TokenDecouple: Frequency-aware Token Merging and Pyramidal Text Decay

Xin Qiu[1] **Jialin Gan**[1,2*]
[1]Zhejiang University
[2]Zhejiang University-University of Illinois Urbana-Champaign Institute
qiuxin688@gmail.com

Abstract

Large language models (LLMs) have enabled time series (TS) analysis by jointly modeling numerical observations and textual context through a shared token interface. However, TS tokens and prompt tokens exhibit fundamentally different information structures, making uniform token processing inefficient. In this paper, we study token efficiency in TS language modeling from an asymmetric-token perspective. We show that TS tokens have highly uneven spectral contributions, where many tokens share redundant frequency patterns while a small subset preserves critical temporal evidence. We also observe that prompt-token influence attenuates with model depth, suggesting that full prompt retention across all layers is unnecessary. Based on these findings, we develop an adaptive token budgeting framework that compresses TS tokens via frequency-domain structure and progressively reduces prompt tokens across layers. Experiments across forecasting, classification, imputation, and anomaly detection demonstrate up to ***7.68***× inference acceleration and performance gains in ***78%*** of evaluated settings, showing the effectiveness of asymmetric token compression for scalable TS foundation models. **Our code and datasets.**

1 Introduction

Large Language Models (LLMs) have reshaped the landscape of NLP, exhibiting remarkable proficiency in understanding and reasoning (Matarazzo and Torlone, 2025). As models scale in both capacity and data diversity, their influence has extended well beyond language, enabling powerful multimodal modeling that unify information from audio, images, and video (Han et al., 2024; Zhang et al., 2025b; Guo et al., 2025). Therefore, a growing body of research integrates LLMs into cross-modal modeling between text and time series (TS), enabling strong performance on event forecasting, causal reasoning, anomaly detection (Wang et al., 2024; Pan et al., 2024; Chang et al., 2025). For example, CrisisTS aligns crisis-related textual reports with meteorological TS signals to improve multilingual urgency assessment (Meunier et al., 2025). Time-MQA supports multi-task TS question answering (QA) by coupling language reasoning with continual pre-training on temporal data (Kong et al., 2025). ITFormer provides an end-to-end pipeline that connects TS modeling with language generation, enabling multimodal QA for engine-maintenance scenarios (Wang et al., 2025).

Although LLMs have begun to show encouraging progress on TS tasks, the nature of temporal signals remains different from language and is still underexplored. TS are highly structured, exhibit strong temporal continuity, and often undergo pronounced distribution shifts in real-world environments (He et al., 2024; Lu et al., 2026). Different frequency components of TS become dispersed in the time domain, yet LLMs lack mechanisms for recognizing these distinct frequency roles (Zhou et al., 2022). This mismatch obscures the true functional contribution of each token and complicates efficient temporal representation learning. Therefore, a key open problem is how to effectively uncover the deeper information embedded within TS tokens, which is both challenging and essential.

To address this issue, it is essential to reexamine TS tokens from a frequency-domain perspective. Our analysis reveals several key mechanisms. Specifically, large groups of tokens share highly similar spectral patterns within certain frequency bands, which manifest inside the model as nearly indistinguishable attention behaviors. In contrast, only a small subset of tokens carries dominant information at specific frequencies. Motivated by these observations, *we introduce the first token-level frequency-based merging strategy for LLM-based TS systems*. By isolating true information-carrying "executors" and removing redundancy.

*Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 40718–40741
October 24-29, 2026 ©2026 Association for Computational Linguistics

In addition, TS are highly abstracted: a local blackout may manifest only as a sudden drop or near-zero collapse in electricity consumption, and a scandal involving a CEO as a downward trend in stock price. Because such rich causes collapse into simple numerical patterns, LLM-based TS systems typically rely on substantial text prompts to provide contextual guidance. These prompts both compensate for the compressed nature of TS data and help activate the model's reasoning ability (Jin et al., 2024; Gruver et al., 2023; Qiu and Liu, 2026). However, introducing large numbers of prompt tokens inevitably incurs substantial computational overhead (Dao et al., 2022; Kwon et al., 2023). This trade-off gives rise to a central challenge: how to reduce token usage but preserving essential information.

Building on this insight, *We reveal a surprisingly **"pyramidal decay"** effect in prompt tokens*: their influence diminishes rapidly as model depth increases. Motivated by this observation, we introduce a prompt token compression strategy that aggressively reduces prompt tokens in early layers. Moreover, we find that textual information is fully absorbed and transferred to TS tokens within only a few shallow layers (at most three layers), indicating that prompt tokens need not propagate through the entire model. As a result, the substantial computational cost can be drastically reduced with negligible impact on model performance. Our contributions can be summarized as follows:

1. We provide *the first systematic frequency-domain perspective on TS tokens*, revealing how different frequency components contribute unevenly to token-level information.

2. We uncover *a pyramidal decay effect in prompt tokens*, showing that their influence rapidly diminishes with model depth and can be safely compressed in early layers.

3. We propose ***TokenDecouple***, *a unified framework for decoupled compression of TS and prompt tokens*, enabling substantial efficiency gains while even improving performance.

2 Related Work

LLMs for TS Analysis. TS analysis covers a wide range of tasks, including forecasting, anomaly detection, classification, and question-answering (QA) (Xie et al., 2025; Liu et al., 2025a,c). Recently, LLMs have been explored as a general-purpose solution for them. A direct way is to feed raw TS points into the LLM, but it fails to bridge the modality gap between TS and text (Gruver et al., 2023; Liu et al., 2024b). Another line of work converts TS into images and relies on vision language Models (VLMs), but they often struggle with fine-grained temporal patterns (Zhong et al., 2025; Ruan et al., 2025). Consequently, the dominant approach is to embed TS into representation space, and then align TS tokens with text tokens to construct LLM-based TS analysis systems. For example, Time-MMD (Liu et al., 2024a), GPT4MTS (Jia et al., 2024), ChatTS (Xie et al., 2025), and TimerBed (Kong et al., 2025) combine TS data with domain knowledge to support TS tasks. Because TS data are already irreversibly compressed during collection, extensive prompts are required to activate LLMs' prior knowledge, but combining them with high-dimensional TS tokens incurs prohibitive computational costs.

TS Representation Analysis. Compared with the relatively sparse spatial distribution of text, TS exhibits a more continuous structure and more concealed semantic features (Zhang et al., 2025a; Yue et al., 2025). In theory, the types of TS patterns are nearly impossible to enumerate, with semantic information hidden within complex trends, cycles, and fluctuations (Li et al., 2025; Qin et al., 2025). While some studies have explored TS from various perspectives, deep analyses remain rare (Yuan and Chen, 2024; He et al., 2025). One key reason is that most existing analyses focus on the time-domain, where multiple semantic factors are entangled within continuous signals, making systematic characterization difficult (Wu et al., 2023; Gan et al., 2026). In contrast, the frequency-domain distribution of TS is more concentrated and discretized. Motivated by these challenges, we conduct a frequency-domain analysis of TS token in LLM-based TS analysis systems, revealing their distribution characteristics and modes of information contribution.

3 Preliminary

3.1 LLM-based TS Analysis System

A standard LLM-based TS analysis system consists of four components: a TS embedder, a text tokenizer, an LLM backbone, and a task-specific head. The TS embedder transforms raw TS signals into TS tokens, and the text tokenizer maps text prompts into prompt tokens. These two token streams are

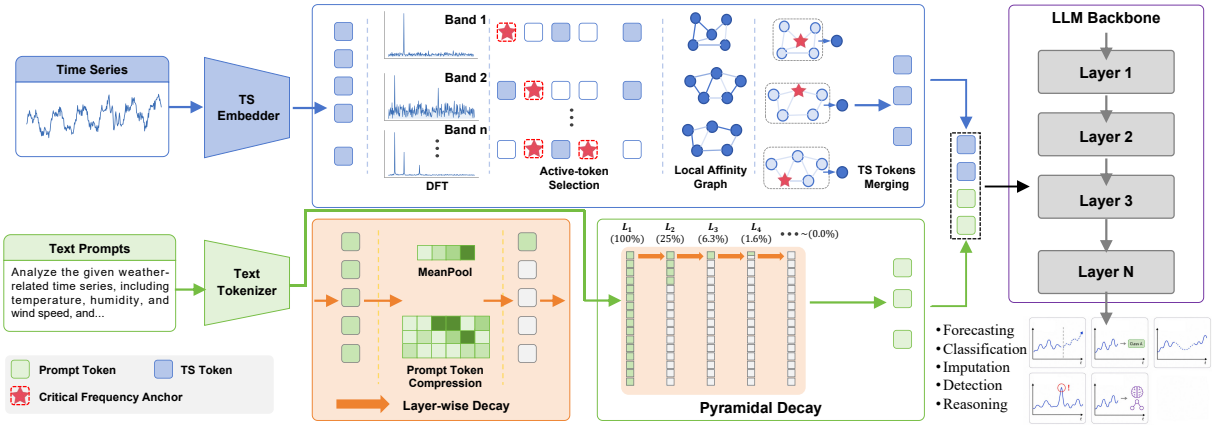


Figure 1: Overview of ***TokenDecouple***. We decouple token compression for TS and prompt tokens by merging redundant TS tokens in the frequency-domain and reducing prompt tokens under the pyramidal decay effect.

then fused within the LLM backbone, which produces the final outputs through a task-specific head. However, because the LLM backbone is pre-trained exclusively on text data, a fundamental distribution shift exists between TS and language. As a result, an effective LLM-based TS analysis system must establish a robust alignment between TS data and text data. Existing approaches follow two major directions: ***(1) input-space alignment***, where TS–text reprogramming is performed before tokens enter the LLM. This reshapes TS tokens into a language-compatible format, allowing them to interact effectively with prompt tokens (Jin et al., 2024); and ***(2) latent-space alignment***, where the distribution mismatch is addressed by lightly fine-tuning a small subset of LLM parameters (Zhou et al., 2023; Tang et al., 2025; Chang et al., 2025). It reduces modality drift while minimizing catastrophic forgetting (French, 1999).

3.2 TS Tasks: From Patterns to Reasoning

In real-world, TS reflect phenomena with varying levels of complexity. This natural hierarchy motivates separating TS analysis into two major groups. The first consists of ***pattern-centric tasks***, where solving the problem depends primarily on understanding the observable structure of the sequence. The second consists of ***reasoning-centric tasks***, where the model must infer unobserved generative factors beyond the visible temporal patterns.

Pattern-centric tasks aim to interpret observable TS patterns and follow a natural progression of difficulty from (i)imputation to (ii)classification, (iii)anomaly detection, and (iv)forecasting. Interpolation primarily relies on strong local continuity and requires minimal reasoning. Classification depends on holistic and multi-scale representations to capture class-specific patterns. Anomaly detection requires modeling normality to identify deviations. Forecasting is the most demanding, as it requires generating unobserved future values while capturing trends, periodicity, and short-term dynamics. Overall, these tasks capture increasingly complex temporal dependencies.

