PERCEIVE, SOLVE, REASON: PROGRESSIVE CAPABILITY FORMATION IN TIME SERIES-LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time Series Large Language Models (TS-LLMs) have shown growing capabilities in perception, task solving, and reasoning, yet these abilities are typically studied independently. We organize TS-LLM capabilities into three stages—Perceive, Solve, and Reason, and investigate how they progressively develop under two representative alignment architectures. Experiments show that Perceive facilitates Solve, while the complete Perceive $\rightarrow$ Solve $\rightarrow$ Reason pathway yields the strongest reasoning performance, especially under limited reasoning supervision. We further find that ordered capability accumulation is more effective than mixed supervision, and that the formation process is affected by input format and stage maturity. Cross-layer representation analysis reveals that different capabilities are transferred through distinct network depths rather than simply accumulated. Moreover, local stage convergence does not necessarily indicate sufficient preparation for subsequent capability development. Based on these findings, we derive a progressive training strategy and build strong Pre-align and Post-align TS-LLMs, achieving the best overall performance under fair out-of-domain evaluation.

1 INTRODUCTION

The rapid advancement of Large Language Models (LLMs) has accelerated the development of intelligent time series analysis (Brown et al., 2020). By representing time series as sequences compatible with language models, Time Series Large Language Models (TS-LLMs) have shown promising capabilities in forecasting, classification, and anomaly detection (Zhou et al., 2023; Jin et al., 2024; Liu et al., 2025). Beyond numerical prediction, TS-LLMs can follow natural language instructions to understand temporal patterns, generate explanations, perform reasoning, and support decision-making (Xie et al., 2025a; Kong et al., 2025; Wang et al., 2025; Guan et al., 2025; Yu et al., 2026). Therefore, empowering language models with stronger temporal understanding and reasoning abilities has emerged as a central research direction (Jiang et al., 2025; Yu et al., 2026).

On the one hand, some studies focus on improving LLMs' perception of fundamental time series patterns. ChatTS enhances LLMs' understanding of basic patterns, such as trends, seasonality, and noise (Xie et al., 2025b). TimeSeriesExam provides a systematic evaluation of LLMs' understanding of fundamental time series concepts (Cai et al., 2024), while CaTS-Bench introduces tasks for evaluating temporal matching (Zhou et al., 2026). However, these efforts remain largely limited to basic time series understanding. On the other hand, some studies focus on solving downstream tasks. FSCA (Hu et al., 2025), SE-LLM (Liu et al., 2026), and Time-CMA (Liu et al., 2025) improve LLM performance on numerical tasks, including forecasting, imputation, anomaly detection, and classification (Liu et al., 2025). However, they mainly optimize task-specific performance and do not investigate whether LLMs acquire diverse capabilities. Furthermore, recent TSAlign (Qiu et al., 2026), Time-Omni-1 (Guan et al., 2025), and ITFormer (Wang et al., 2025) construct time series question-answering datasets and leverage instruction tuning, chain-of-thought, and reinforcement learning to advance TS-LLMs toward intelligent analysis.

Complex time series reasoning relies on fundamental perception (Chen et al., 2025; Yu et al., 2026). This suggests that TS-LLMs may follow a capability development trajectory similar to human cognition: Perceive, Solve, Reason. However, although existing studies have separately explored these three stages, they rarely investigate how the corresponding capabilities emerge, interact, and progressively develop in TS-LLMs. To address this gap, we study TS-LLM capability formation from

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.

a capability development perspective. We divide TS-LLM capabilities into three stages. The three stages progressively capture fundamental temporal properties, solve standard time series tasks, and support higher-level temporal reasoning.

Across extensive experiments, we find that TS-LLM capabilities exhibit a clear progressive dependency. Learning Perceive before Solve improves downstream task performance, while the complete Perceive $\rightarrow$ Solve $\rightarrow$ Reason pathway yields the strongest high-level reasoning ability under two main alignment architectures. Moreover, progressive training is particularly advantageous when reasoning supervision is limited. These findings suggest that high-level reasoning is better formed through ordered capability accumulation than through mixed supervision. In addition, our study reveals several principles underlying this formation process. Progressive training improves optimization alignment for Reason. Our representation analysis shows that Perceive, Solve, and Reason are transferred through different network depths rather than simply accumulated. We further find that local convergence does not imply capability sufficiency, as additional training can still benefit subsequent capability development.

Finally, we translate these findings into a practical recipe for training a strong TS-LLM from scratch. We organize training data according to the Perceive, Solve, and Reason stages, adopt dedicated alignment architectures, and use stage-wise optimization supervision. Following this data–architecture–strategy design, we train both Pre-align and Post-align TS-LLMs. Under a fair out-of-domain evaluation, our models achieve the best performance, showing that progressive capability formation provides an effective path toward building strong and comprehensive TS-LLMs.

Our contributions are summarized as follows:

- We propose a progressive capability formation framework for TS-LLMs, organizing temporal capabilities into three stages: Perceive, Solve, and Reason.
- We systematically investigate how capability order, stage maturity, input format, optimization dynamics, and representation transfer affect progressive capability development.
- We derive a practical training strategy from these findings and build strong TS-LLMs, which achieve the best overall performance under fair out-of-domain evaluation.

2 Related Work

LLM for Time Series Analysis. Recent studies have explored the potential of LLMs for time series analysis across various tasks, including forecasting, classification, anomaly detection, and QA (Prithyani et al., 2024; Kong et al., 2025; Wang et al., 2025; Xu et al., 2026). These works investigate how the semantic understanding and instruction-following capabilities of LLMs can be adapted to temporal data (Jin et al., 2024; He et al., 2026). For instance, CrisisTS integrates textual crisis reports with meteorological time series to improve multilingual urgency assessment beyond text-only approaches (Meunier et al., 2025). Time-MQA extends LLMs to multi-task temporal QA by combining language reasoning with time series information (Kong et al., 2025), while ITFormer develops an end-to-end framework that jointly models temporal patterns and language generation for multimodal QA in engine maintenance scenarios (Wang et al., 2025). Existing studies mainly target individual downstream tasks, while general temporal capability formation remains underexplored.

QA with Language and Time Series. Some studies have explored the integration of language models with time series data to enable interactive temporal QA (Xie et al., 2025b; Kong et al., 2025; Wang et al., 2025; Guan et al., 2025; Yu et al., 2026). These works construct time series QA datasets and leverage instruction tuning, chain-of-thought reasoning, and alignment strategies to extend TS-LLMs from numerical prediction toward temporal understanding (Xie et al., 2025b; Kong et al., 2025; Wang et al., 2025; Guan et al., 2025). For example, Time-MQA introduces multi-task temporal QA by combining language reasoning with time series information (Kong et al., 2025), while recent frameworks further investigate complex temporal analysis through instruction following and reasoning-oriented training (Wang et al., 2025; Guan et al., 2025; Yu et al., 2026). Existing approaches mainly optimize QA performance, while the temporal perception and task-solving foundations of reasoning remain underexplored. Direct QA supervision may therefore yield strong responses without grounded temporal understanding.

3 A PROGRESSIVE THREE-STAGE CAPABILITY DIMENSIONS FOR TS-LLMS

Despite recent progress in applying LLMs to time series perception, task solving, and reasoning, these capabilities are often studied independently. To analyze the relationships among different tasks, we organize the capabilities of TS-LLMs into three stages: Perceive, Solve, and Reason. More data details are provided in Appendix A& C.

Model Architecture. Existing studies have developed three main paradigms in Figure 1. ***(1) Time Series Textualization*** directly converts time series into textual sequences that can be processed by LLMs. ***(2) Pre-align*** architecture extracts temporal representations with an independent time series encoder and projects them into the LLM embedding space, enabling joint processing with text inputs. ***(3) Post-align*** architecture first encodes raw time series into continuous temporal embeddings and feeds them into the LLM as additional inputs. In our experiments, we adopt TSAlign (Qiu et al., 2026) and Time-MQA (Kong et al., 2025) as representative Pre-align and Post-align architectures, with Qwen3-4B (Yang et al., 2025) as the LLM backbone. In Pre-align, we train only the encoder and alignment module; in Post-align, we train the encoder and LLM backbone; and in the textualization setting, only the LLM is fine-tuned via SFT.

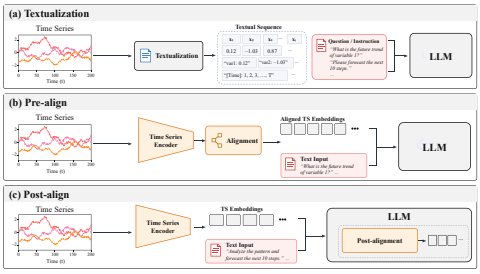


Figure 1: Three Model Architecture.

Stage 1: Fundamental Attribute Perception. Advanced time series analysis relies on understanding fundamental temporal dynamics. For example, forecasting requires capturing trend patterns, while anomaly analysis requires identifying local behaviors. Therefore, Stage 1 aims to equip LLMs with basic temporal perception. The questions focus on trends, seasonality, anomalies, local patterns, and multivariate relationships. ***Training Set.*** The training data is mainly collected from ChatTS, with approximately ∼105K samples (Xie et al., 2025b). ***Evaluation Set.*** We use two datasets to assess perception. Dataset A contains 525 manually constructed questions. Dataset B contains 1,616 synthetic questions. The evaluation includes *categorical* and *numerical* tasks. Categorical tasks are evaluated by *F1-score*, while numerical tasks are evaluated by *relative accuracy*.

Stage 2: Temporal Task Solving. Many real-world time series applications require models to solve specific tasks. Therefore, Stage 2 investigates whether models can transfer basic temporal perception abilities to downstream task solving. ***Training Set.*** We use TSQA (Chen et al., 2025) and TS Instruct (Quinlan et al., 2026) to train task-solving abilities. TSQA contains 155,214 samples, including 42,557 forecasting, 38,657 imputation, 37,000 anomaly detection, and 37,000 classification samples. TS Instruct provides an additional 2,945 classification samples, resulting in 158,159 samples in total. All samples are split into training and testing sets with an 8:2 ratio. ***Evaluation Set.*** We evaluate four standard tasks. Forecasting and imputation are evaluated using *Mean Squared Error (MSE)*, while anomaly detection and classification are evaluated using *Accuracy*.

