	.0584	.5800	.2695	.3196
MOMENT	.3845	.3428	.3990	.2516	.3140	.0922	.7105	.8640	.4198
Sundial	.3351	.3169	.3283	.2304	.2756	.0682	.4383	.1943	.2734
TimeMoE	.3341	.3199	.3238	.2311	.2731	.0586	.4489	.1894	.2724
Timer	.3433	.3169	.3392	.2378	.2461	.0786	.5764	.3245	.3078
TimerXL	.3600	.3199	.3399	.2364	.2354	.0822	.4512	.2408	.2832
TimesFM	.2984	.3117	.3423	.2506	.3575	.0431	.5040	.1932	.2876
Traditional Ensemble									
Ens_{avg}	.3106	.3033	.3272	.2249	.2632	.0620	.4959	.2738	.2826
Ens_{rand}	.3127	.3055	.3298	.2269	.2679	.0623	.5018	.2813	.2860
$\text{InvMSE}_{\text{val}}$	.3110	.3037	.3266	.2248	.2510	.0441	.4790	.2073	.2684
RLMC	.2852	.3038	.3271	.2251	.2314	.0539	**.4381**	**.1802**	.2556
$\text{InvMSE}_{\text{tr}}$	.3032	.3035	.3264	.2247	.2511	.0589	.4793	.2090	.2695
OptW_{tr}	.2954	.3010	.3227	.2228	.2330	.0562	.4477	.1972	.2595
OptW_{val}	.3105	.3028	.3239	.2228	.2339	.0453	.4472	.1970	.2604
LLM-based Ensemble									
GPT-5.2	.2913	.2929	.3170	.2183	.2462	.0578	.4633	.2211	.2635
Codex	.2920	.2927	.3181	.2183	.2457	.0568	.4721	.2379	.2667
GPT-5.5	.2938	.2960	.3200	.2211	.2504	.0592	.4692	.2280	.2672
DeepSeek-V3.2	.2872	.2896	.3146	.2162	.2399	.0551	.4615	.2273	.2614
Grok-4	.2899	.2928	.3185	.2175	.2464	.0532	.4692	.2307	.2648
REATS (Ours)									
REATS-SFT	**.2734**	**.2788**	**.3052**	**.2110**	**.2281**	.0414	.4435	.1979	**.2474**

(c) N=2 (MOMENT, TimerXL)

Method	Exch	H1	H2	M1	M2	Wea	Elec	Traf	Avg
Individual Models									
MOMENT	.3845	.3428	.3990	.2516	.3140	.0922	.7105	.8640	.4198
TimerXL	.3600	.3199	.3399	.2364	.2354	.0822	**.4512**	.2408	.2832
Traditional Ensemble									
Ens_{avg}	.3710	.3270	.3517	.2421	.2552	.0867	.5356	.4873	.3321
Ens_{rand}	.3714	.3284	.3576	.2427	.2617	.0869	.5504	.5056	.3381
$\text{InvMSE}_{\text{val}}$	.3706	.3265	.3493	.2418	.2471	.0862	.4970	.2649	.2979
RLMC	.3601	.3219	.3393	.2364	.2351	.0822	.4512	**.2408**	.2834
$\text{InvMSE}_{\text{tr}}$	.3704	.3265	.3481	.2417	.2475	.0864	.4978	.2664	.2981
OptW_{tr}	.3604	.3201	.3393	.2366	.2349	.0829	.4530	.2454	.2841
OptW_{val}	.3606	.3203	.3406	.2366	.2349	.0825	.4527	.2454	.2842
LLM-based Ensemble									
GPT-5.2	.3631	.3207	.3365	.2382	.2383	.0833	.4752	.2999	.2944
Codex	.3657	.3229	.3410	.2396	.2407	.0846	.4836	.3026	.2976
GPT-5.5	.3656	.3234	.3444	.2400	.2453	.0842	.4972	.3598	.3075
DeepSeek-V3.2	.3633	.3211	.3359	.2384	.2368	.0834	.4690	.2706	.2898
Grok-4	.3644	.3222	.3411	.2395	.2434	.0833	.4861	.3310	.3014
REATS (Ours)									
REATS-SFT	**.3590**	**.3179**	**.3319**	**.2363**	**.2349**	**.0820**	.4546	.2482	**.2831**