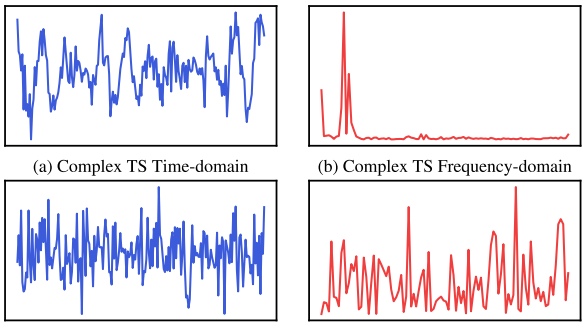
Reasoning-centric tasks move beyond pattern recognition and require the model to infer the causal structure, categorical logic, latent events, or unobserved factors. (i) Simple deterministic reasoning represents the easiest level, where decisions are governed by single-cue determination and rely on salient local patterns. (ii) Complex deterministic reasoning is harder, as outcomes depend on the joint configuration of multiple physiological or domain-specific features rather than any single cue. (iii) Probabilistic reasoning is the most challenging, requiring the model to handle probability distributions, ambiguity, and latent variables.

4 TokenDecouple: Decoupled Compression of TS and Prompt Tokens

As shown in Fig. 1, ***TokenDecouple*** decouples token compression for TS tokens and prompt tokens. Specifically, it performs frequency-domain merging for TS tokens, while applying pyramidal compression to prompt tokens to reduce redundant textual overhead in shallow layers.

4.1 Two Approaches: Time-Domain vs. Frequency-Domain

Due to the large computational burden imposed by TS tokens, effective compression is essential for efficient TS analysis. Token merging has emerged



(a) Complex TS Time-domain (b) Complex TS Frequency-domain

(c) White Noise Time-domain (d) White Noise Frequency-domain

Figure 2: The complex TS appears entangled in the time domain but shows concentrated energy in a few frequency bands, while white noise has a nearly uniform spectrum, indicating poor compressibility.

as a common strategy. Most existing methods operate in the time-domain, for example by merging adjacent intervals, lowering temporal resolution, or pooling local segments (Liu et al., 2022b; Nie et al., 2023). However, local time-domain segments entangle multi-scale information. As a result, it does not separate functional information, making reliable information preservation difficult.

In contrast, the frequency domain provides a more measurable representation of TS, where physical components such as trends, periodic patterns, and disturbances often manifest as interpretable spectral structures (Zhou et al., 2022; Liu et al., 2022b). As an illustrative baseline, white noise represents an extreme case of spectral dispersion. For zero-mean white noise, the autocorrelation reduces to a Dirac delta and the corresponding power spectral density (PSD) is constant: $S_x(\omega) = \sigma^2$. It implies that signal energy is uniformly distributed over the entire frequency range, rendering the frequency-domain representation non-compressible. Fortunately, real-world TS are rarely purely random and contain structured components, as shown in Fig. 2. According, a canonical decomposition of a TS can be expressed as:

$$x(t) = m(t) + \sum_{k=1}^{K} a_k \cos(\omega_k t + \phi_k) + \varepsilon(t), \quad (1)$$

where $m(t)$ denotes a slowly varying trend component, $\sum_{k=1}^{K} a_k \cos(\omega_k t + \phi_k)$ captures periodic patterns, and $\varepsilon(t)$ represents stochastic disturbances. Correspondingly, the PSD of such signals can be approximated as:

$$S_x(\omega) \approx S_m(\omega) + \sum_{k=1}^{K} c_k \delta(\omega - \omega_k) + S_\varepsilon(\omega), \quad (2)$$

indicating that the majority of signal energy is concentrated in a small number of frequency bands. Therefore, the frequency-domain offers a principled space for removing redundancy.

4.2 TS Token Merging

In the following, we investigate TS token merging from a frequency-domain perspective. Given the TS tokens produced by a TS embedder, we denote $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^D$, where N denotes the number of tokens and D is the feature dimension. We perform a discrete Fourier transform (DFT) along the token axis, mapping the token sequence from the time-domain into the frequency-domain as: $\mathbf{F} = \{\mathbf{F}_k\}_{k=0}^{N-1}, \mathbf{F}_k \in \mathbb{C}^D$.

Due to the conjugate symmetry of real-valued TS in the frequency-domain, we retain only the non-negative frequency components $k = 0, \ldots, K-1$, where $K = \lfloor N/2 \rfloor + 1$. For the b-th sample, its spectral magnitude is $\hat{F}_{b,i,k} = |F_{b,i,k}|$. We sort tokens and select the minimal subset whose cumulative magnitude accounts for at least 95% of the total magnitude at that frequency. Tokens in this subset are treated as the active token set $\mathcal{I}_{b,k}$, and its size is denoted by $n_{b,k} = |\mathcal{I}_{b,k}|$.

The next core question is *how to determine the merging granularity for each frequency component?* Specifically, we denote by $m_{b,k}$ the number of TS tokens after merging, which satisfies $1 \leq m_{b,k} \leq n_{b,k}$. In practice, selecting an appropriate ratio is difficult. So we introduce an adaptive frequency merging strategy. Our key idea is to quantify the structural contribution of each frequency component through its *spectral identifiability*, estimated from locally stationary token segments, where tokens enabling sharper peaks or low-noise periodic patterns contribute more.

Let $I_{b,i}(\omega)$ denote the periodogram of this local segment, and let $S(\omega; \boldsymbol{\theta}_{b,k})$ be a parametric PSD model with parameters $\boldsymbol{\theta}_{b,k}$. Using the Whittle likelihood approximation, the local log-likelihood is defined as:

$$\ell_{b,i} = -\sum_{\omega \in \Omega_k} \left(\log S(\omega; \boldsymbol{\theta}_{b,k}) + \frac{I_{b,i}(\omega)}{S(\omega; \boldsymbol{\theta}_{b,k})} \right), \quad (3)$$

where Ω_k denotes a frequency neighborhood centered. We measure how strongly token i supports the identification through the Fisher information (Casella and Berger, 2024):

$$\mathcal{I}_{b,i,k} = -\mathbb{E}\left[\nabla^2_{\boldsymbol{\theta}_{b,k}} \ell_{b,i}(\boldsymbol{\theta}_{b,k}) \right], \quad (4)$$

and use its trace $s_{b,i,k} = \mathrm{tr}(\mathcal{I}_{b,i,k})$ as the contribution. We aggregate contributions into a score:

$$S_{b,k} = \frac{1}{n_{b,k}} \sum_{i \in \mathcal{I}_{b,k}} s_{b,i,k}, \tag{5}$$

Then, we convert scores into a merging strength $r_{b,k}$ via a sigmoid mapping. The adaptive merging budget is determined as: $m_{b,k} = \lceil (1 - r_{b,k}) \cdot n_{b,k} \rceil$. We introduce a normalized regularization term to encourage compact representations while keeping task performance: $L_{\text{pen}} = \lambda \sum_k m_{b,k} / \sum_k n_{b,k}$. After determining the merging budget $m_{b,k}$, we first build a sparse local graph by computing similarities only within a neighborhood $|i - j| < r_{\text{loc}}$:

$$\begin{aligned} \mathrm{sim}_{i,j} &= \mathrm{sim}\big(\phi(|F_{i,k}|),\, \phi(|F_{j,k}|)\big), \\ A_{i,j} &= \exp(\mathrm{sim}_{i,j}/\tau)\,, \end{aligned} \tag{6}$$

and the affinity matrix $A_{i,j}$ constrains which tokens can be merging. Let $\{\mathcal{G}_{b,k}^{(u)}\}_{u=1}^{m_{b,k}}$ denote a partition of the active indices $\mathcal{I}_{b,k}$ into $m_{b,k}$ local groups. For each group, the merged representative is:

$$\begin{aligned} \omega_i^{(u,k)} &= \frac{\sum_{j \in \mathcal{G}_{b,k}^{(u)}} A_{i,j}}{\sum_{p \in \mathcal{G}_{b,k}^{(u)}} \sum_{q \in \mathcal{G}_{b,k}^{(u)}} A_{p,q}}, \\ \tilde{\mathbf{F}}_{u,k} &= \sum_{i \in \mathcal{G}_{b,k}^{(u)}} \omega_i^{(u,k)} \mathbf{F}_{i,k}, \end{aligned} \tag{7}$$

Finally, $\{\tilde{\mathbf{F}}_{u,k}\}_{k=0}^{K-1}$ are transformed back to the time-domain via an IDFT along the frequency axis.

4.3 Prompt Tokens Compression with Pyramidal Decay

In typical TS–text tasks, text prompts provide high-level guidance rather than describing numerical variations. As TS representations progressively abstract from low-level patterns to higher-level semantics, textual guidance plays a stronger role in shallow layers but diminishes with depth, where representation learning increasingly relies on the TS itself. Consequently, retaining full prompt tokens across all layers offers limited benefit while incurring additional overhead, suggesting that prompt influence should be gradually absorbed into TS representations at early stages.

Let the prompt tokens at layer l be $\mathbf{H}_P^{(l)} \in \mathbb{R}^{P_l \times D}$, where the number of prompt tokens is reduced as $P_{l+1} = \lceil 0.25\, P_l \rceil$. We first compress the prompt tokens $\mathbf{S}^{(l)} = \mathbf{W}^{(l)} \mathbf{H}_P^{(l)}$, with $\mathbf{W}^{(l)} \in \mathbb{R}^{P_{l+1} \times P_l}$ whose rows are normalized by a softmax. To preserve global prompt information, we additionally compute a residual summary $\mathbf{r}^{(l)} = \mathrm{MeanPool}(\mathbf{H}_P^{(l)})$. The prompt tokens for the next layer is then updated as:

$$\mathbf{H}_P^{(l+1)} = \mathbf{S}^{(l)} + \mathbf{1}_{P_{l+1}} \mathbf{r}^{(l)}. \tag{8}$$

As depth increases, the number of prompt tokens decays rapidly (25%, 6.3%, 1.6%, 0.4%, ...) relative to the input. Therefore, from around the fourth layer, the computational overhead is negligible.

5 Experiments

5.1 Preparations

LLM-based TS Models. We conduct experiments on four advanced models, including OFA (Zhou et al., 2023), TimeLLM (Jin et al., 2024), CALF (Liu et al., 2025c) and S^2IP (Pan et al., 2024), referring to App. B.1. By spanning a wide range of TS–text alignment strategies for adapting LLMs to time series analysis, this evaluation provides a rigorous and challenging testbed.