Stage 3: High-level Reasoning. Stage 3 aims to advance TS-LLMs to intelligent systems capable of analysis. ***Training Set.*** TSQA reasoning QA provides 26,163 samples, ChatTS TSEvol and Instruct Follow (Xie et al., 2025b) provide 29,320 samples, the TSCognition (Qiu et al., 2026) training set provides 32,869 samples, and TS Instruct provides 12,852 samples, resulting in 105,751 samples in total. ***Evaluation Set.*** We use two independent sets. Set A contains 4,547 samples covering binary, multiple-choice, and open-ended questions, evaluated using *Accuracy*, *ROUGE-L*, and *BLEU*. Set B contains 6,163 samples and evaluates five dimensions: *Decoding*, *Grounding*, *Inferring*, *Extrapolating*, and *Acting*. The metric is the accuracy of multiple-choice questions.

4 INVESTIGATING CAPABILITY FORMATION IN TS-LLMS

We formulate five research questions (RQs) to investigate the mechanisms underlying capability relationships in TS-LLMs. They examine cross-stage transfer, progressive supervision, input format, stage-wise convergence, and the conditions that support high-level capability formation. The LLM backbone is primarily Qwen3-4B, with additional Qwen3-1.7B results reported in Appendix E.

RQ1: Can Temporal Perception Facilitate to Task Solving? We first evaluate the perception ability of models after Stage 1 training. As shown in Figure 2& Table 16, we compare two TS-LLM architectures, Pre-align and Post-align, with general multimodal LLMs, GPT-4o (Text) and GPT-4o

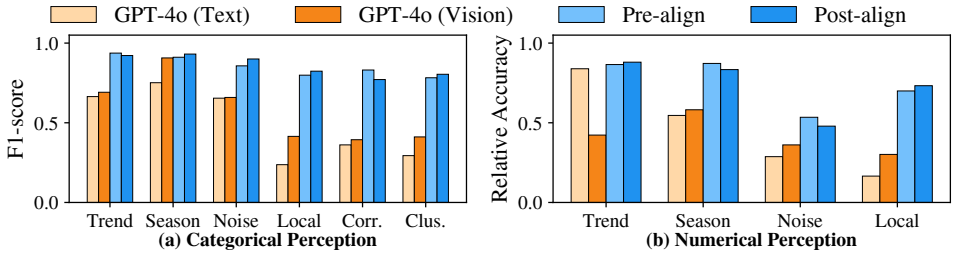


Figure 2: Categorical and numerical perception comparison across different methods.

Model	Training Setting	Fore. MSE ↓	Impu. MSE ↓	Ano Det. Acc. ↑	Class. Acc. ↑
Qwen3-4B	No fine-tuning	1.970	0.0286	0.333	0.296
Pre-align	Stage 1 only	1.842	0.0194	0.587	0.459
	Stage 2 only	1.368	0.0136	0.682	0.567
	Stage (1 → 2)	**1.328**	**0.0132**	**0.739**	**0.628**
Post-align	Stage 1 only	1.716	0.0217	0.612	0.431
	Stage 2 only	1.423	**0.0146**	0.660	0.529
	Stage (1 → 2)	**1.286**	0.0159	**0.720**	**0.597**

Table 1: Results of four task solving evaluation sets.

(Vision), where the GPT-4o results are adopted from (Xie et al., 2025a). Both Pre-align and Post-align TS-LLMs outperform GPT-4o on most tasks, showing that single Stage 1 training enables TS-LLMs to develop effective fundamental perception. We compare different learning paradigm on the Stage 2 task solving evaluation set to investigate whether perception abilities can facilitate task solving. As shown in Table 1, ***Stage 1 only*** denotes training only on the training set of Temporal Perception. ***Stage 2 only*** directly trains the model on the training set of Task Soving, and ***Stage (1 → 2)*** denotes sequential training first on Temporal Perception and then on Task Solving.

Conclusion: *(1) Stage 1 provides effective fundamental perception.* Compared with the non-fine-tuned model, Stage 1 improves all tasks, indicating effective acquisition of fundamental temporal attributes. *(2) Stage 2 is the primary source of performance improvement.* Training on task-solving objectives enhances the corresponding downstream capabilities. *(3) Temporal perception can effectively facilitate downstream task solving.* Stage (1 → 2) obtains the best results on most tasks.

RQ2: Do Perceive and Solve Facilitate Reason Formation? Recent studies have advanced TS-LLMs toward complex QA reasoning (Wang et al., 2025; Guan et al., 2025). However, QA supervision mainly optimizes final answers and may overlook critical temporal evidence. We therefore compare four capability development paths across five dimensions. Experimental results in Table 2 and 3 show that introducing preceding stages improves reasoning performance compared with directly training on Reasoning datasets. On Evaluation Set A, ***Stage (1 → 3)*** achieves higher accuracy than ***Stage 3 Only***, indicating that perception helps models better utilize evidence. ***Stage (2 → 3)*** also improves multiple reasoning dimensions, suggesting that task solving experience provides richer temporal dynamics for subsequent reasoning. Similar trends are observed on Evaluation Set B, where both preceding-stage strategies improve True/False, multiple-choice, and open-ended QA performance in most cases. Overall, ***Stage (1 → 2 → 3)*** achieves the best results on most metrics.

Conclusion: *(1) Perceive and Solve jointly facilitate the formation of Reason.* Temporal perception provides fundamental knowledge, while task-solving experience exposes models to richer dynamics, together supporting more effective high-level reasoning. *(2) The complete Stage (1 → 2 → 3) pathway achieves the best overall performance.*

RQ3: Does Reason Benefit from Progressive Capability Accumulation? RQ2 has shown that both Perceive and Solve provide effective foundations for Reason. We further investigate whether these capabilities need to be accumulated progressively following the stage order. To this end, we compare three training strategies: ***Stage 3***

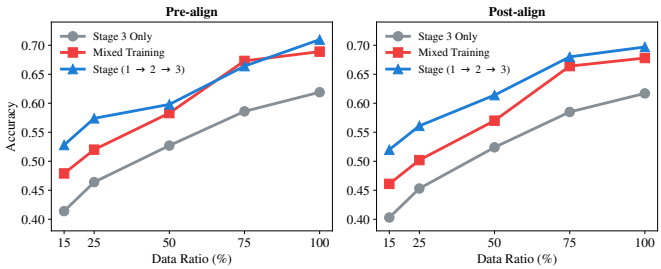


Figure 3: Accuracy across different reasoning training data scales.

Model	Training Setting	Decoding	Grounding	Inferring	Extrapolating	Acting
Pre-align	Stage 3 Only	72.2	68.5	61.3	57.2	50.7
	Stage (1 → 3)	77.8	**81.5**	63.3	62.2	51.8
	Stage (2 → 3)	75.1	72.0	65.8	59.5	53.9
	Stage (1 → 2 → 3)	**82.5**	79.1	**70.7**	**65.4**	**57.2**
Post-align	Stage 3 Only	68.5	66.2	58.9	58.8	49.2
	Stage (1 → 3)	75.6	73.8	61.2	64.1	54.3
	Stage (2 → 3)	73.9	72.7	**66.0**	**68.9**	58.8
	Stage (1 → 2 → 3)	**79.0**	**79.4**	64.2	66.7	**61.4**

Table 2: Accuracy ↑(%) of different strategies on the Evaluation Set A of high-level reasoning.

Model	Training Setting	Ture/False QA Acc. ↑	Multi-Choice QA Acc. ↑	Open-end QA ROUGE-L↑	Open-end QA BLEU↑
Pre-align	Stage 3 Only	71.4	58.6	43.8	24.7
	Stage (1 → 3)	78.3	62.2	45.5	25.8
	Stage (2 → 3)	74.5	65.1	47.2	26.7
	Stage (1 → 2 → 3)	**80.6**	**68.8**	**51.6**	**31.5**
Post-align	Stage 3 Only	66.3	60.3	42.1	22.3
	Stage (1 → 3)	70.4	63.6	40.8	24.5
	Stage (2 → 3)	68.6	67.6	44.4	25.9
	Stage (1 → 2 → 3)	**72.3**	**71.5**	**49.7**	**28.9**

Table 3: Performance on the Evaluation Set B of high-level reasoning.

Only, ***Mixed Training***, and ***Stage (1→ 2 → 3)***. We gradually vary the proportion of high-level reasoning data from 15% to 100%, while keeping the perception and task-solving training data unchanged. Figure 3 & Table 17 shows that progressive capability accumulation achieves stronger Reason performance on the Evaluation Set A of high-level reasoning across most data scales. When only 15% of Stage 3 data is used, ***Stage (1→2→3)*** reaches 0.528 and 0.520 on Pre-align and Post-align, clearly outperforming both ***Mixed Training*** and ***Stage 3 Only***. As more Stage 3 data is introduced, all strategies improve, while the progressive pathway remains highly competitive.

Conclusion: *(1) Ordered capability accumulation is more effective than mixed supervision for Reason formation.* Under the same supervision sources, progressive training consistently outperforms mixed training. *(2) Progressive capability accumulation improves Reason supervision efficiency.* Its advantage becomes more pronounced when Stage 3 data is limited.

RQ4: Does Time Series Input Format Affect Progressive Formation? We adopt the *Textualization* paradigm, which directly feeds time series representations into the LLM, and compare ***Stage 3 Only***, ***Mixed Training***, ***Stage (1 → 3)***, ***Stage (2 → 3)***, and ***Stage (1 → 2 → 3)*** under the same training paths. Reason performance is evaluated on the Evaluation Set A of high-levle reasoning. Results are shown in Figure 4 & Table 17. ***No Training*** denotes direct evaluation of the original Qwen3-4B without any additional training. We observe that under the Textualization paradigm, all training strategies improve steadily as more training data is introduced. However, the gaps between different capability paths are much smaller than those under Pre-align and Post-align. In addition, the complete ***Stage (1 → 2 → 3)*** pathway provides only limited gains and performs similarly to ***Mixed Training***, ***Stage (1 → 3)***, and ***Stage (2 → 3)***. Our results indicate that progressive capability formation is more evident in TS-LLMs with dedicated cross-modal alignment, while the cross-stage accumulation effect is weakened under direct textualized input.