TS Analysis Tasks and Datasets. Based on the 7 subtasks in Sec. 3.2, including ***Pattern-centric Tasks (Imputation, Classification, Anomaly Detection, and Forecasting)*** and ***Reasoning-centric Tasks (Simple Deterministic, Complex Deterministic, and Probabilistic)***, we collect a total of 27 datasets. These datasets span a wide range of domains, including climate, energy, transportation, economics, physiology, chemistry, speech, machine monitoring, cybersecurity, and industrial systems, all sourced from real-world settings.

Baseline Token Merging Methods. We include three advanced token merging methods as baselines. ***Global token merging for TS*** (Shao et al., 2026) reduces the number of tokens at each layer by merging the most similar token pairs based on global similarity. ***Local token merging for TS*** (Bolya et al., 2023) introduces locality constraints by computing token similarity and performing merging only within adjacent or local windows, which preserves temporal ordering and local dependency structures. ***Dynamic token merging*** (Feng and Zhang, 2023) adaptively adjusts the merging ratio according to token similarity, allowing the merging process to flexibly respond to different inputs and model depths. Please refer to App. B.2.

5.2 Evaluation on Pattern-centric Tasks

Setups. **For the imputation task**, we evaluate four models on six datasets. For each dataset, we randomly mask $\{12.5\%, 25\%, 50\%\}$ of time points from TS segments of length 96 to control the missing data ratio, following the protocol of TimesNet. MSE and MAE are adopted as evaluation metrics. This results in 12 comparative experiments per dataset and 72 comparative experiments across 6 datasets, comparing the original models with TokenDecouple (TS token merging + Prompt token compression). **For the classification task**, we evaluate four models on five datasets using Accuracy as the evaluation metric, resulting in 20 comparative experiments. **For the anomaly detection task**, four models are evaluated on five datasets with F1-score as the metric. To ensure a fair comparison, only the classical reconstruction error is adopted, consistent with the setting in TimesNet, leading to 20 comparative experiments. **For the forecasting task**, we adopt two prediction horizons $H \in \{192, 720\}$. The input TS length T is set to 336, 512, 96, and 512 for OFA, TimeLLM, CALF, and S^2IP, respectively, following their original papers. MSE and MAE are used as evaluation metrics. Each dataset involves 8 experiments, resulting in 40 comparative experiments.

Results. Across the four tasks, the TokenDecouple models outperform the original models in 76%, 60%, 85%, and 85% of the cases, respectively, as shown in Tab. 1. Importantly, even in the loss cases, the maximum performance degradation remains minor across all tasks. Specifically, the largest increase in imputation MAE is only 0.008 (2.2%), the maximum drop in classification accuracy is 1.3%, the largest decrease in anomaly detection F1-Score is 0.41%, and for forecasting, the maximum MAE increase is 0.01 (5.0%).

5.3 Evaluation on Reasoning-centric Tasks

Setups. The reasoning-centric tasks cover three levels of complexity. Simple Deterministic Reasoning includes Right Whale Call detection (RCW) and Transient Electromagnetic Event classification (TEE), where class labels correspond to well-defined single patterns identified by human annotators. Complex Deterministic Reasoning involves ECG record diagnosis and EMG signal diagnosis, requiring label assignment based on the joint interpretation of multiple temporal patterns by medical experts. Probabilistic Reasoning comprises Human Activity Recognition (HAR) and Computer Type Usage classification (CTU), in which class boundaries are inherently uncertain due to stochastic feature–label relationships.

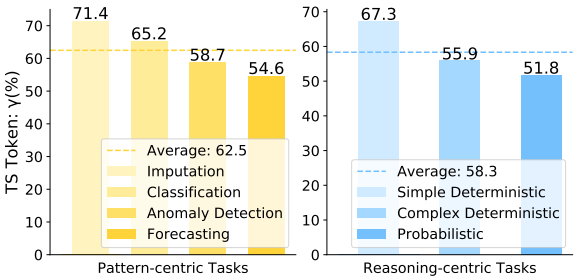


Figure 3: Average TS token merging ratio γ (%), induced by frequency-domain merging across seven tasks.

Results. We compare the performance of the original models and TokenDecouple across four models and six subtasks, as shown in Tab. 2. Among the 24 task settings, TokenDecouple achieves better performance in 17 cases, accounting for 71% . Notably, even in the few cases with performance drops, the maximum accuracy degradation is only 3.2%.

5.4 Efficiency Evaluation of TokenDecouple

Computational Cost of TS Tokens. Given N_{TS} TS tokens, token merging reduces them to $\hat{N}_{\text{TS}}$ tokens. We define the merging ratio as $\gamma = 1 - \hat{N}_{\text{TS}}/N_{\text{TS}}$, where a larger γ indicates a higher degree of merging. Fig. 3 shows the average γ of four models across seven tasks. As task difficulty increases, γ gradually decreases, reflecting the growing amount of information required. Nevertheless, even for the most challenging tasks, at least 50% the TS tokens can still be merged.

Computational Cost of Prompt Tokens. In addition, given N_{Text} prompt tokens, their number follows a pyramidal decay across layers, taking values of 100%, 25%, 6.3%, 1.6%, and 0.4%. Specifically, the number of prompt tokens at layer l is $N_{\text{Text}}^{(l)} = 0.25^{l} N_{\text{Text}}$. The total number of prompt tokens processed across L layers is $\sum_{l=1}^{L} N_{\text{Text}}^{(l)} = N_{\text{Text}}(1 - 0.25^{L})/(1 - 0.25)$, leading to an average per layer of $\bar{N}_{\text{Text}} = \frac{N_{\text{Text}}}{L(1-0.25)}(1 - 0.25^{L})$. When $L \geq 4$, the term 0.25^{L} is negligible, yielding $\bar{N}_{\text{Text}} \approx \frac{4}{3L} N_{\text{Text}}$. Accordingly, for an L-layer LLM, the prompt token compression ratio is $\eta_{\text{L}} = 1 - \bar{N}_{\text{Text}}/N_{\text{Text}} \approx 1 - \frac{4}{3L}$. Consequently, prompt-induced overhead decreases sharply with model depth, with $\eta_{12} \approx 89\%$ and $\eta_{24} \approx 94\%$ for 12-layer and 24-layer models, respectively.

Table 1: Performance comparison between the original models (Ori) and TokenDecouple (TD) on pattern-centric tasks. Across 152 comparative experiments, TokenDecouple achieves better performance in 118 cases, accounting for 78%, while significantly reducing token-level computational cost. In 1.3% of the cases, the performance remains on par with the original models. For the Imputation task, MAE is averaged over three masking ratios. For the Forecasting task, MAE is averaged over two prediction horizons.

Models	OFA		TimeLLM		CALF		S^2IP		Wins	Draws	Losses	Total	Win Rate↑
	Ori	TD	Ori	TD	Ori	TD	Ori	TD					
Imputation Task (Metric: avg. MAE↓)													
ETTh1	0.166	0.162	0.089	0.087	0.302	0.297	0.135	0.126	11	0	1	12	92%
ETTh2	0.131	0.128	0.141	0.138	0.378	0.363	0.104	0.106	9	0	3	12	75%
ETTm1	0.097	0.095	0.071	0.073	0.306	0.300	0.092	0.091	9	0	3	12	75%
ETTm2	0.079	0.074	0.069	0.067	0.285	0.279	0.043	0.044	8	1	3	12	67%
Weather	0.050	0.047	0.043	0.040	0.122	0.120	0.044	0.040	10	0	2	12	83%
Electricity	0.197	0.197	0.079	0.078	0.237	0.225	0.187	0.178	8	1	3	12	67%
Wins: 55, Draws: 2, Losses: 15, average Win Rate: 76%													
Classification Task (Metric: Accuracy↑ (%))													
JapaneseVowels	98.2	98.2	30.1	31.2	95.9	95.3	93.6	93.8	2	1	1	4	50%
Handwriting	35.2	35.9	12.5	12.3	27.1	28.9	27.7	30.4	3	0	1	4	75%
Heartbeat	76.8	77.1	72.9	72.4	78.7	77.4	72.1	72.7	2	0	2	4	50%
EthanolConcentration	34.8	35.0	31.2	32.9	29.6	29.8	29.7	29.4	3	0	1	4	75%
PEMS-SF	84.2	85.5	25.1	27.8	78.9	78.5	82.8	82.5	2	0	2	4	50%
Wins: 12, Draws: 1, Losses: 7, average Win Rate: 60%													
Anomaly Detection Task (Metric: F1-Score↑ (%))													
SMD	85.74	85.79	84.12	84.60	86.54	87.08	83.34	83.55	4	0	0	4	100%
MSL	85.11	85.68	69.76	70.12	84.68	84.62	80.22	80.96	3	0	1	4	75%
SMAP	70.17	72.32	70.44	71.18	68.60	68.87	71.85	72.26	4	0	0	4	100%
SWaT	93.09	92.86	91.63	92.48	94.56	94.91	83.46	85.11	3	0	1	4	75%
PSM	97.76	97.35	92.46	93.33	96.24	96.89	95.61	96.54	3	0	1	4	75%
Wins: 17, Draws: 0, Losses: 3, average Win Rate: 85%													
Forecasting Task (Metric: avg. MAE↓)													
Traffic	0.416	0.404	0.402	0.397	0.461	0.458	0.404	0.405	7	0	1	8	88%
Solar	0.198	0.204	0.175	0.166	0.250	0.248	0.191	0.186	5	0	3	8	63%
PEMS04	0.513	0.509	0.486	0.485	0.779	0.770	0.581	0.577	6	0	2	8	75%
PEMS08	0.566	0.548	0.557	0.538	0.859	0.855	0.734	0.725	8	0	0	8	100%
Exchange	0.511	0.490	0.553	0.538	0.503	0.486	0.599	0.590	8	0	0	8	100%
Wins: 34, Draws: 0, Losses: 6, average Win Rate: 85%													

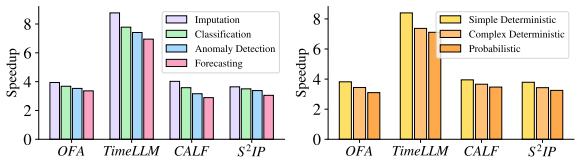


Figure 4: Inference speedup of four models with TokenDecouple across seven tasks.

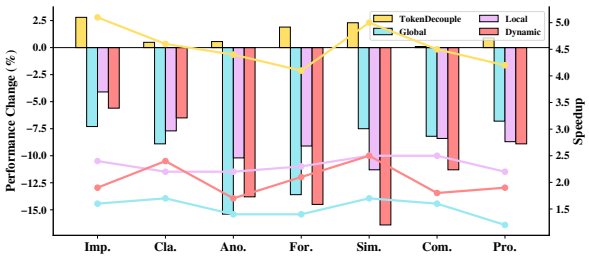


Figure 5: Performance change (bar chart, "+" indicates improvement and "-" indicates degradation) and inference speedup (line chart) of TokenDecouple and three token merging methods, averaged over four models across seven tasks.

Inference Speedup. Under identical experimental settings, we evaluate the inference-time acceleration brought by TokenDecouple, as illustrated in Fig. 4. Consistent inference speedups are observed across seven tasks, with OFA, TimeLLM, CALF, and S^2IP achieving average speedups of **3.55×**, **7.68×**, **3.53×**, and **3.43×**, respectively. TS merging ratios positively correlate with inference speedups, while model architectures further influence the overall acceleration.

5.5 Comparison with Other Methods

Fig. 5 compares TokenDecouple with three representative token merging methods, namely global merging, local merging, and dynamic merging, in terms of both performance and inference speedup. While all three baselines exhibit noticeable performance degradation across tasks, TokenDecouple consistently preserves competitive performance. Moreover, TokenDecouple achieves an average inference speedup of **4.55×**, substantially surpassing global, local, and dynamic merging, which obtain average speedups of 1.2×, 2.2×, and 1.9×, respectively. These results indicate that TokenDecouple effectively balances efficiency and performance.

Table 2: Performance comparison between the original models (Ori) and TokenDecouple (TD) on reasoning-centric tasks, evaluated using accuracy(%). Across 24 comparative experiments, TokenDecouple achieves better performance in 17 cases, accounting for 71%.

Tasks	Simple Deterministic				Complex Deterministic				Probabilistic			
	RCW		TEE		ECG		EMG		CTU		HAR	
	Ori	TD	Ori	TD	Ori	TD	Ori	TD	Ori	TD	Ori	TD
OFA	78.20	80.25	62.31	64.39	21.39	20.44	75.45	75.27	69.88	71.62	87.64	88.25
TimeLLM	62.50	64.53	47.28	50.36	25.59	23.68	63.84	65.26	60.20	61.84	88.43	85.22
CALF	74.56	77.48	53.44	57.86	24.24	23.55	66.73	69.31	55.62	56.08	84.97	86.60
S^2IP	81.50	78.43	59.16	64.28	27.45	30.54	82.68	80.27	64.19	66.39	75.22	77.44

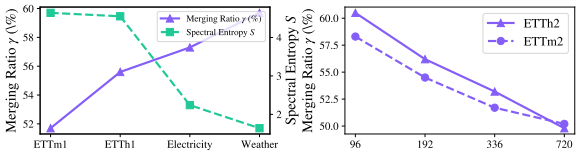


Figure 6: Merging ratio γ under different datasets or prediction horizons, averaged over four models.

Table 3: Ablation analysis. **A**: Original models; **B**: A - text prompt; **C**: A + prompt token compression; **D**: A + TS token merging; **E**: A + prompt token compression + TS token merging. {**Imp.**, **For.**} metric: avg. MSE over four models (lower is better); {**Ano.**} metric: avg. F1-score (%) over four models (higher is better); {**Cla.**, **Sim.**, **Com.**, **Pro.**} metric: avg. accuracy (%) over four models (higher is better).

Types	Imp.	Cla.	Ano.	For.	Sim.	Com.	Pro.
A	0.144	55.86	84.27	0.487	64.87	48.42	73.26
B	0.152	54.15	82.16	0.513	58.22	41.39	62.39
C	0.144	55.60	83.96	0.495	64.55	48.30	72.91
D	0.141	56.42	84.79	0.477	67.11	48.46	73.98
E	0.140	56.35	84.83	0.479	67.20	48.54	74.18

5.6 Ablation Study

Based on the ablation analysis shown in Tab. 3, we verify that incorporating text prompts benefits LLM-based TS analysis systems (A vs. B). Moreover, A vs. C and A vs. D show that prompt token compression and TS token merging each introduce only limited performance degradation when applied individually. When combined, they achieve a better trade-off between cost and performance.

6 Discussion & Conclusion

TS Token Merging Ratio Decreases with Increasing Task and Dataset Difficulty. As shown in Fig. 3, the ratio γ exhibits a clear decreasing trend across tasks of increasing difficulty. Furthermore, focusing on the forecasting task, we observe that γ also decreases as the **prediction horizon** becomes longer or as the dataset distribution exhibits higher **spectral entropy**. This indicates more TS tokens is required under higher prediction difficulty or increased data complexity, as shown in Fig. 6.

Table 4: MAE of Original (Ori) & TokenDecouple (TD) under different SNR levels (dB) on forecasting tasks with a prediction horizon of 192.

Models		OFA			TimeLLM		
SNR(dB)		10	20	30	10	20	30
Traffic	Ori	0.438	0.423	0.408	0.424	0.405	0.381
	TD	0.417	0.411	0.404	0.410	0.399	0.383
Exchange	Ori	0.258	0.218	0.192	0.277	0.233	0.215
	TD	0.230	0.214	0.196	0.256	0.228	0.211

Robustness Analysis of TokenDecouple. Taking the forecasting on the Weather and Exchange as examples, we inject additive Gaussian white noise with SNR $\in \{30, 20, 10\}$ dB to examine whether TokenDecouple exhibits failure. Tab. 4 reports the performance of OFA and TimeLLM under different SNR levels, both with and without TokenDecouple. We observe that as the noise level increases (i.e., SNR decreases), the $|\mathrm{MAE}_{\mathrm{Ori}} - \mathrm{MAE}_{\mathrm{TD}}|$ enlarges. This indicates that token merging enhances robustness to noise-induced frequency perturbations.

Conclusion. Our work revisits LLM-based TS analysis from a token-centric perspective. We show that TS tokens contain frequency-domain redundancy, while prompt tokens exhibit a pyramidal decay effect. Based on these findings, we propose ***TokenDecouple***, which decouples TS token merging and prompt token compression. Extensive experiments show that TokenDecouple improves inference efficiency while maintaining or enhancing performance, highlighting the value of token-aware design for efficient TS–language integration.

Limitations

Although TokenDecouple demonstrates a favorable efficiency–performance trade-off across multiple time series tasks, this study does not systematically evaluate its applicability or robustness under highly non-stationary or irregularly sampled conditions, which are common in certain real-world scenarios. Our experiments primarily focus on general time series settings, and the behavior of the proposed token decoupling strategy in more extreme, noisy, or rapidly evolving environments remains unclear. In addition, our study is limited to token-level compression during inference and does not examine its interaction with other efficiency-oriented techniques, such as structured pruning, quantization, or model-level compression and acceleration. As a result, the effectiveness of TokenDecouple in extremely large-scale models or strictly resource-constrained settings has not been fully assessed.

Ethics Statement

This work focuses on improving the efficiency of LLM-based time-series analysis through token compression and does not involve the collection of private user data or human-subject experiments. All evaluated datasets are publicly available benchmarks or standard research datasets. While TokenDecouple can reduce computational cost and resource consumption, its use in safety-critical applications should still consider the reliability and potential errors of the underlying models.

References

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2485–2494.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*.

George Casella and Roger Berger. 2024. *Statistical inference*. Chapman and Hall/CRC.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2025. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–20.

Marco Cuturi. 2011. Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 929–936.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Zhanzhou Feng and Shiliang Zhang. 2023. Efficient vision transformer via token merger. *IEEE Transactions on Image Processing*, 32:4156–4169.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Jialin Gan, Xin Qiu, Guangzhe Chen, and Xue Wang. 2026. Beyond uniform tokens: Adaptive compression for time series language models. *arXiv preprint arXiv:2606.13624*.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

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

Yuxin Guo, Shuailei Ma, Shijie Ma, Xiaoyi Bao, Chen-Wei Xie, Kecheng Zheng, Tingyu Weng, Siyang Sun, Yun Zheng, and Wei Zou. 2025. Aligned better, listen better for audio-visual large language models. In *International Conference on Learning Representations*, volume 2025, pages 4247–4271.

Soyeon Caren Han, Feiqi Cao, Josiah Poon, and Roberto Navigli. 2024. Multimodal large language models and tunings: Vision, language, sensors, audio, and beyond. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11294–11295.

Cheng He, Xijie Liang, Zengrong Zheng, Patrick PC Lee, Xu Huang, Zhaoyi Li, Hong Xie, Defu Lian, and Enhong Chen. 2025. A unified frequency domain decomposition framework for interpretable and robust time series forecasting. *arXiv preprint arXiv:2510.10145*.

Hui He, Qi Zhang, Kun Yi, Xiaojun Xue, Shoujin Wang, Liang Hu, and Longbing Cao. 2024. Robust multivariate time series forecasting against intra- and inter-series transitional shift. *arXiv preprint arXiv:2407.13194*.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. 2024. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23343–23351.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and 1 others. 2024. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pages 23857–23880.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. 2025. 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)*, pages 29736–29753.

Mineichi Kudo, Jun Toyama, and Masaru Shimbo. 1999. Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters*, 20(11-13):1103–1111.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

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

James Large, E Kate Kemsley, Nikolaus Wellner, Ian Goodall, and Anthony Bagnall. 2018. Detecting forged alcohol non-invasively through vibrational spectroscopy and machine learning. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 298–309. Springer.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32.

Zhengnan Li, Yuting Tan, Xilong Cheng, and Yunxiao Qin. 2025. Ftmixer: Frequency and time domain representations fusion for time series forecasting. *IEEE Signal Processing Letters*.

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

Haoxin Liu, Chenghao Liu, and B Aditya Prakash. 2025b. A picture is worth a thousand numbers: Enabling llms reason about time series via visualization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7486–7518.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and 1 others. 2024a. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B Aditya Prakash. 2024b. Lst-prompt: Large language models as zero-shot time series forecasters by long-short-term prompting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7832–7840.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. 2022a. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in neural information processing systems*, 35:5816–5828.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. 2025c. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18915–18923.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. 2022b. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Junkai Lu, Peng Chen, Chenjuan Guo, Yang Shu, Meng Wang, and Bin Yang. 2026. Towards non-stationary time series forecasting with temporal stabilization and frequency differencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24070–24078.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*.

Aditya P Mathur and Nils Ole Tippenhauer. 2016. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pages 31–36. IEEE.

Romain Meunier, Farah Benamara, Véronique Moriceau, Zhongzheng Qiao, and Savitha Ramasamy. 2025. 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)*, pages 16082–16099.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. S2ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Zhenkai Qin, Baozhong Wei, Caifeng Gao, Xiaolong Chen, Hongfeng Zhang, and Cora Un In Wong. 2025. Sfdformer: a frequency-based sparse decomposition transformer for air pollution time series prediction. *Frontiers in Environmental Science*, 13:1549209.

Xin Qiu and Wenjie Liu. 2026. Rpgfusion: 4d radar prior-guided multi-modal fusion for 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 284–294.

Weilin Ruan, Siru Zhong, Haomin Wen, and Yuxuan Liang. 2025. Vision-enhanced time series forecasting via latent diffusion models. *arXiv preprint arXiv:2502.14887*.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2026. Holitom: Holistic token merging for fast video large language models. *Advances in Neural Information Processing Systems*, 38:135547–135570.