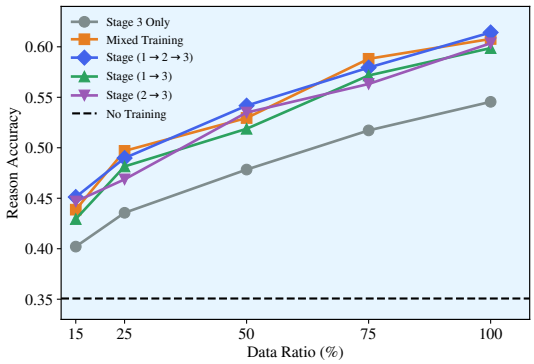


Figure 4: Reason under the Textualization paradigm across training data scales of high-level reasoning.

Conclusion: *(1) Time series input format affects progressive capability formation.* Input strategies with temporal modeling exhibit clearer progressive gains, while direct textualization weakens this effect. *(2) Explicit temporal integration facilitates cross-stage capability transfer.* It better preserves and transfers preceding capabilities across progressive stages.

Training Setting	Training Data	Evaluation Performance	Relative Perf.	I→II→III
Stage 1 only	(40% → 100%) I	0.613, 0.662, **0.738**, 0.745, 0.751	98.3%	0.751
Stage (1 → 2)	70% I, (40% → 100%) II	0.542, 0.603, **0.644**, 0.649, 0.652	98.7%	0.684
Stage (1 → 2 → 3)	70% I, 70% II, (40% → 100%) III	0.537, 0.569, 0.616, **0.663**, 0.682	97.2%	0.710

Table 4: Stage-wise performance under increasing training data scales. For each stage, we progressively increase the amount of training data while fixing the preceding stages at their identified saturation points. **Bold** denote the local saturation points, where the performance has already reached nearly the full-data performance.

RQ5: Does Stage-wise Convergence Ensure TS-LLMs Progression? During progressive learning, a key question is whether convergence at a preceding stage provides a sufficiently stable foundation for subsequent capability development. Similar to how pretraining states affect post-training in LLMs, local convergence does not necessarily imply sufficient preparation for later capability formation (Akter et al., 2026; Chen et al., 2026). We therefore examine two processes: ***(i) when increasing data in a preceding stage causes its stage-specific performance to enter a saturation regime;*** (ii) ***whether additional data from that stage can still provide gains for subsequent capability formation after its local performance has largely stabilized.*** As shown in Table 4 & Table 18, local performance at each stage reaches a clear saturation regime before using the full training set. ***Stage 1 only*** and ***Stage (1→2)*** achieve 98.3% and 98.7% of their full-stage performance at around 70% data, while ***Stage (1→2→3)*** reaches 97.2% at 85%. This indicates that each stage appears nearly converged under its own evaluation. However, local saturation does not imply that the model is fully prepared for subsequent capability development. Full I→II→III training still yields additional gains, especially for Solve and Reason.

Conclusion: *Stage-wise convergence does not necessarily indicate capability sufficiency for subsequent progression.* A capability can approach its local performance ceiling while additional training at that stage still provides valuable foundations for later capability formation, suggesting that local saturation alone is insufficient to determine when progressive training should advance.

5 Mechanisms of Progressive Capability Formation

To understand why progressive learning facilitates capability development, we examine the internal mechanisms underlying the three-stage pathway. We conduct analyses from the perspectives of optimization dynamics (Sec. 5.1), temporal attribute contribution (Sec. 5.2), representation transfer across stages(Sec. 5.4), and alignment-driven capability formation (Sec. 5.3).

5.1 Optimization Dynamics

To understand the optimization mechanism behind progressive capability formation, we analyze the training dynamics from the perspective of data-induced gradient evolution.

Formulation. *We consider 3 capability stages, with data distributions denoted as $\mathcal{D}_P$, $\mathcal{D}_S$, and $\mathcal{D}_R$. All stages share the same objective: $\ell(\theta; x, y) = -\sum_t \log p_\theta(y_t|x, y_{<t})$. For each capability stage $i \in \{P, S, R\}$, the corresponding optimization objective is defined over its data distribution: $\mathcal{L}_i(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[\ell(\theta; x, y)]$. The gradient induced by each capability objective is given by: $g_i(\theta) = \nabla_\theta \mathcal{L}_i(\theta)$. Under Mixed Training, multiple capability objectives are optimized simultaneously (Sener & Koltun, 2018). The overall update direction is determined by the aggregated gradient: $g_{\text{mix}}(\theta) = \sum_{i\in\{P,S,R\}} \alpha_i g_i(\theta)$, where α_i denotes the sampling proportion of each stage. Balancing gradients from different objectives is important for stable multi-task optimization (Chen et al., 2018). Such joint optimization may introduce competition among capability objectives, causing different stages to follow inconsistent optimization trajectories (Yu et al., 2020; Liu et al., 2021). Progressive Training follows the ordered capability pathway: $P \to S \to R$, where each stage is optimized sequentially. This separates the optimization process and provides stage-specific updates during capability formation.*

Definition. ***(1) Optimization Strength.*** *For each capability stage i, the optimization strength is measured by the squared gradient norm: $G_i(\theta) = \|g_i(\theta)\|_2^2$, where $g_i(\theta)$ denotes the gradient induced by stage i data. A larger $G_i(\theta)$ indicates stronger optimization signals from the corresponding capability objective. Under mixed training, the relative gradient contribution is computed as $R_i = G_i(\theta)/\sum_j G_j(\theta)$.* ***(2) Optimization Alignment.*** *The directional compatibil-*

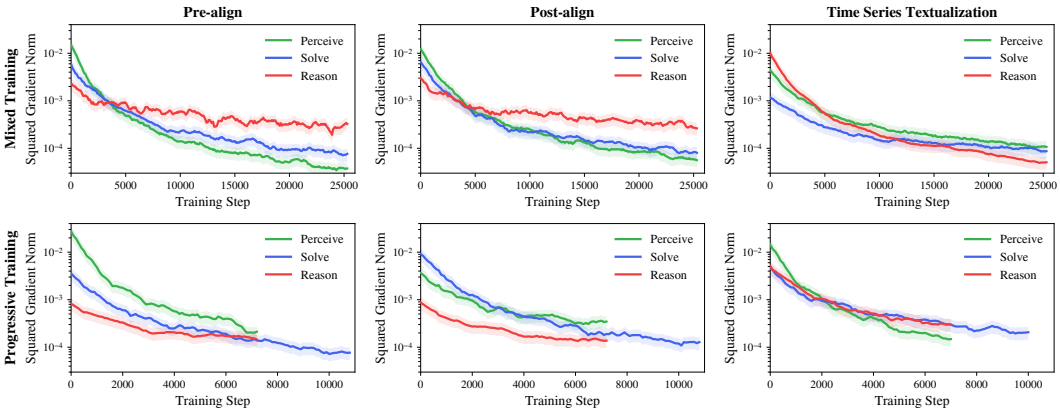


Figure 5: Optimization strength dynamics across capability stages.

Model	Setting	Perceive	Solve	Reason
Pre-align	Mixed	0.463	0.382	0.214
	Progressive	0.612	0.574	0.593
Post-align	Mixed	0.421	0.347	0.236
	Progressive	0.534	0.536	0.565
TS Textualization	Mixed	0.318	0.219	0.241
	Progressive	0.301	0.254	0.226

Table 5: Optimization alignment between specific and overall optimization directions.

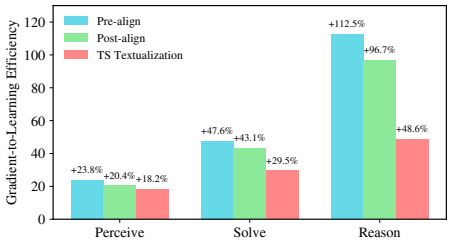


Figure 6: Gradient update efficiency.

ity between capability-specific optimization and the overall training objective is quantified by: $A_i = g_i^T g_{\text{all}} / \|g_i\|_2 \|g_{\text{all}}\|_2$, *where* $g_{\text{all}} = \sum_{i \in \{P,S,R\}} g_i$ *denotes the aggregated gradient across all capability stages. A larger* A_i *indicates that the capability-specific gradient follows the global optimization direction more closely.* ***(3) Gradient Update Efficiency.*** *The efficiency of converting gradients from each capability stage into parameter updates is:* $U_i = \frac{\|\theta_{t+\Delta} - \theta_t\|_2}{\sum_{k=t}^{t+\Delta} \|g_i(k)\|_2}$, *where* $\theta_{t+\Delta} - \theta_t$ *denotes the accumulated parameter update during optimization. A larger* U_i *indicates that gradients from a capability stage are converted into parameter updates more effectively. We further compare the relative update efficiency across capability stages by normalizing* U_i *with respect to the overall optimization efficiency:* $\Delta U_i = \frac{U_i}{\sum_j U_j} \times 100\%$.

Analysis. As shown in Figure 5, Optimization Strength reveals distinct optimization dynamics between Mixed Training and progressive training. We quantify the optimization strength of each capability stage using the squared gradient norm. Mixed Training jointly optimizes multiple capability objectives, leading to imbalanced optimization signals, where Reason supervision may dominate parameter updates. In contrast, progressive training allocates optimization strength according to the capability development order, allowing each stage to receive more concentrated optimization signals. TS Textualization fails to achieve optimization dynamics comparable to architectures with Pre-align and Post-align. Therefore, structured temporal representations provide more stable optimization signals. Building on the analysis of optimization signal magnitude, we compute the alignment score of each capability stage during training. This metric measures the directional consistency between capability-specific gradients and the aggregated gradient, providing a comparison of optimization directions. As shown in Table 5, Progressive Training improves gradient alignment under both Pre-align and Post-align architectures. The improvement is particularly evident in the Reason stage, suggesting that progressive training enables high-level capability optimization. Finally, we compute the relative efficiency gain of Progressive Training over Mixed Training to evaluate how effectively optimization signals are utilized under different temporal representations. As shown in Figure 6, Progressive Training improves optimization efficiency across all temporal representation paradigms. The gain is most significant in the Reason stage, where Pre-align and Post-align achieve efficiency improvements of 112.5% and 96.7%, and TS Textualization achieves only 48.6%.

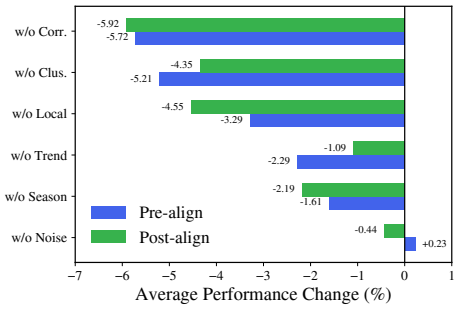


Figure 7: Temporal attribute ablation analysis.