Mohammad Shokoohi-Yekta, Bing Hu, Hongxia Jin, Jun Wang, and Eamonn Keogh. 2017. Generalizing dtw to the multi-dimensional case requires an adaptive approach. *Data mining and knowledge discovery*, 31(1):1–31.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837.

Jialiang Tang, Shuo Chen, Chen Gong, Jing Zhang, and Dacheng Tao. 2025. Empowering large language models for time series forecasting with patterns and semantics. In *2025 IEEE International Conference on Data Mining (ICDM)*, pages 733–742. IEEE.

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

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

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

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

Xiaobing Yuan and Ling Chen. 2024. D-pad: Deep-shallow multi-frequency patterns disentangling for time series forecasting. *arXiv preprint arXiv:2403.17814*.

Wenzhen Yue, Yong Liu, Xianghua Ying, Bowei Xing, Ruohao Guo, and Ji Shi. 2025. Freeformer: frequency enhanced transformer for multivariate time series forecasting. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 3606–3614.

Qianru Zhang, Yuting Sun, Honggang Wen, Peng Yang, Xinzhu Li, Ming Li, Kwok-Yan Lam, Siu-Ming Yiu, and Hongzhi Yin. 2025a. Time series analysis in frequency domain: A survey of open challenges, opportunities and benchmarks. *arXiv preprint arXiv:2504.07099*.

Shaolei Zhang, Qingkai Fang, Yang Yang, and Yang Feng. 2025b. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, volume 2025, pages 53285–53310.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. 2025. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. In *International Conference on Machine Learning*, pages 78478–78497. PMLR.

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, and 1 others. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355.

A Broader Impacts

This work aims to improve the efficiency of LLM-based time-series analysis by reducing redundant TS and prompt tokens. The proposed method can lower inference latency and computational cost, which may benefit real-world applications that require efficient temporal modeling, such as healthcare monitoring, energy management, financial analysis, industrial maintenance, and climate-related forecasting. By reducing the token burden of LLM-based TS systems, TokenDecouple may also help decrease GPU memory consumption and energy usage, making such models more accessible in resource-constrained settings.

However, more efficient TS analysis systems may also amplify risks when deployed in high-stakes domains. For example, inaccurate predictions or reasoning results in healthcare, finance, or infrastructure monitoring could lead to harmful decisions if used without expert verification. In addition, compression-based methods may discard subtle but important temporal signals under severe distribution shifts or rare-event scenarios. Therefore, we recommend that TokenDecouple be used as an efficiency-enhancing component rather than an autonomous decision-making system. For safety-critical applications, model outputs should be accompanied by uncertainty estimation, domain-specific validation, and human oversight.

B Baselines

B.1 LLM-based TS Models

OFA (Zhou et al., 2023). OFA is an LLM-based time series model that transfers pretrained transformer backbones from NLP to time series analysis by freezing self-attention and feedforward layers while selectively fine-tuning embeddings and normalization layers. It redesigns the input embedding to project normalized and patched time series into the token space of pretrained models, enabling efficient cross-domain adaptation with minimal parameter updates. This design demonstrates that frozen pretrained transformers can be effectively repurposed for time series tasks through lightweight architectural modifications.

TimeLLM (Jin et al., 2024). TimeLLM reprograms a frozen pretrained large language model, such as LLaMA or GPT-2, for time series forecasting without fine-tuning the backbone. It transforms normalized and patched time series into patch embeddings and aligns them with the language model via a patch reprogramming mechanism using a small set of learned text prototypes, enabling effective cross-modal alignment. Additional prompt-as-prefix conditioning is used to inject task instructions, dataset context, and input statistics, while only lightweight input transformation and output projection modules are trained.

CALF (Liu et al., 2025c). CALF is an LLM-based time series model that adopts a dual-branch architecture consisting of a textual source branch and a temporal target branch with shared pretrained LLM weights. It aligns time series inputs with the language embedding space through a cross-modal match module based on principal word embeddings, while further enforcing alignment via intermediate feature regularization and output consistency losses. By explicitly matching representations across modalities, CALF aims to reduce modality gaps and better adapt pretrained language models to time series data.

S^2IP (Pan et al., 2024). S^2IP is an LLM-based time series forecasting model that integrates time series tokenization with semantic space informed prompting. It decomposes normalized time series into trend, seasonal, and residual components, applies patching to obtain time series embeddings, and aligns them with a reduced set of semantic anchors derived from pretrained word embeddings. The top-K relevant semantic anchors are retrieved as prefix prompts to guide a frozen LLM, while only lightweight input/output mappings and selected embedding components are fine-tuned.

B.2 Token Merging Methods

Global Token Merging (Casella and Berger, 2024). Global token merging for time series reduces the number of tokens at each layer by merging the most similar token pairs based on a global similarity metric, e.g., cosine similarity $s_{ij} = \frac{\mathbf{t}_i \cdot \mathbf{t}_j}{\|\mathbf{t}_i\| \|\mathbf{t}_j\|}$. At each merging step, token pairs with the highest similarity scores are combined, regardless of their temporal positions.

Local Token Merging (Bolya et al., 2023). Local token merging for time series introduces locality constraints by restricting similarity computation and merging operations to adjacent or local temporal windows. Tokens are only merged if $|i-j| \leq w$, where w denotes the window size. This design preserves temporal ordering and local dependency structures that are critical for time series modeling.

Dynamic Token Merging (Feng and Zhang, 2023). Dynamic token merging adaptively adjusts the merging ratio according to token similarity and model depth. Instead of using a fixed reduction rate, the number of merged tokens is determined by the distribution of similarity scores, allowing the effective token count N_l at layer l to vary with the input and representation complexity.

C Datasets

C.1 Datasets for Imputation Task

ETTh1 (Zhou et al., 2021). ETTh1 is a used public benchmark for long-term time series task, designed to model temperature variations and load dynamics in electric transformers. It contains 2 years of hourly observations collected from 2016/06 to 2018/06 , where each record includes a timestamp, the target variable oil temperature, and six load-related covariates. Following the standard setting, the dataset is split into 12 months for training, 4 months for validation, and 4 months for testing.

ETTh2 (Zhou et al., 2021). ETTh2 is an industrial time series benchmark from the ETT dataset family released by Tsinghua University. It contains hourly observations from 2016 to 2018, with 17,420 samples and seven variables, including six power-load features and oil temperature as the target. Compared with ETTh1, ETTh2 is collected from a transformer station in a different region, where load fluctuations and temperature regulation patterns exhibit distinct regional characteristics.

ETTm1 (Zhou et al., 2021). ETTm1 is a minute-level variant of the ETT benchmark for time series tasks. It contains two years of transformer operation records from July 2016 to July 2018, collected at 15-minute intervals. Each sample includes a timestamp, six power-load covariates, and oil temperature as the target. The load-related variables are HUFL, HULL, MUFL, MULL, LUFL, and LULL, corresponding to high-, middle-, and low-usage load measurements under different operating states. Compared with the hourly ETT datasets, ETTm1 provides finer-grained temporal observations, making it suitable for evaluating models' ability to capture short-term fluctuations.

ETTm2 (Zhou et al., 2021). ETTm2 is minute-level benchmark from the ETT dataset family, designed for fine-grained electric transformer temperature forecasting. It contains two years of records from July 2016 to July 2018, collected at 15-minute intervals. Each record consists of a timestamp, six power-load measurements, and oil temperature as the prediction target. The six covariates, including HUFL, HULL, MUFL, MULL, LUFL, and LULL, describe high-, middle-, and low-usage load behaviors under different operating states.

Weather (Wu et al., 2023). Weather is a used benchmark dataset, containing 21 meteorological variables such as air temperature, humidity, wind speed, precipitation, and atmospheric pressure. The dataset records weather observations at 10-minute intervals over one year, providing fine-grained temporal dynamics for environmental modeling. Due to its diverse weather indicators and high sampling frequency, Weather is widely used to evaluate forecasting models' ability to capture short-term fluctuations, periodic patterns, and complex dependencies among meteorological variables.

Electricity (Li et al., 2019). Electricity is a public time series benchmark that records electricity consumption and related power measurements across customers or households. Depending on the setting, it contains either hourly consumption records from 2012 to 2014 for 321 customers, or minute-level household power records from 2006 to 2010. The dataset includes temporal fields such as date and time, together with variables including global active power, global reactive power, voltage, global intensity, and sub-metering values for different appliance groups.

C.2 Datasets for Classification Task

JapaneseVowels (Kudo et al., 1999). Japanese Vowels is a multivariate time series classification dataset constructed from speech recordings of nine male speakers pronouncing the Japanese vowel sequence /ae/. It contains 640 time series instances, with 270 samples for training and 370 samples for testing. Each time step is represented by 12 LPC cepstral coefficients, and the sequence length varies from 7 to 29. The recordings were sampled at 10 kHz, with a frame length of 25.6 ms and a frame shift of 6.4 ms. This dataset is commonly used to evaluate models on speaker or speech-pattern classification from short multivariate temporal signals.

Handwriting (Shokoohi-Yekta et al., 2017). Handwriting is a multivariate time series classification dataset collected from smartwatch motion signals while subjects write the 26 letters of the alphabet. It contains 1,000 instances, including 150 training samples and 850 test samples. Each sequence consists of three dimensions corresponding to the three-axis accelerometer readings from the smartwatch. The original sequences were padded to a consistent format, making the dataset suitable for evaluating models on gesture and motion-pattern classification from wearable sensor data.

Heartbeat (Bagnall et al., 2018). Heartbeat is a multivariate time series dataset derived from the PhysioNet/CinC Challenge 2016 heart sound recordings. The data were collected from multiple contributors worldwide in both clinical and non-clinical environments, covering healthy subjects and patients with confirmed cardiac diagnoses. Each recording is labeled as either normal or abnormal, where abnormal cases are mainly associated with cardiac diseases such as valve defects and coronary artery disease. The recordings were truncated to 5 seconds and converted into spectrograms, where each dimension corresponds to a frequency band. The dataset contains two classes, with 113 normal instances and 296 abnormal instances, making it suitable for biomedical signal classification.

EthanolConcentration (Large et al., 2018). Ethanol Concentration is a multivariate time series classification dataset based on raw spectra of water–ethanol solutions measured from 44 real whisky bottles. The samples correspond to four ethanol concentration levels: 35%, 38%, 40%, and 45%. Since Scotch whisky is legally required to contain at least 40% alcohol, the task is to classify the ethanol concentration of an unknown sample from its spectral signal. This dataset is commonly used to evaluate models on fine-grained classification of chemical or spectral time series data.