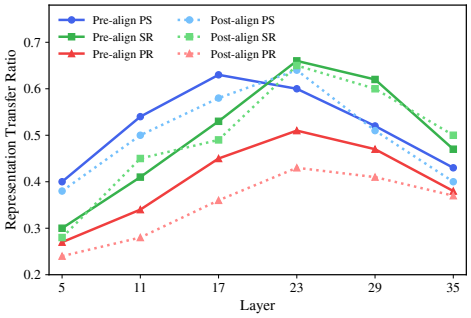


Figure 8: Representation transfer across layers.

5.2 Impact of Temporal Attributes on Reasoning Capability

To analyze the role of perception abilities, we conduct an ablation study over six Stage 1 attributes: *Trend*, *Seasonality*, *Noise*, *Local Pattern*, *Correlation*, and *Clustering*. Specifically, we remove one attribute category at a time while keeping the same three-stage progressive strategy. The final models are evaluated on Evaluation Set A of high-level reasoning to measure their Reason performance.

As shown in Figure 7, removing different attributes from Stage 1 leads to varying impacts on the final Reason capability. Among them, removing Correlation, Local Pattern, and Clustering causes the most significant performance degradation. In contrast, Trend and Seasonality have relatively smaller impacts, but still contribute to specific reasoning dimensions. In contrast, removing Noise causes little degradation, showing that Reason capability relies less on noise-related temporal patterns.

5.3 Alignment-Driven Capability Formation

Under the Pre-align architecture, the Temporal Encoder is updated only during Stage 1 and then frozen throughout Stages 2 and 3, while only the input-side alignment modules continue to be optimized. Compared with full progressive training, freezing the encoder causes only limited degradation on task solving in Table 6, with forecasting, imputation, and anomaly detection showing only small changes and classification remaining comparable. A similar pattern is observed for high-level reasoning in Figure 9, where the performance gap between frozen and non-frozen settings remains small across capability stages. Our observation suggests that Stage 1 already provides a sufficiently mature temporal feature space to support subsequent capability development. The P $\rightarrow$ S$\rightarrow$ R formation therefore appears to rely mainly on how fixed temporal features are progressively aligned and reorganized for complex downstream capabilities. In this sense, later-stage capability growth is driven more by the evolution of the alignment process than by further modification of the underlying temporal encoder.

Metric	Non Frozen	Frozen
Fore. MSE ↓	1.328	1.336
Impu. MSE ↓	0.0136	0.0142
Ano. Acc. ↑	0.739	0.718
Class. Acc. ↑	0.628	0.643

Table 6: Effect of freezing the encoder in task solving evaluation.

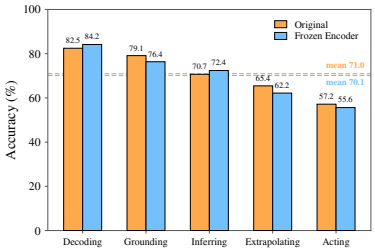


Figure 9: Effect of freezing the encoder in reasoning evaluation.

5.4 Representation Transfer across Stages

We further investigate capability transfer from the representation perspective. Following the three-stage learning process $P \rightarrow S \rightarrow R$, we adopt the representation projection analysis in (Ke et al., 2021; Zhao et al., 2023) to measure the proportion of later-stage representations preserved in the representation space of preceding stages. This transfer ratio quantifies the information transmission between stages, where a higher value indicates stronger representation retention across stages.

We conduct a cross-layer analysis on Qwen3-4B to examine representation transfer patterns. We explore three capability transitions: Perceive-to-Solve (PS), Solve-to-Reason (SR), and Perceive-to-Reason (PR). As shown in Figure 8, different capability transitions exhibit distinct hierarchical

Models	Weather		Finance (7 days)			Finance (30 days)		
	MCQ. ↑	Pre. ↑	3-way ↑	5-way ↑	MSE ↓	3-way ↑	5-way ↑	MSE ↓
TimeOmni-1-7B	41.67	51.40	40.76	34.97	0.622	37.39	20.00	1.490
ITFormer-7B	57.14	36.26	40.14	33.57	0.174	51.03	31.72	0.907
ChatTS-7B	56.21	28.79	19.71	16.09	2.047	19.38	13.46	3.342
PATRA-7B	60.46	52.04	63.33	60.80	0.191	54.39	43.70	0.822
TS-LLM (Pre-align)	67.24	57.17	68.19	57.82	0.187	60.29	41.51	0.837
TS-LLM (Post-align)	65.92	52.76	59.88	55.43	0.160	65.78	47.16	0.744

Table 7: Out-of-domain results on reasoning tasks. Red/blue indicate the 1st/2nd best results.

Method	TimeSeriesExam					CaTS-Bench			
	Ano. Det.	Causality	Noise	Pattern	Similarity	Caption Ret.	TS Ret.	Amp. Comp.	Mean Comp.
Original Paper	28.0	**52.0**	26.0	47.0	45.0	44.0	29.0	43.0	35.0
TimeOmni-1	42.6	37.5	56.0	56.4	46.7	**57.0**	37.0	**92.5**	67.5
TS-LLM (Pre-align)	**49.1**	47.2	**60.7**	61.6	**55.0**	52.0	**46.0**	82.5	**77.5**

Table 8: Results on TimeSeriesExam and CaTS-Bench. **Bold** indicates the best results.

transfer patterns. PS transfer mainly occurs in intermediate layers, suggesting that fundamental temporal structures are transformed into task-solving representations within middle layers. In contrast, SR transfer reaches its peak in deeper layers, indicating that Reason primarily inherits high-level task-oriented representations. PR transfer remains relatively limited, suggesting that advanced reasoning is more likely constructed through the intermediate Solve stage rather than direct capability transfer. Our findings reveal the representation evolution mechanism of Progressive Training: ***different capabilities are not simply stacked together but are progressively transformed across different network depths, forming a three-stage representation development pathway***.

6 Application: Train a Powerful TS-LLM

Based on the above analysis, we adopt a three-stage paradigm, Perceive → Solve → Reason, to train **TS-LLM (Pre-align)** and **TS-LLM (Post-align)**. Stage 1 uses about 105K training samples to learn fundamental temporal attributes, including trends, seasonality, anomalies, local patterns, and multivariate relationships. Stage 2 is built on 158,159 samples, split 8:2 for training and testing, covering forecasting, imputation, anomaly detection, and classification. Stage 3 introduces multi-source data to develop evidence-based temporal analysis. For the **Pre-align** architecture, the Encoder is updated only in Stage 1 and frozen in Stages 2 and 3. Finally, we obtain two powerful TS-LLMs.

We conduct out-of-domain (OOD) experiments on MTBench (Chen et al., 2025) while ensuring that no evaluation data are exposed during training. We compare our models with representative TS-LLMs, including ITFormer, ChatTS, TimeOmni-1, and PATRA (Wang et al., 2025; Xie et al., 2025b; Guan et al., 2025; Lu et al., 2026). More detailed descriptions of the baselines and datasets are provided in Appendix B & C. As shown in Table 7, our two TS-LLMs achieve the best or second-best results on most metrics across both Weather and Finance domains. TS-LLM (Pre-align) performs particularly well on Weather MCQA, weather prediction, and 7-day finance trend prediction, while TS-LLM (Post-align) achieves the strongest results on 7-day finance MSE and most 30-day finance metrics. In addition, we evaluate temporal perception on TimeSeriesExam (Cai et al., 2024) and CaTS-Bench (Zhou et al., 2026). As shown in Table 8, TS-LLM (Pre-align) achieves the best performance on most metrics. our results show that the proposed progressive training strategy generalizes effectively to unseen domains and maintains strong performance across different tasks.

7 Conclusion

Our work investigates how temporal capabilities progressively form in TS-LLMs. We organize model abilities into three stages: Perceive, Solve, and Reason, and show that ordered capability accumulation provides a stronger foundation for high-level reasoning than isolated or mixed training. Further analyses reveal that capability formation is influenced by training order, input format, stage maturity, and cross-layer representation transfer. We also find that local stage convergence does not necessarily indicate sufficient preparation for subsequent capability development. These observations provide a clearer understanding of how lower-level temporal abilities support more advanced reasoning. Based on these findings, we develop a progressive training strategy for TS-LLMs, which achieves strong out-of-domain performance and offers a practical pathway for building more comprehensive temporal reasoning models.

AI USE STATEMENT

Generative AI tools were used solely for language polishing, including improving grammar, clarity, fluency, and overall readability of the manuscript. They were also used to assist in constructing candidate answer and distractor templates, with the detailed generation and verification procedure described in the Appendix. They were not relied upon to determine the scientific claims, experimental results, or conclusions of this work. All AI-assisted content was carefully reviewed, verified, and, when necessary, further modified by the authors to ensure accuracy and consistency with the original technical content. We take full responsibility for the final manuscript and all of its contents.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide comprehensive implementation and evaluation details in the appendix. Appendix B describes all baseline methods and comparison settings. Appendix C documents the datasets used across the three capability stages, while Appendix A reports detailed data statistics and task distributions. Appendix F specifies the model backbones, hardware environment, optimization settings, training configurations, and evaluation protocols. Appendix D provides the numerical results underlying the figures and analyses in the main paper, and Appendix E reports additional experiments with alternative model scales. We further include representative case studies and failure cases in Appendix G to facilitate qualitative verification of model behavior. Across comparative experiments, we maintain consistent data splits, initialization, training settings, and evaluation procedures whenever applicable to ensure fair and reproducible comparisons.

ETHICS STATEMENT

Our work focuses on methodological and scientific research and does not involve human subjects, animals, personal or sensitive data, or environmentally sensitive materials. We do not identify any specific ethical risks or conflicts of interest associated with this study. All experiments are conducted following standard research practices, and the authors take responsibility for ensuring the integrity, validity, and reliability of the reported results.

REFERENCES

Syeda Nahida Akter, Shrimai Prabhumoye, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, Yejin Choi, and Bryan Catanzaro. Front-loading reasoning: The synergy between pretraining and post-training data. In *International Conference on Learning Representations*, volume 2026, pp. 80909–80928, 2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. Timeseriesexam: A time series understanding exam. *arXiv preprint arXiv:2410.14752*, 2024.

Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan Ash, Akshay Krishnamurthy, and Dylan Foster. The coverage principle: How pre-training enables post-training. In *International Conference on Learning Representations*, volume 2026, pp. 112225–112293, 2026.

Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*, 2025.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

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

Zelin He, Sarah Alnegheimish, and Matthew Reimherr. Harnessing vision-language models for time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21690–21698, 2026.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llms capabilities in time series. In *International Conference on Learning Representations*, volume 2025, pp. 90696–90722, 2025.

Yushan Jiang, Kanghui Ning, Zijie Pan, Xuyang Shen, Jingchao Ni, Wenchao Yu, Anderson Schneider, Haifeng Chen, Yuriy Nevmyvaka, and Dongjin Song. Multi-modal time series analysis: A tutorial and survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6043–6053, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23857–23880, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/680b2a8135b9c71278a09cafb605869e-Paper-Conference.pdf.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34: 22443–22456, 2021.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

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

Hao Liu, Chun Yang, Xiaobin Zhu, et al. Semantic-enhanced time-series forecasting via large language models. In *International Conference on Learning Representations*, volume 2026, pp. 101404–101425, 2026.

Junkai Lu, Peng Chen, Xingjian Wu, Yang Shu, Chenjuan Guo, Christian S Jensen, and Bin Yang. Patra: Pattern-aware alignment and balanced reasoning for time series question answering. *arXiv preprint arXiv:2602.23161*, 2026.

Romain Meunier, Farah Benamara, Véronique Moriceau, Zhongzheng Qiao, and Savitha Ramasamy. Crisists: Coupling social media textual data and meteorological time series for urgency classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16082–16099, 2025.

Vinay Prithyani, Mohsin Mohammed, Richa Gadgil, Ricardo Buitrago, Vinija Jain, and Aman Chadha. On the feasibility of vision-language models for time-series classification. *arXiv preprint arXiv:2412.17304*, 2024.

Xin Qiu, Junlong Tong, Yao Zhang, Yunpu Ma, Wei Zhang, and Xiaoyu Shen. From recognition to understanding: Unlocking cognitive time series reasoning with llms. *arXiv preprint arXiv:2606.22126*, 2026.

Paul Quinlan, Qingguo Li, and Xiaodan Zhu. Chat-ts: Enhancing multi-modal reasoning over time-series and natural language data. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5621–5647, 2026.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang Li, and Zhongyu Wei. Itformer: Bridging time series and natural language for multi-modal qa with large-scale multitask dataset. In *International Conference on Machine Learning*, pp. 63324–63344. PMLR, 2025.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *Proc. VLDB Endow.*, 18(8):2385–2398, April 2025a. ISSN 2150-8097. doi: 10.14778/3742728.3742735. URL https://doi.org/10.14778/3742728.3742735.

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

Xiongxiao Xu, Haoran Wang, Yueqing Liang, Philip S Yu, Yue Zhao, and Kai Shu. Can multimodal llms perform time series anomaly detection? In *Proceedings of the ACM Web Conference 2026*, pp. 5392–5403, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Fangxu Yu, Xingang Guo, Lingzhi Yuan, Haoqiang Kang, Hongyu Zhao, Lianhui Qin, Furong Huang, Bin Hu, and Tianyi Zhou. Tsrbench: A comprehensive multi-task multi-modal time series reasoning benchmark for generalist models. *arXiv preprint arXiv:2601.18744*, 2026.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3718–3727. IEEE, 2023.

Luca Zhou, Pratham Yashwante, Marshall Fisher, Alessio Sampieri, Zihao Zhou, Fabio Galasso, and Rose Yu. Cats-bench: Can language models describe time series? In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34479–34519, 2026.

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

A STATISTICS OVERVIEW

Stage 1: Fundamental Perception. Table 9 summarizes the data used for temporal perception. The training set contains approximately 105K samples from ChatTS, while two independent evaluation sets contain 525 manually constructed questions and 1,616 synthetic questions, respectively. The evaluation covers both categorical and numerical perception tasks.

Split	Source	Samples	Task Type	Metric
Training	ChatTS	∼105K	Temporal Perception	–
Evaluation A	Manual	525	Categorical / Numerical	F1 / Relative Acc.
Evaluation B	Synthetic	1,616	Categorical / Numerical	F1 / Relative Acc.

Table 9: Statistics of the Stage 1 perception datasets.

Stage 2: Temporal Task Solving. Table 10 reports the detailed statistics of the task-solving data. The dataset covers forecasting, imputation, anomaly detection, and classification, with 158,159 samples in total. All samples are split into training and testing sets with an 8:2 ratio, and task-specific metrics are used for evaluation.

Source	Task	Total Samples	Train Samples	Test Samples	Metric
TSQA	Forecasting	42,557	34,046	8,511	MSE
	Imputation	38,657	30,926	7,731	MSE
	Anomaly Detection	37,000	29,600	7,400	Accuracy
	Classification	37,000	29,600	7,400	Accuracy
TS Instruct	Classification	2,945	2,356	589	Accuracy
Total	–	**158,159**	**126,528**	**31,631**	–

Table 10: Detailed statistics of the Stage 2 task-solving datasets.

Stage 3: High-level Reasoning. Table 11 presents the composition of the reasoning data. The training set is collected from multiple sources, including TSQA, ChatTS, TSCognition, and TS Instruct. Two independent evaluation sets are used to assess both QA formats and five reasoning dimensions: Decoding, Grounding, Inferring, Extrapolating, and Acting.

Split	Source	# Samples	Metric
Training	TSQA Reasoning QA	26,163	–
	ChatTS TSEvol + Instruct Follow	29,320	–
	TSCognition	32,869	–
	TS Instruct	12,852	–
Evaluation A	Binary QA	4,547	Accuracy
	Multiple-Choice QA		Accuracy
	Open-ended QA		ROUGE-L / BLEU
Evaluation B	Decoding	6,163	Accuracy
	Grounding		Accuracy
	Inferring		Accuracy
	Extrapolating		Accuracy
	Acting		Accuracy

Table 11: Detailed statistics of the Stage 3 high-level reasoning datasets.

B BASELINE

ChatTS (Xie et al., 2025b). ChatTS is a time series understanding framework that enhances LLMs with instruction-based temporal data covering fundamental patterns such as trends, seasonality, anomalies, and local variations. It converts time series into representations compatible with language models and performs instruction tuning for temporal question answering. We use ChatTS as a representative baseline for evaluating whether models trained on temporal understanding data can generalize to more comprehensive perception and reasoning tasks.

TimeOmni-1 (Guan et al., 2025). TimeOmni-1 is a unified TS-LLM designed to support multiple time series understanding and reasoning tasks within a single framework. It integrates temporal information with language instructions and is trained on diverse time series tasks to improve cross-task generalization. We include TimeOmni-1 as a representative general-purpose TS-LLM baseline for comparing comprehensive temporal perception and reasoning capabilities.

ITFormer (Wang et al., 2025). ITFormer is an end-to-end multimodal framework for time series question answering that jointly models temporal representations and language generation. It directly integrates time series features with the language model to support temporal understanding and reasoning over multimodal inputs. We use ITFormer as a representative alignment-based TS-LLM baseline for evaluating high-level temporal reasoning and out-of-domain generalization.

PATRA (Lu et al., 2026). PATRA is a recent reasoning-oriented TS-LLM developed for complex temporal analysis and question answering. It emphasizes the integration of time series representations with language reasoning to improve performance on diverse temporal reasoning tasks. We include PATRA as a strong recent baseline for comparing the reasoning and generalization capabilities of our progressively trained TS-LLMs.

Qwen3-4B (Yang et al., 2025). Qwen3-4B is the base large language model used in our framework. We adopt it as a non-specialized LLM baseline by directly applying the same input format without progressive time series capability training. This baseline is used to evaluate the temporal understanding and reasoning ability provided by the pretrained language model itself, and to measure the gains introduced by our stage-wise Perceive $\rightarrow$ Solve $\rightarrow$ Reason training strategy.

C DATASET

ChatTS (Xie et al., 2025b). ChatTS constructs fully synthetic time series–text data for temporal understanding and reasoning. Its training data consist of five subsets: UTS, MTS-Shape, MTS-Local, TSEvol, and Instruct Follow. UTS contains 35K samples for basic univariate temporal attributes, including global and local patterns; MTS-Shape contains 35K multivariate samples focusing on global trend correlations; and MTS-Local contains 35K samples emphasizing correlated local fluctuations across variables. In addition, TSEvol provides 24,270 diverse question–answer pairs for temporal reasoning, while Instruct Follow contains 5,050 samples for instruction-following ability. The generated time series cover lengths from 64 to 1024 and include numerical supervision such as extrema, segmented averages, local fluctuation positions and amplitudes, seasonality, trends, and point-wise values. In our framework, the three alignment subsets are mainly used for Stage 1 temporal perception, while TSEvol and Instruct Follow are incorporated into Stage 3 reasoning training.

TSQA (Kong et al., 2025). TSQA is a large-scale time series QA dataset introduced with Time-MQA, containing approximately 200K QA pairs across 12 real-world application domains, including healthcare, environment, energy, finance, transportation, IoT, AIOps, and human activities. It covers five task categories: forecasting, imputation, anomaly detection, classification, and open-ended reasoning. Specifically, it contains 42,557 forecasting, 38,657 imputation, 37,000 anomaly detection, 37,000 classification, and 37,629 open-ended QA samples. The reasoning subset further includes true/false, multiple-choice, and open-ended questions. In our framework, the numerical task subsets are mainly used in Stage 2, while the reasoning QA subset is incorporated into Stage 3.

TS Instruct (Quinlan et al., 2026). TS Instruct is a multimodal instruction-tuning dataset introduced with Chat-TS, pairing diverse time series with natural-language instructions and responses. The training set contains 18,306 samples covering four categories: 9,726 reasoning samples, 3,126 decision-making samples, 2,945 classification samples, and 2,509 mathematical reasoning samples. The dataset is generated using time series properties, visual representations, task descriptors, and

synthetic conversations to support joint reasoning over temporal and textual information. In our framework, the classification subset is used in Stage 2, while the remaining reasoning-oriented samples are incorporated into Stage 3.