PEMS-SF (Cuturi, 2011). PEMS-SF is a multivariate time series classification dataset derived from traffic sensor data provided by the California Department of Transportation. It contains 15 months of daily freeway lane-occupancy measurements from the San Francisco Bay Area, covering January 1, 2008 to March 30, 2009, sampled every 10 minutes. Each instance represents one day, with 963 sensor dimensions and a sequence length of 144. After removing public holidays and anomalous days, the dataset contains 440 daily time series. The classification task is to identify the corresponding day of the week, from Monday to Sunday, based on the observed traffic occupancy patterns.

C.3 Datasets for Anomaly Detection Task

SMD (Su et al., 2019). SMD (Server Machine Dataset) is a public multivariate time series anomaly detection dataset released by the NetMan Lab at Tsinghua University for intelligent system monitoring. It contains five weeks of operation records from 28 server machines, where each machine is represented by 38 monitored variables. The data are collected at one-minute intervals, while the exact timestamp information is anonymized. Due to its machine-level monitoring signals and labeled abnormal segments, SMD is commonly used to evaluate anomaly detection methods under realistic server operation scenarios.

MSL (Hundman et al., 2018). MSL (Mars Science Laboratory) is a real-world time series anomaly detection dataset provided by NASA, derived from telemetry signals of the Curiosity rover. It consists of 27 sensor channels, each represented with 55 feature dimensions. The dataset is split into training and testing sets, where the training data contain only normal patterns and the test data include expert-labeled anomalies. Due to its spacecraft telemetry signals and realistic abnormal events, MSL is commonly used to evaluate anomaly detection methods under complex system monitoring scenarios.

SMAP (Hundman et al., 2018). SMAP (Soil Moisture Active Passive) is a real-world time series anomaly detection dataset provided by NASA, derived from spacecraft telemetry signals. It con-

tains 55 telemetry channels with a total of 429,735 data points, including 69 expert-labeled anomalies. Similar to MSL, SMAP provides realistic system-monitoring signals with annotated abnormal events, making it commonly used for evaluating anomaly detection methods on multivariate spacecraft telemetry data.

SWaT (Mathur and Tippenhauer, 2016). SWaT (Secure Water Treatment) is a real-world industrial control system dataset collected from a scaled water treatment testbed. It records the operation of a water treatment process under both normal conditions and cyber-attack scenarios. The dataset contains multivariate sensor and actuator signals from different stages of the treatment pipeline, reflecting physical process dynamics and control-system behaviors. Due to its realistic industrial setting and attack annotations, SWaT is commonly used for anomaly detection and security analysis in industrial control systems.

PSM (Abdulaal et al., 2021). PSM (Pooled Server Metric Dataset) is a public time series anomaly detection dataset collected by eBay from multiple application server nodes. It contains 13 weeks of training data and 8 weeks of test data, with 132,481 records in total. Each record includes a timestamp and 24 server-performance metrics, such as CPU usage, memory usage, and other system-level indicators. With real server monitoring signals and anomaly annotations, PSM is commonly used for evaluating anomaly detection methods in large-scale service operation scenarios.

Traffic (Wu et al., 2023). Traffic is a public time series benchmark that records hourly road occupancy rates from freeway sensors in the San Francisco Bay Area between 2015 and 2016. The dataset captures traffic dynamics across multiple sensors, reflecting temporal variations in road usage, congestion patterns, and daily or weekly mobility behaviors. It is commonly used for analyzing large-scale transportation time series and evaluating models on traffic-related temporal patterns.

Solar (Lai et al., 2018). Solar is a public renewable-energy time series benchmark containing solar power production records from 137 photovoltaic plants in Alabama State. The data were collected in 2006 at 10-minute intervals, capturing fine-grained variations in solar generation across different PV plants. Due to its strong daily periodicity, weather-related fluctuations, and multi-site generation patterns, the dataset is commonly used for analyzing temporal dynamics in solar energy production.

PEMS04 (Liu et al., 2022a). PEMS is a collection of public traffic-network time series datasets, including PEMS03, PEMS04, PEMS07, and PEMS08. These datasets record traffic measurements from road sensor networks and contain complex spatial-temporal dependencies across different locations. Due to their large-scale sensor structure and dynamic traffic patterns, PEMS datasets are widely used for analyzing urban traffic flow, road-network dynamics, and spatial-temporal modeling.

PEMS08 (Liu et al., 2022a). PEMS is a collection of public traffic-network time series datasets, including PEMS03, PEMS04, PEMS07, and PEMS08. These datasets record traffic measurements from road sensor networks and contain complex spatial-temporal dependencies across different locations. Due to their large-scale sensor structure and dynamic traffic patterns, PEMS datasets are widely used for analyzing urban traffic flow, road-network dynamics, and spatial-temporal modeling.

Exchange (Wu et al., 2023). Exchange-Rate is a public financial time series benchmark containing daily exchange rates from 1990 to 2016. It covers eight economies, including Australia, Britain, Canada, Switzerland, China, Japan, New Zealand, and Singapore. The dataset records long-term currency dynamics and cross-country exchange-rate fluctuations, making it suitable for analyzing financial temporal patterns, trend changes, and interactions among multiple foreign exchange series.

C.4 Datasets for Reasoning Task

RCW (Liu et al., 2025b). RCW (North Atlantic Right Whale Calls) is an audio-based time series classification dataset used for detecting right whale calls in oceanic environmental noise. Each sample corresponds to an acoustic signal segment, and the task is to determine whether it contains a North Atlantic right whale call. The key discriminative cue is a short rising "whoop" sound, which provides clear signal-level evidence for the positive class.

TEE (Liu et al., 2025b). TEE (Transient Electromagnetic Event) is a time series classification dataset derived from FORTE satellite observations. The satellite uses optical and radio-frequency instruments to detect transient electromagnetic events

associated with lightning activity. Each class corresponds to a specific physical event type with distinguishable temporal characteristics in the measured signal. For example, the "CG Positive" category is related to positive cloud-to-ground discharge and is typically reflected as a sharp radiation spike followed by subsequent signal variations in the power-density time series.

ECG (Liu et al., 2025b). ECG is a single-lead electrocardiogram classification dataset covering four cardiac rhythm categories: normal sinus rhythm, atrial fibrillation, other alternative rhythms, and noisy or unclassifiable recordings. Unlike simple pattern-based tasks, each category depends on multiple clinical signal features. For example, atrial fibrillation is typically identified by combining cues such as irregular rhythm, missing P waves, abnormal baseline patterns, QRS characteristics, and fibrillatory waves.

EMG (Liu et al., 2025b). EMG is a single-channel electromyography classification dataset based on muscle responses to neural stimulation. It includes three categories: healthy, neuropathy, and myopathy. The task requires integrating multiple signal characteristics rather than relying on a single cue. For example, neuropathy is typically associated with fibrillations, positive sharp waves, high-amplitude and long-duration motor unit action potentials, and polyphasic waveforms.

HAR (Liu et al., 2025b). HAR is a human activity recognition dataset collected from smartphone wearable sensors. It records six daily activities, including walking, walking upstairs, walking downstairs, sitting, standing, and lying down. We use three sensor channels corresponding to body linear acceleration. Since the data were collected from 30 users with different motion habits, the same activity may exhibit user-specific variations, making HAR suitable for evaluating probabilistic reasoning over behavioral time series.

CTU (Liu et al., 2025b). CTU is a computer type usage dataset based on 24-hour electricity consumption patterns. The task is to infer whether the device is a desktop or a laptop from consumer electricity usage behavior. The dataset was collected from 251 users, whose individual usage habits introduce unobserved variability into the signals.

D Dataset Statistics

Imputation. For the imputation task, we evaluate the models on six widely used time series benchmarks, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, and Electricity, whose statistics are reported in Tab. 5. Following the standard protocol of TimesNet, we construct incomplete inputs by randomly masking $\{12.5\%, 25\%, 50\%\}$ of time points from time series segments with length 96. The models are required to reconstruct the missing values based on the remaining observed context. We compare each original model with its TokenDecouple-enhanced variant under all missing ratios, and report MSE and MAE as metrics.

Table 5: Statistics of datasets used for imputation.

Dataset	Length	Dimension	Frequency
ETTh1	17420	7	1 hour
ETTh2	17420	7	1 hour
ETTm1	69680	7	15 min
ETTm2	69680	7	15 min
Weather	52696	22	10 min
Electricity	26304	321	1 hour

Classification. For the classification task, we evaluate the models on five multivariate time series classification benchmarks, including JapaneseVowels, Handwriting, Heartbeat, EthanolConcentration, and PEMS-SF. Their statistics are summarized in Tab. 6. These datasets cover diverse domains, including speech signals, wearable motion signals, biomedical recordings, chemical spectra, and traffic sensor measurements. We use Accuracy as the evaluation metric and compare each baseline model with its TokenDecouple-enhanced counterpart.

Table 6: Statistics of datasets used for classification.

Dataset	Train Cases	Test Cases	Dimensions	Length	Classes
JapaneseVowels	270	370	12	29	9
Handwriting	150	850	3	152	26
Heartbeat	204	205	61	405	2
EthanolConcentration	261	263	3	1751	4
PEMS-SF	267	173	963	144	7

Anomaly Detection. For the anomaly detection task, we evaluate the models on three widely used multivariate time series anomaly detection benchmarks, including SMAP, MSL, and SMD. Their statistics are summarized in Tab. 7. These datasets cover spacecraft telemetry and server-machine monitoring scenarios, where anomalies are identified from multivariate temporal signals. We use

F1-score as the evaluation metric and compare each baseline model with its TokenDecouple-enhanced counterpart on all datasets.

Table 7: Statistics of datasets used for anomaly detection.

Dataset	Entities	Dimensions	Training Size	Testing Size	Anomaly Ratio (%)
SMAP	55	25	135,183	427,617	13.13
MSL	27	55	58,317	73,729	10.72
SMD	28	38	708,405	708,420	4.16

Forecasting. For the forecasting task, we evaluate the models on five widely used multivariate time series benchmarks, including Traffic, Solar, PEMS04, PEMS08, and Exchange. Their statistics are summarized in Tab. 8. In the table, *Dim* denotes the number of variables, *Size* denotes the number of time points in the train, validation, and test splits, respectively, and *Frequency* denotes the sampling interval. We adopt two prediction horizons $H \in \{192, 720\}$ and report MSE and MAE as evaluation metrics.

Table 8: Statistics of datasets used for forecasting.

Dataset	Dim	Size (Train, Val, Test)	Frequency
Traffic	862	(12185, 1757, 3509)	Hourly
Solar	137	(36601, 5161, 10417)	10 min
PEMS04	307	(10172, 3375, 3375)	5 min
PEMS08	170	(10690, 3548, 3548)	5 min
Exchange	8	(5120, 665, 1422)	Daily

Reasoning-centric Tasks. We evaluate reasoning tasks across three levels of complexity, including Simple Deterministic Reasoning, Complex Deterministic Reasoning, and Probabilistic Reasoning. The dataset statistics are summarized in Table 9. Simple Deterministic Reasoning includes RCW and TEE, where class labels correspond to clearly defined temporal patterns annotated by human experts. Complex Deterministic Reasoning includes ECG record diagnosis and EMG signal diagnosis, where labels require the joint interpretation of multiple temporal patterns by medical experts. Probabilistic Reasoning includes HAR and CTU, where class boundaries are less deterministic due to uncertain feature–label relationships and unobserved user-specific behaviors.