TSCognition (Qiu et al., 2026). TSCognition is constructed from 15 public real-world datasets spanning eight domains, including Energy (Buildings-900K[1] and UMass Smart*[2]), Retail (Favorita Sales[3] and Dominick[4]), Healthcare (Project Tycho[5] and JHU CSSE COVID-19[6]), Transportation (LargeST[7] and METR-LA[8]), Climate (SubseasonalClimateUSA[9] and Oikolab Weather[10]), Finance (Bitcoin[11]), Web (Web Traffic[12]), and Industry (WADI[13], C-MAPSS[14], and Paderborn[15]). It organizes time series cognition into five hierarchical tasks: *Decoding*, *Grounding*, *Inferring*, *Extrapolating*, and *Acting*. Decoding focuses on extracting fundamental temporal patterns, Grounding aligns temporal signals with contextual information, Inferring models dependencies and relational structures, Extrapolating extends learned patterns to future dynamics, and Acting integrates preceding reasoning results to support decision making. The sample distribution across domains and tasks is shown in Figure 10, while detailed task-level statistics, including variable numbers, time series lengths, and answer choices, are summarized in Table 12. In our framework, the TSCognition training split is incorporated into Stage 3 to strengthen high-level temporal reasoning capability.

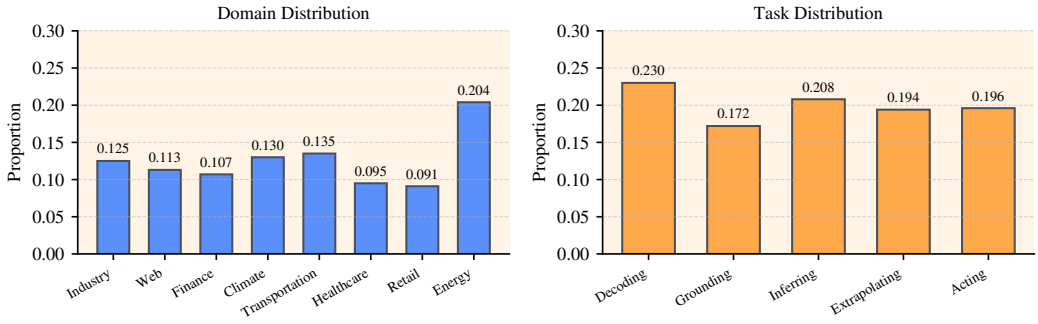


Figure 10: Distribution of TSCognition samples across eight domains and five hierarchical tasks.

Tasks	Variable Numbers			Time Series Length			Choices		
	Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.
Decoding Task	4	5	4.7	437	982	742	4	5	4.7
Grounding Task	3	6	4.4	656	1018	871	4	5	4.3
Inferring Task	3	6	5.1	882	1059	912	4	5	4.6
Extrapolating Task	3	6	5.2	601	1052	846	4	5	4.5
Acting Task	4	6	5.5	468	1047	795	4	5	4.2
Overall	3	6	4.9	437	1059	785	4	5	4.5

Table 12: Task-level statistics of time series in TSCognition.

[1] https://data.openei.org/submissions/5859
[2] https://traces.cs.umass.edu/docs/traces/smartstar/
[3] https://www.kaggle.com/competitions/favorita-grocery-sales-forecasting/data
[4] https://www.chicagobooth.edu/research/kilts/datasets/dominicks
[5] https://www.tycho.pitt.edu/
[6] https://github.com/CSSEGISandData/COVID-19
[7] https://github.com/liuxu77/LargeST
[8] https://github.com/liyaguang/DCRNN
[9] https://github.com/microsoft/subseasonal_data
[10] https://docs.oikolab.com/datasets/
[11] https://zenodo.org/records/5122101
[12] https://www.kaggle.com/competitions/web-traffic-time-series-forecasting/data
[13] https://www.sutd.edu.sg/itrust/itrust-labs/datasets/dataset-characteristics/wadi/
[14] https://data.nasa.gov/dataset/c-mapss-aircraft-engine-simulator-data
[15] https://mb.uni-paderborn.de/en/kat/research/bearing-datacenter

TimeSeriesExam (Cai et al., 2024). TimeSeriesExam is a configurable multiple-choice benchmark designed to evaluate fundamental time series understanding in LLMs. It covers five categories: *pattern recognition*, *noise understanding*, *similarity analysis*, *anomaly detection*, and *causality analysis*. The benchmark is procedurally generated from 104 expert-designed question templates and contains more than 700 questions. Its synthetic time series are constructed from diverse baseline signal components, including linear, nonlinear, and cyclic patterns, allowing controlled evaluation of different temporal properties. The detailed category statistics are summarized in Table 13.

Category	Samples
Anomaly Detection	108
Causality	72
Noise Understanding	84
Pattern Recognition	362
Similarity Analysis	120
Total	**746**

Table 13: Statistics of TimeSeriesExam.

CaTS-Bench (Zhou et al., 2026). CaTS-Bench is a real-world benchmark for context-aware time series captioning and reasoning across 11 diverse domains. Each sample combines a numeric time series segment with contextual metadata, a line-chart visualization, and a natural-language caption. In addition to caption generation, CaTS-Bench provides diagnostic multiple-choice tasks for evaluating fine-grained temporal understanding, including caption retrieval, time series retrieval, amplitude comparison, and mean comparison in Table 14. In our experiments, we use these diagnostic tasks to evaluate whether the learned temporal perception capability generalizes to unseen real-world data and cross-modal reasoning settings.

Task	Samples
Caption Retrieval	100
Time Series Retrieval	100
Amplitude Comparison	40
Mean Comparison	40
Total	**280**

Table 14: Statistics of CaTS-Bench used in our evaluation.

MTBench (Chen et al., 2025). MTBench is a multimodal time series benchmark for evaluating temporal reasoning over numerical signals and textual context in the weather and finance domains. It includes weather question answering and prediction, as well as short- and long-horizon financial trend prediction and forecasting tasks. We use MTBench exclusively for out-of-domain evaluation to assess model generalization to unseen domains and reasoning settings. The evaluation tasks used in our experiments are summarized in Table 15.

Domain	Evaluation Tasks
Weather	MCQA / Prediction
Finance (7 days)	3-way / 5-way / Forecasting
Finance (30 days)	3-way / 5-way / Forecasting

Table 15: Evaluation settings of MTBench.

D Detailed Results

Perception Results. Table 16 reports the detailed categorical and numerical perception results across different methods. It compares performance on trend, seasonality, noise, local patterns, correlation, and clustering, corresponding to the perception analysis shown in Figure 2.

Method	Trend		Season		Noise		Local		Corr.	Clus.
	Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Cate.
Pre-align	0.937	0.865	0.911	0.872	0.857	0.534	0.799	0.700	0.831	0.783
GPT-4o (Text)	0.664	0.839	0.751	0.546	0.654	0.287	0.237	0.165	0.361	0.294
GPT-4o (Vision)	0.691	0.423	0.907	0.582	0.659	0.361	0.415	0.302	0.394	0.411
Post-align	0.922	0.880	0.931	0.834	0.900	0.479	0.824	0.733	0.771	0.804

Table 16: Detailed categorical and numerical perception results.

Reasoning Scaling Results. Table 17 reports the detailed reasoning accuracy under different proportions of Stage 3 training data, corresponding to Figure 3. For both Pre-align and Post-align architectures, progressive Stage (1→2→3) training consistently outperforms Stage 3 Only and Mixed Training, with a larger advantage under limited reasoning supervision. The Directly setting shows weaker overall performance, further indicating that explicit temporal representation learning benefits progressive capability formation.

Model	Training Setting	15%	25%	50%	75%	100%
Pre-align	Stage 3 Only	0.414	0.464	0.527	0.586	0.619
	Mixed Training	0.479	0.520	0.583	0.673	0.689
	Stage (1→2→3)	0.528	0.574	0.598	0.664	0.710
Post-align	Stage 3 Only	0.403	0.453	0.524	0.585	0.617
	Mixed Training	0.461	0.502	0.570	0.664	0.678
	Stage (1→2→3)	0.520	0.561	0.614	0.680	0.697
Directly	Stage 3 Only	0.402	0.436	0.478	0.517	0.546
	Mixed Training	0.439	0.497	0.530	0.588	0.608
	Stage (1→2→3)	0.451	0.490	0.542	0.579	0.614
	Stage (1→3)	0.429	0.482	0.519	0.572	0.599
	Stage (2→3)	0.447	0.469	0.535	0.563	0.604
	No Training	0.351	0.351	0.351	0.351	0.351

Table 17: Detailed reasoning accuracy under different Stage 3 training data scales.

Stage-wise Saturation Results. Table 18 provides the detailed results corresponding to Table 4. Stage 1 and Stage (1→2) reach local saturation at approximately 70% of the corresponding training data, achieving 98.3% and 98.7% of their full-stage performance, respectively. Stage (1→2→3) reaches a similar saturation region at 85%, with 97.2% of its full-stage performance. These results further show that local convergence can occur before the complete progressive pathway reaches its best performance.

Training Setting	Data Scale	Evaluation Performance	Relative Perf.	Full Progressive
Stage 1	40%	0.613	–	0.751
	55%	0.662	–	
	70%	**0.738**	**98.3%**	
	85%	0.745	–	
	100%	0.751	100%	
Stage (1→2)	40%	0.542	–	0.684
	55%	0.603	–	
	70%	**0.644**	**98.7%**	
	85%	0.649	–	
	100%	0.652	100%	
Stage (1→2→3)	40%	0.537	–	0.710
	55%	0.569	–	
	70%	0.616	–	
	85%	**0.663**	**97.2%**	
	100%	0.682	100%	

Table 18: Detailed stage-wise performance under increasing training data scales. For Stage (1→2), Stage 1 is fixed at its 70% saturation point; for Stage (1→2→3), Stages 1 and 2 are both fixed at 70%. **Bold** denotes the identified local saturation points.

E Supplementary Experiments

To examine whether the observed progressive capability formation is specific to the Qwen3-4B backbone, we further conduct supplementary experiments using Qwen3-1.7B. We follow the same

three-stage training paradigm and evaluation protocol as the main experiments. The results across perception, task solving, and high-level reasoning show similar trends, indicating that the progressive capability relationships remain consistent under a smaller LLM backbone.

Stage 1: Fundamental Attribute Perception. Table 19 reports the Stage 1 perception results of Pre-align and Post-align. They show strong performance across categorical and numerical temporal attributes, including trend, seasonality, noise, local patterns, correlation, and clustering.

Model	Trend		Season		Noise		Local		Multivariate	
	Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Num.	Corr.	Clus.
Pre-align	0.894	0.843	0.859	0.843	0.822	0.558	0.786	0.727	0.788	0.761
Post-align	0.873	0.846	0.892	0.834	0.861	0.506	0.752	0.709	0.755	0.747

Table 19: Stage 1 perception results on Qwen3-1.7B.

Stage 2: Temporal Task Solving. Table 20 presents the Stage 2 results. Stage 1-only training already improves all four downstream tasks over the non-fine-tuned Qwen3-1.7B baseline. Further task-specific supervision provides substantial gains, while Stage (1→2) achieves competitive or better performance on most metrics.

Model	Training Setting	Fore. MSE ↓	Impu. MSE ↓	Ano. Det. Acc. ↑	Class. Acc. ↑
Qwen3-1.7B	No Fine-tuning	1.884	0.0312	0.303	0.234
Pre-align	Stage 1 Only	1.813	0.0218	0.548	0.421
	Stage 2 Only	**1.412**	0.0151	0.651	0.542
	Stage (1→2)	1.438	**0.0144**	**0.684**	**0.576**
Post-align	Stage 1 Only	1.782	0.0235	0.574	0.408
	Stage 2 Only	1.469	0.0162	**0.648**	0.507
	Stage (1→2)	**1.391**	**0.0150**	0.626	**0.559**

Table 20: Stage 2 task-solving results on Qwen3-1.7B.

Stage 3: High-level Reasoning. Table 21 reports the reasoning results on the first Stage 3 evaluation set. Compared with Stage 3 Only, incorporating preceding perception or task-solving stages consistently improves reasoning performance. The complete Stage (1→2→3) pathway achieves the strongest overall results across the five reasoning dimensions.

Model	Training Setting	Decoding	Grounding	Inferring	Extrapolating	Acting
Pre-align	Stage 3 Only	65.8	63.7	54.2	53.6	46.6
	Stage (1→3)	73.8	**76.9**	60.7	58.9	49.4
	Stage (2→3)	74.9	69.5	**65.6**	58.1	52.1
	Stage (1→2→3)	**78.1**	74.2	63.1	**65.2**	**55.6**
Post-align	Stage 3 Only	62.9	62.8	53.1	55.4	47.7
	Stage (1→3)	72.5	70.3	59.4	61.5	50.9
	Stage (2→3)	70.4	69.8	**64.2**	63.4	**57.3**
	Stage (1→2→3)	**75.4**	**74.3**	61.6	**67.7**	53.2

Table 21: Stage 3 reasoning results on the first evaluation set using Qwen3-1.7B.

Overall, the Qwen3-1.7B exhibit trends consistent with the main Qwen3-4B results. Fundamental perception provides transferable temporal knowledge for downstream task solving, while the combination of perception and task-solving supervision further supports high-level reasoning. These results suggest that the observed capability formation is not restricted to a specific model scale.

F REPRODUCIBILITY

We adopt unified implementation, training, and evaluation settings whenever possible to ensure fair and reproducible comparisons in Talbe 22. Unless otherwise specified, all experiments are conducted on two NVIDIA A800 GPUs using PyTorch, HuggingFace Transformers, DeepSpeed, and Accelerate. We use Qwen3-4B as the primary LLM backbone and additionally conduct supplementary experiments on Qwen3-1.7B. The same data splits, stage definitions, evaluation protocols, and prompt formats are maintained within each group of comparative experiments. We fix the random seed to 2026 and use bfloat16 mixed-precision training for all LLM-based experiments.

Configuration	**Setting**
Primary backbone	Qwen3-4B
Additional backbone	Qwen3-1.7B
GPUs	$2 \times$ NVIDIA A800
Framework	PyTorch + HuggingFace Transformers
Distributed training	DeepSpeed + Accelerate
Precision	bfloat16
Optimizer	AdamW
Learning rate	2×10^{-5}
Weight decay	1×10^{-2}
Batch size per GPU	4
Gradient accumulation steps	4
Effective batch size	32
Gradient clipping	1.0
Learning rate scheduler	Cosine decay
Warmup ratio	0.0
Random seed	2026

Table 22: Implementation and training configurations used in our experiments.

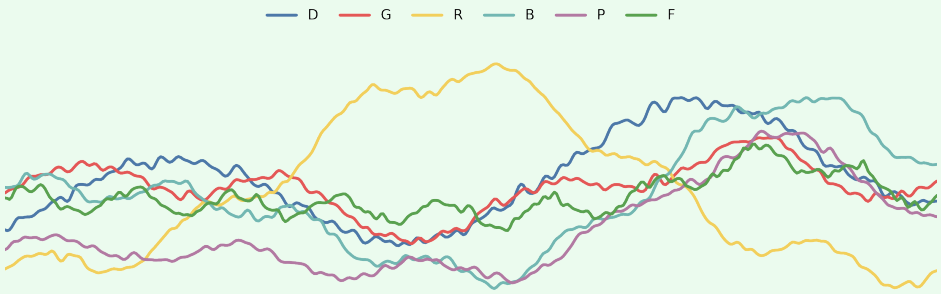
All capability stages are trained using supervised fine-tuning with the same optimization configuration unless otherwise specified. For the Pre-align architecture, the Temporal Encoder and alignment module are optimized in Stage 1, while the Temporal Encoder is frozen in subsequent stages and the remaining alignment components are updated. For the Post-align architecture, the Temporal Encoder and LLM backbone are jointly optimized. For the textualization setting, the numerical time series are directly converted into textual sequences and only the LLM is fine-tuned. Across different capability-development paths, we keep the model architecture, data split, initialization, and evaluation procedure unchanged so that performance differences mainly reflect the training stage and supervision strategy. This ensures a controlled comparison across different training paradigms.

For evaluation, Stage 1 separately measures categorical and numerical temporal perception, Stage 2 evaluates forecasting, imputation, anomaly detection, and classification, and Stage 3 evaluates high-level temporal reasoning. Forecasting and imputation use MSE, anomaly detection and classification use accuracy, while the reasoning dimensions are evaluated using multiple-choice accuracy when applicable. All reported comparative results are obtained under the same evaluation protocol within each experiment. This provides consistent evaluation criteria across all capability stages.

G Case Study and Failure Analysis

Example of the Decoding task

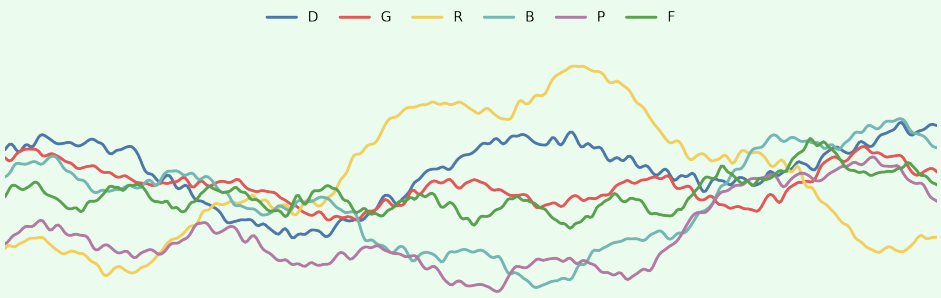
Instruction: You are an expert in time series analysis and temporal reasoning. A temporal token representing the input time series is provided to the model. Please answer the multiple-choice question based on the temporal evidence, context, and candidate answers. Only one answer is correct.
Variable Information: D: Load Demand; G: Thermal Generation; R: Renewable Output; B: Battery Power; P: Electricity Price; F: Frequency Deviation.



Context: The data describe the operation of a regional power system. The operator monitors load demand, thermal generation, renewable output, battery behavior, electricity price, and frequency deviation to evaluate supply-demand balance and dispatch decisions. Battery power reflects storage operation, while electricity price and frequency deviation may indicate system stress. The actual temporal state should be inferred from the provided time series representation.
Question: Which statement best characterizes the joint temporal behavior of renewable output and battery power?
Candidate Answers:
A. Renewable output stays low throughout the period, while the battery remains inactive with no clear charging or discharging phase.
B. Renewable output reaches its highest level only during the final high-demand period, while the battery mainly charges at the same time.
C. Renewable output and battery power increase monotonically together, indicating that storage output always follows renewable generation.
D. Renewable output first increases to a mid-period peak and then declines, while the battery tends to charge during the renewable-rich phase and discharge later.
E. Renewable output fluctuates randomly without a recognizable phase, while battery behavior shows no relation to the system state.
Output: Please output the correct candidate answer.
Answer: *D. Renewable output forms a clear mid-period peak before decreasing. Battery power shifts from charging during the renewable-rich phase to discharging afterward.*

Example of the Grounding task

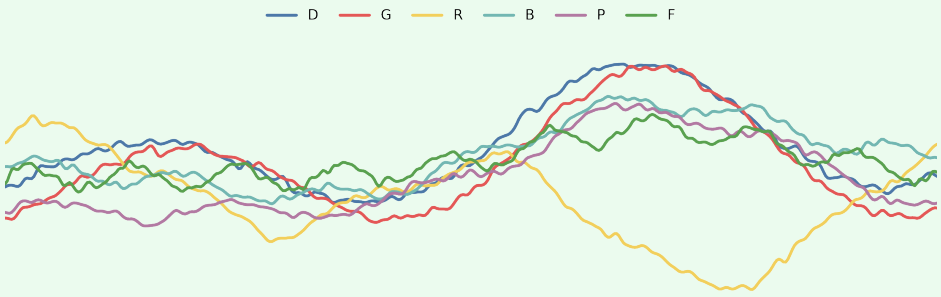
Instruction: You are an expert in time series analysis and temporal reasoning. A temporal token representing the input time series is provided to the model. Please answer the multiple-choice question based on the temporal evidence, context, and candidate answers. Only one answer is correct.
Variable Information: D: Load Demand; G: Thermal Generation; R: Renewable Output; B: Battery Power; P: Electricity Price; F: Frequency Deviation.