Statistical Characteristics of Datasets. We characterize each dataset from seven statistical perspectives, including **shifting**, **stationarity**, **transition**, **seasonality**, **trend**, **correlation**, and **non-Gaussianity**. Shifting measures the degree of distributional variation across time, while stationarity reflects whether the statistical properties of the sequence remain stable. Transition captures abrupt changes or regime switches in temporal dynamics. Seasonality and trend describe periodic structures and long-term directional movements, respectively. Correlation reflects the dependency among variables, and non-Gaussianity measures the deviation of the data distribution from a Gaussian assumption. Together, these attributes provide a comprehensive description of the temporal complexity and distributional diversity of different datasets.

Table 9: Statistics of datasets used for reasoning tasks.

Dataset	Variables	Length	Classes	Samples
RCW	1	4000	2	30,000
TEE	1	319	7	143
ECG	1	1500	4	43,673
EMG	1	1500	3	205
CTU	1	720	2	500
HAR	3	206	6	10,299

The datasets exhibit diverse statistical properties. For example, ETT datasets show relatively strong trend patterns, Electricity and Solar contain clear seasonality, Traffic and PEMS08 exhibit strong inter-variable correlations, while Exchange presents more evident stationarity variation and transition behavior. This diversity indicates that the evaluation covers a broad range of temporal regimes, rather than being restricted to datasets with similar dynamics. Therefore, the experimental results reflect model behavior under heterogeneous temporal structures and distributional conditions.

The broad coverage of dataset characteristics also helps assess the robustness of our method. Despite substantial differences in shifting, seasonality, trend, correlation, and distributional shape, our method maintains stable performance gains across datasets. This suggests that the proposed token decoupling strategy is not overly dependent on a specific statistical pattern or dataset type. Instead, by separately compressing TS tokens and prompt tokens according to their distinct functional roles, the method remains effective under diverse temporal dynamics and is less sensitive to dataset-specific statistical variations.

E Theoretical Analysis of TS Token Merging

This section provides a stability analysis for the proposed frequency-domain TS token merging strategy. The goal is not to prove that token merging is lossless or always improves task performance. Instead, we show that the perturbation introduced by TS token merging is explicitly bounded under mild assumptions. Specifically, we prove that: (i) frequency-domain perturbation is equivalent to time-domain token perturbation under an orthonormal DFT; (ii) active spectral token selection controls the residual energy of inactive tokens; (iii) local affinity-based merging controls the group-wise merging error; and (iv) under the Lipschitz stability of the LLM backbone and task head, the induced output perturbation is also bounded. This analysis explains why frequency-domain TS token merging can reduce token redundancy while preserving dominant temporal information.

Setup. Let $X_b \in \mathbb{R}^{N\times D}$ denote the TS token sequence of the b-th sample, where N is the number of TS tokens and D is the token dimension. We apply an orthonormal DFT along the token dimension and obtain $F_b = \mathcal{F}X_b \in \mathbb{C}^{N\times D}$, where $\mathcal{F}$ is a unitary DFT matrix satisfying $\mathcal{F}^*\mathcal{F} = I$. For frequency component k, let $F_{b,i,k} \in \mathbb{C}^D$ denote the spectral vector of token i and define its spectral energy as $\|F_{b,i,k}\|_2^2$.

For each frequency k, the active token set $\mathcal{I}_{b,k}$ is selected to preserve at least a ρ fraction of the spectral energy, i.e., $\sum_{i\in\mathcal{I}_{b,k}} \|F_{b,i,k}\|_2^2 \geq \rho \sum_{i=1}^{N} \|F_{b,i,k}\|_2^2$. The active set is partitioned into local merging groups $\{\mathcal{G}_{b,k}^{(u)}\}_{u=1}^{m_{b,k}}$. For each group, the merged representative is defined as $\tilde{F}_{b,u,k} = \sum_{i\in\mathcal{G}_{b,k}^{(u)}} \omega_{b,i}^{(u,k)} F_{b,i,k}$, where $\omega_{b,i}^{(u,k)} \geq 0$ and $\sum_{i\in\mathcal{G}_{b,k}^{(u)}} \omega_{b,i}^{(u,k)} = 1$. Thus, the merged token is a convex combination of tokens within the same local spectral group.

Since the compressed representation has fewer tokens than the original one, we introduce an expanded merged representation only for theoretical comparison. For every $i \in \mathcal{G}_{b,k}^{(u)}$, we set $\bar{F}_{b,i,k} = \tilde{F}_{b,u,k}$. The resulting $\bar{F}_b$ has the same index structure as F_b, and $\bar{X}_b = \mathcal{F}^{-1}\bar{F}_b$ denotes its inverse-DFT counterpart. This expanded representation is only used in the proof and does not change the actual compressed inference process.

Lemma 1 (Frequency-domain perturbation equals time-domain perturbation). If $\mathcal{F}$ is an orthonormal DFT matrix, then the token-level perturbation in the time domain equals the perturbation in the frequency domain:

$$\|X_b - \bar{X}_b\|_F = \|F_b - \bar{F}_b\|_F.$$

Proof. Since $F_b = \mathcal{F}X_b$ and $\bar{F}_b = \mathcal{F}\bar{X}_b$, we have $F_b - \bar{F}_b = \mathcal{F}(X_b - \bar{X}_b)$. Because $\mathcal{F}^*\mathcal{F} = I$, the Frobenius norm is preserved under the orthonormal DFT. Therefore, $\|F_b - \bar{F}_b\|_F = \|X_b - \bar{X}_b\|_F$. This shows that controlling the merging error in the frequency domain is equivalent to controlling the perturbation in the original TS token space.

Lemma 2 (Inactive spectral residual is bounded). For each frequency component k, if the active token set $\mathcal{I}_{b,k}$ preserves at least a ρ fraction of the spectral energy, then the residual energy of inactive tokens is bounded by

$$\sum_{k=0}^{K-1} \sum_{i\notin\mathcal{I}_{b,k}} \|F_{b,i,k}\|_2^2 \leq (1-\rho)\|F_b\|_F^2.$$

Proof. By the definition of $\mathcal{I}_{b,k}$, we have $\sum_{i\in\mathcal{I}_{b,k}} \|F_{b,i,k}\|_2^2 \geq \rho \sum_{i=1}^{N} \|F_{b,i,k}\|_2^2$. Hence, the inactive part at frequency k satisfies $\sum_{i\notin\mathcal{I}_{b,k}} \|F_{b,i,k}\|_2^2 \leq (1-\rho) \sum_{i=1}^{N} \|F_{b,i,k}\|_2^2$. Summing this inequality over all frequencies gives the stated bound. This result shows that active token selection directly controls the spectral information discarded by merging.

Lemma 3 (Local spectral merging error is bounded). For each local group $\mathcal{G}_{b,k}^{(u)}$, define its spectral diameter as $\delta_{b,k}^{(u)} = \max_{i,j\in\mathcal{G}_{b,k}^{(u)}} \|F_{b,i,k} - F_{b,j,k}\|_2$. Then replacing all tokens in this group by their merged representative introduces the following bounded error:

$$\sum_{i\in\mathcal{G}_{b,k}^{(u)}} \|F_{b,i,k} - \bar{F}_{b,i,k}\|_2^2 \leq |\mathcal{G}_{b,k}^{(u)}|(\delta_{b,k}^{(u)})^2.$$

Proof. For any $i \in \mathcal{G}_{b,k}^{(u)}$, since $\bar{F}_{b,i,k} = \tilde{F}_{b,u,k}$ and $\tilde{F}_{b,u,k}$ is a convex combination of tokens in the same group, we have

$$\|F_{b,i,k} - \bar{F}_{b,i,k}\|_2 = \left\| F_{b,i,k} - \sum_{j\in\mathcal{G}_{b,k}^{(u)}} \omega_{b,j}^{(u,k)} F_{b,j,k} \right\|_2 \leq \sum_{j\in\mathcal{G}_{b,k}^{(u)}} \omega_{b,j}^{(u,k)} \|F_{b,i,k} - F_{b,j,k}\|_2 \leq \delta_{b,k}^{(u)}.$$

Squaring the inequality and summing over all tokens in $\mathcal{G}_{b,k}^{(u)}$ gives the result. Therefore, the local merging error is controlled by the spectral diameter of the group, which justifies restricting merging to locally similar TS tokens.

Theorem 1 (Frequency-domain TS token merging induces bounded output perturbation). Assume that the LLM backbone f_Θ is L_f-Lipschitz continuous and the task-specific head h is L_h-Lipschitz continuous under the Frobenius norm. Then the final output perturbation induced by frequency-domain token merging is bounded by:

$$\|h(f_\Theta(X_b)) - h(f_\Theta(\bar{X}_b))\|_F \leq L_h L_f \left[\sum_{k=0}^{K-1}\sum_{u=1}^{m_{b,k}} |\mathcal{G}_{b,k}^{(u)}|(\delta_{b,k}^{(u)})^2 + (1-\rho)\|F_b\|_F^2\right]^{1/2}.$$

Proof. We first decompose the frequency-domain perturbation into the active merging error and the inactive residual. By Lemma 3, the active merging error over all groups and frequencies is bounded by $\sum_{k=0}^{K-1}\sum_{u=1}^{m_{b,k}} |\mathcal{G}_{b,k}^{(u)}|(\delta_{b,k}^{(u)})^2$. By Lemma 2, the inactive spectral residual is bounded by $(1-\rho)\|F_b\|_F^2$. Therefore,

$$\|F_b - \bar{F}_b\|_F^2 \leq \sum_{k=0}^{K-1}\sum_{u=1}^{m_{b,k}} |\mathcal{G}_{b,k}^{(u)}|(\delta_{b,k}^{(u)})^2 + (1-\rho)\|F_b\|_F^2.$$

Using Lemma 1, we further have $\|X_b - \bar{X}_b\|_F = \|F_b - \bar{F}_b\|_F$. Since f_Θ and h are Lipschitz continuous, the output perturbation satisfies:

$$\begin{aligned}\|h(f_\Theta(X_b)) - h(f_\Theta(\bar{X}_b))\|_F &\leq L_h\|f_\Theta(X_b) - f_\Theta(\bar{X}_b)\|_F \\ &\leq L_h L_f\|X_b - \bar{X}_b\|_F \\ &= L_h L_f\|F_b - \bar{F}_b\|_F.\end{aligned}$$

Substituting the frequency-domain perturbation bound into the above inequality gives the theorem.