Context: The data describe the operation of a regional power system. The operator monitors load demand, thermal generation, renewable output, battery behavior, electricity price, and frequency deviation to evaluate supply-demand balance and dispatch decisions. Battery power reflects storage operation, while electricity price and frequency deviation may indicate system stress. The actual temporal state should be inferred from the provided time series representation.
Question: Which interpretation best grounds the observed temporal patterns in the operational context of power dispatch?
Candidate Answers:
A. A renewable-rich period is reflected by higher renewable output together with battery charging, suggesting that surplus renewable energy is being absorbed.
B. A renewable-rich period is reflected by high electricity price and large frequency deviation, because these always indicate abundant renewable supply.
C. A peak-load period is reflected by low demand and strong battery charging, suggesting that the system is under severe supply shortage.
D. A stable operating period is reflected by simultaneous increases in load demand, electricity price, and frequency deviation, regardless of renewable output.
E. The storage behavior cannot be grounded in the power-system context because battery power is unrelated to renewable output or demand.
Output: Please output the correct candidate answer.
Answer: *A. The high-renewable interval coincides with sustained battery charging. This pattern indicates that the battery is absorbing surplus renewable energy.*

Example of the Inferring task

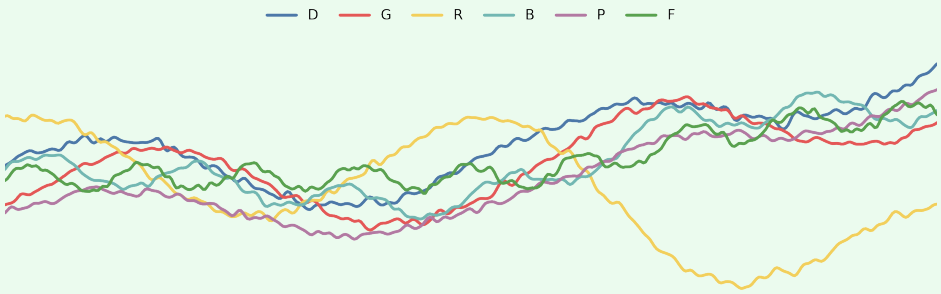
Instruction: You are an expert in time series analysis and temporal reasoning. A temporal token representing the input time series is provided to the model. Please answer the multiple-choice question based on the temporal evidence, context, and candidate answers. Only one answer is correct.
Variable Information: D: Load Demand; G: Thermal Generation; R: Renewable Output; B: Battery Power; P: Electricity Price; F: Frequency Deviation.



Context: The data describe the operation of a regional power system. The operator monitors load demand, thermal generation, renewable output, battery behavior, electricity price, and frequency deviation to evaluate supply-demand balance and dispatch decisions. Battery power reflects storage operation, while electricity price and frequency deviation may indicate system stress. The actual temporal state should be inferred from the provided time series representation.
Question: What is the most plausible explanation for the system pressure indicated by electricity price and frequency deviation during the high-load phase?
Candidate Answers:
A. The pressure is mainly caused by decreasing demand, since lower load usually forces the system to increase electricity price and frequency deviation.
B. The pressure is likely caused by the coincidence of high load demand, reduced renewable support, and increased dependence on dispatchable resources.
C. The pressure is mainly caused by excessive renewable generation, because abundant renewable output always increases electricity price and frequency deviation.
D. The pressure is unrelated to supply-demand balance and is only caused by random fluctuations in the battery power curve.
E. The pressure disappears because battery discharge and thermal generation fully eliminate all frequency deviation.
Output: Please output the correct candidate answer.
Answer: *B. Load demand rises while renewable output weakens during the high-pressure phase. The concurrent increases in thermal generation, electricity price, and frequency deviation indicate greater reliance on dispatchable resources.*

Example of the Extrapolating task

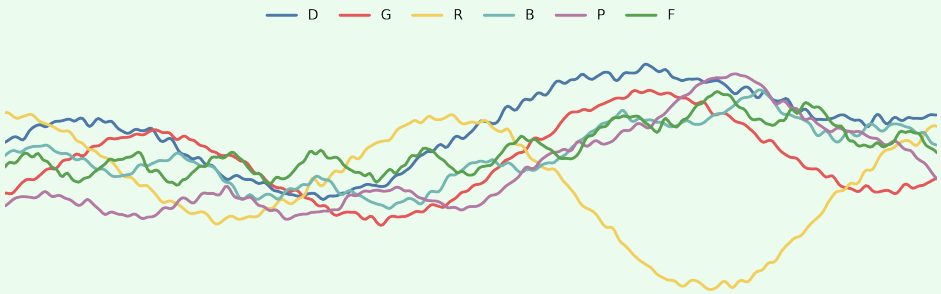
Instruction: You are an expert in time series analysis and temporal reasoning. A temporal token representing the input time series is provided to the model. Please answer the multiple-choice question based on the temporal evidence, context, and candidate answers. Only one answer is correct.
Variable Information: D: Load Demand; G: Thermal Generation; R: Renewable Output; B: Battery Power; P: Electricity Price; F: Frequency Deviation.



Context: The data describe the operation of a regional power system. The operator monitors load demand, thermal generation, renewable output, battery behavior, electricity price, and frequency deviation to evaluate supply-demand balance and dispatch decisions. Battery power reflects storage operation, while electricity price and frequency deviation may indicate system stress. The actual temporal state should be inferred from the provided time series representation.
Question: Assume that the current high-load phase continues for the next interval and renewable output does not recover. Which future evolution is most consistent with the observed temporal evidence?
Candidate Answers:
A. The system will remain risk-free even if demand keeps increasing, because frequency deviation is independent of supply-demand imbalance.
B. Renewable output will necessarily rebound immediately, so electricity price and frequency deviation should both drop to their minimum values.
C. Battery power is more likely to switch to strong charging, because a high-load phase implies surplus electricity in the system.
D. Thermal generation will become unnecessary, since declining renewable output reduces the need for dispatchable resources.
E. Electricity price and frequency deviation may further increase unless additional dispatchable generation or storage discharge is available.
Output: Please output the correct candidate answer.
Answer: *E. The final interval shows persistently high demand and continuously weakening renewable support. If this trend continues without additional controllable supply, electricity price and frequency deviation are likely to increase further.*

Example of the Acting task

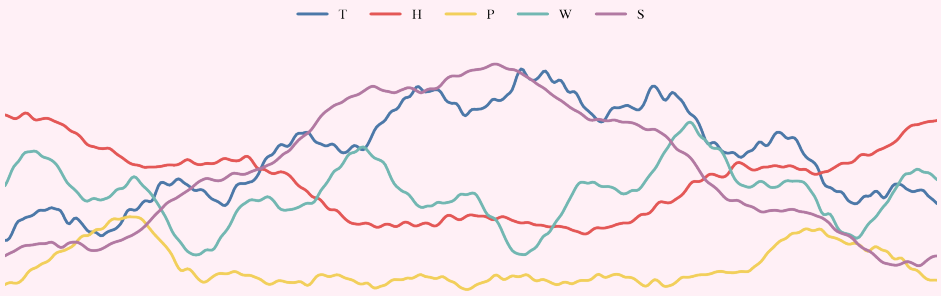
Instruction: You are an expert in time series analysis and temporal reasoning. A temporal token representing the input time series is provided to the model. Please answer the multiple-choice question based on the temporal evidence, context, and candidate answers. Only one answer is correct.
Variable Information: D: Load Demand; G: Thermal Generation; R: Renewable Output; B: Battery Power; P: Electricity Price; F: Frequency Deviation.



Context: The data describe the operation of a regional power system. The operator monitors load demand, thermal generation, renewable output, battery behavior, electricity price, and frequency deviation to evaluate supply-demand balance and dispatch decisions. Battery power reflects storage operation, while electricity price and frequency deviation may indicate system stress. The actual temporal state should be inferred from the provided time series representation.
Question: Which operational decision best matches the observed system state and dispatch objective?
Candidate Answers:
A. Increase battery charging and reduce dispatchable generation, because the system appears to have large surplus supply during the high-load phase.
B. Maintain the current dispatch without intervention, because electricity price and frequency deviation provide no useful signal about operating pressure.
C. Discharge the battery and increase dispatchable generation to relieve the high-load pressure while monitoring frequency stability.
D. Shut down renewable generation completely, because removing renewable output is the most direct way to stabilize price and frequency.
E. Reduce both thermal generation and battery output, because lower controllable supply will automatically reduce demand-side pressure.
Output: Please output the correct candidate answer.
Answer: *C. High demand and weak renewable output coincide with elevated electricity price and frequency deviation. Increasing dispatchable generation and battery discharge is therefore the appropriate response to relieve system pressure.*

Failure Case

Instruction: You are an expert in time series analysis and temporal reasoning. A temporal token representing the input time series is provided to the model. Please answer the multiple-choice question based on the temporal evidence, context, and candidate answers. Only one answer is correct.
Variable Information: T: Surface Temperature; H: Relative Humidity; P: Precipitation; W: Wind Speed; S: Solar Radiation.



Context: The data describe the operation of a regional weather monitoring system. The operator monitors surface temperature, relative humidity, precipitation, wind speed, and solar radiation to evaluate atmospheric conditions and weather evolution. Temperature and solar radiation reflect thermal conditions, while humidity and precipitation characterize moisture variation. The actual temporal state should be inferred from the provided time series representation.
Question: Which statement best describes the temporal relationship between surface temperature and solar radiation over the observed period?
Candidate Answers:
A. Surface temperature gradually increases throughout the period, while solar radiation decreases after an early peak, indicating a delayed thermal response to weakening radiation.
B. Solar radiation increases toward the middle of the period and then decreases, while surface temperature follows a similar pattern with a warming phase followed by cooling.
C. Surface temperature decreases during the period, while solar radiation reaches its maximum value near the end, suggesting reduced thermal input and delayed warming.
D. Solar radiation remains relatively stable with small fluctuations, while surface temperature exhibits strong variations unrelated to radiation changes.
E. Surface temperature and solar radiation both increase continuously throughout the period, showing a consistent upward relationship between thermal conditions and radiation intensity.
Answer: *E. Surface temperature and solar radiation exhibit a broadly aligned upward trend despite several local fluctuations.*
Ground Truth: *B.*
Failure Analysis: *The model captures the initial increasing trend but fails to recognize the subsequent reversal shared by surface temperature and solar radiation. It therefore overemphasizes the early local pattern and incorrectly predicts a continuous increase. This failure indicates difficulty in tracking coordinated phase transitions across variables over the full temporal horizon.*