Discussion. Theorem 1 shows that the effect of TS token merging is governed by two interpretable terms. The first term, $\sum_{k,u} |\mathcal{G}_{b,k}^{(u)}|(\delta_{b,k}^{(u)})^2$, is the local merging error. It becomes small when tokens within each group have similar spectral structures, which supports the use of the local affinity graph. The second term, $(1-\rho)\|F_b\|_F^2$, is the inactive spectral residual. It is controlled by the retained spectral energy ratio ρ, which explains the role of active spectral token selection. Therefore, the proposed TS token merging strategy is not arbitrary compression: it preserves dominant spectral components and merges only locally similar tokens, leading to a bounded perturbation on the output.

This theoretical result directly supports the design of the TS token merging module in TokenDecouple. The DFT step provides a space where spectral redundancy can be measured; active token selection preserves dominant frequency information; local affinity-based grouping controls the merging error; and the adaptive merging budget adjusts the compression strength across frequencies. Together, these components reduce the number of TS tokens while keeping the induced representation and output perturbations bounded.

F Computational Cost of Prompt Tokens

We consider an L-layer LLM with an initial number of prompt tokens denoted as N_{Text}. To reduce the computational overhead introduced by prompt tokens, we adopt a pyramidal decay strategy across layers, where the number of prompt tokens decreases geometrically with model depth.

Specifically, we assume that the prompt token count is reduced by a constant ratio of 0.25 at each successive layer. As a result, the number of prompt tokens processed at layer l is given by:

$$N_{\text{Text}}^{(l)} = 0.25^{\,l} N_{\text{Text}},$$

which produces a rapidly decaying sequence of approximately 100%, 25%, 6.3%, 1.6%, and 0.4% of the original prompt length across early layers.

The total number of prompt tokens processed throughout the entire network is obtained by summing the contributions from all layers:

$$\sum_{l=1}^{L} N_{\text{Text}}^{(l)} = \sum_{l=1}^{L} 0.25^{\,l} N_{\text{Text}}.$$

This summation forms a finite geometric series and can be evaluated in closed form as:

$$\sum_{l=1}^{L} N_{\text{Text}}^{(l)} = N_{\text{Text}}\frac{1-0.25^L}{1-0.25}.$$

To characterize the effective prompt length experienced by each layer on average, we divide the total prompt token count by the number of layers L, yielding:

$$\bar{N}_{\text{Text}} = \frac{1}{L}\sum_{l=1}^{L} N_{\text{Text}}^{(l)} = \frac{N_{\text{Text}}}{L(1-0.25)}(1-0.25^L).$$

For moderately deep language models (i.e., $L \geq 4$), the term 0.25^L rapidly approaches zero and can be safely neglected. Under this approximation, the average number of prompt tokens per layer simplifies to:

$$\bar{N}_{\text{Text}} \approx \frac{4}{3L} N_{\text{Text}}.$$

We then define the prompt token compression ratio as:

$$\eta_L = 1 - \frac{\bar{N}_{\text{Text}}}{N_{\text{Text}}},$$

which leads to:

$$\eta_L \approx 1 - \frac{4}{3L}.$$

This analysis shows that prompt-induced computational overhead decreases sharply with model depth. For example, a 12-layer LLM achieves a compression ratio of approximately $\eta_{12} \approx 89\%$, while a 24-layer model reaches $\eta_{24} \approx 94\%$, demonstrating that deeper models naturally amortize prompt costs under pyramidal token decay.

G Layer-wise Prompt Contribution Analysis

To understand the layer-wise behavior of prompt token compression, we analyze the actual contribution of prompt tokens to TS tokens. Motivated by this, we first examine the interaction between TS tokens and prompt tokens in Transformer attention. Let A_l^h denote the attention matrix of the h-th attention head at layer l. Let $\mathcal{T}$ and $\mathcal{P}$ denote the index sets of TS tokens and prompt tokens, respectively. For any TS token $i \in \mathcal{T}$, its total attention assigned to all prompt tokens is defined as:

$$C_{l,h}(i) = \sum_{j \in \mathcal{P}} A_l^h(i,j).$$

This quantity measures the proportion of information absorbed by TS token i from prompt tokens when updating its representation in the h-th head of layer l. A larger $C_{l,h}(i)$ indicates that prompt tokens still exert a strong influence on TS representation learning at this layer, whereas a smaller value suggests that TS tokens rely less on prompt tokens. To characterize the layer-wise trend, we define the Prompt Contribution Ratio as:

$$\text{PCR}_l = \frac{1}{H|\mathcal{T}|} \sum_{h=1}^{H} \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{P}} A_l^h(i,j),$$

where H is the number of attention heads and $|\mathcal{T}|$ is the number of TS tokens. PCR_l measures the average attention mass assigned by TS tokens to prompt tokens at layer l. A higher PCR_l indicates a stronger contribution of prompt tokens to TS representation updates, while a lower value implies a weaker marginal effect of prompt tokens.

However, PCR_l alone is insufficient to determine the compression ratio, since it only captures the contribution strength of prompt tokens but not the token reduction induced by the compression strategy. Therefore, we further introduce the Prompt Retention Ratio to describe how many prompt tokens are preserved at each layer under different compression ratios. Let P_0 denote the number of prompt tokens at the input layer and P_l denote the number of prompt tokens retained at layer l. The retention ratio is defined as:

$$R_l(\alpha) = \frac{P_l}{P_0}.$$

When a fixed hierarchical compression ratio α is used, the number of prompt tokens is updated by

$$P_{l+1} = \lceil \alpha P_l \rceil.$$

Ignoring the rounding effect, we have

$$R_l(\alpha) \approx \alpha^l.$$

We further analyze the layer-wise behavior of prompt token compression over the first 12 layers. Specifically, Fig. 7 reports the Layer-wise Prompt Contribution Ratio, which measures the average attention mass assigned by TS tokens to prompt tokens at each layer. The results show that prompt contribution is relatively high in the shallow layers but decreases rapidly as the model depth increases. This indicates that prompt tokens mainly provide semantic guidance in the early stages, where TS tokens absorb task instructions, contextual information, and answer-related constraints. After several shallow layers, the direct contribution of prompt tokens becomes marginal, suggesting that keeping full prompt tokens in deeper layers introduces substantial redundant computation.

To further relate this observation to the compression strategy, Fig. 8 shows the Prompt Retention Ratio under different hierarchical compression ratios. The retention curves indicate that larger ratios such as $\alpha = 0.5$ and $\alpha = 0.75$ preserve many prompt tokens even in deeper layers, which is inconsistent with the rapidly decaying prompt contribution observed in Fig. 7. In contrast, an overly

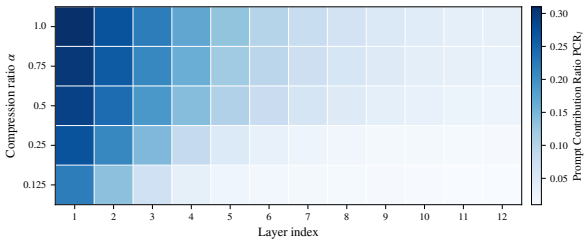


Figure 7: Layer-wise Prompt Contribution Ratio over the first 12 layers under different prompt compression ratios. Higher values indicate stronger attention from TS tokens to prompt tokens, showing that prompt contribution is mainly concentrated in shallow layers and rapidly decays with depth.

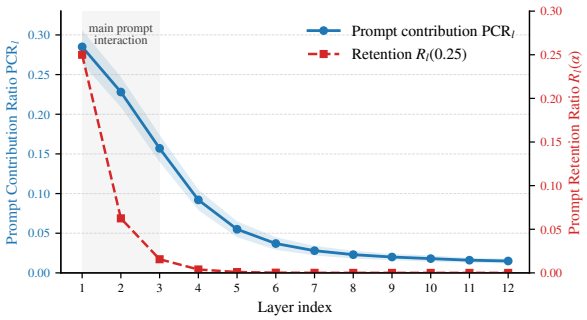


Figure 8: Layer-wise comparison between Prompt Contribution Ratio and Prompt Retention Ratio. The contribution of prompt tokens decreases rapidly across layers, while $\alpha = 0.25$ provides a matching retention decay pattern that preserves shallow-layer semantic guidance and reduces deep-layer prompt redundancy.

small ratio such as $\alpha = 0.125$ removes prompt tokens too aggressively and may reduce prompt information before it is sufficiently absorbed. The setting $\alpha = 0.25$ provides a more balanced decay pattern: it preserves prompt tokens in the shallow layers while quickly reducing them after the main contribution window. These results support the design of hierarchical prompt compression, showing that prompt tokens should be progressively compressed according to their layer-wise contribution rather than fully retained across all layers.

H Implementation Details

During training, we optimize all models using the AdamW optimizer (Loshchilov and Hutter, 2019) and adopt a cosine learning rate schedule for learning rate decay. To improve training stability, we apply a warmup strategy at the early training stage, with the warmup ratio set to 0.03 (Goyal et al., 2017). Mixed precision training is used to reduce GPU memory consumption and improve training efficiency. For each group of comparative experiments, we keep the batch size, number of training epochs, and optimizer settings consistent across all compared methods. The main hyperparameters include learning rate, weight decay, batch size, maximum number of epochs, gradient clipping threshold, and dropout ratio. For TokenDecouple, the compression regularization term $\mathcal{L}_{\text{pen}}$ is jointly optimized with the main task loss, where the coefficient λ controls the trade-off between compression strength and task performance.

During inference, TokenDecouple enables both TS token merging and prompt token compression. TS token merging reduces the number of time-series tokens at the input stage, thereby lowering the computational cost of subsequent self-attention operations. Prompt token compression progressively reduces prompt tokens in the shallow layers of the LLM, preventing unnecessary attention computation over redundant tokens in deeper layers.

All experiments are conducted on the same hardware platform. We use 2 NVIDIA A800 GPUs for both training and inference, and implement all models based on PyTorch (Paszke et al., 2019) and HuggingFace Transformers. For large-scale model training, we employ mixed precision and gradient accumulation to accommodate GPU memory constraints. All stochastic processes, including data splitting, parameter initialization, and training sampling, are controlled with fixed random seeds. To ensure the reliability of the results, the main experiments are repeated under multiple random seeds, and the average performance is reported.

I Hyperparameter Sensitivity Analysis

I.1 Effect of Regularization Weight

To examine the effect of the compression regularization weight λ, we conduct a sensitivity analysis on the reasoning tasks and report the average performance over six tasks. As shown in Fig. 9, λ controls the trade-off between compactness and task performance. When λ is too small, the compression constraint is weak and the model cannot fully exploit token redundancy. As λ increases to 5×10^{-3}, both OFA and TimeLLM achieve their best or near-best performance, indicating that a moderate compactness constraint can remove redundant TS tokens while preserving useful frequency information. However, when λ further increases to 10^{-2} or 5×10^{-2}, the performance drops noticeably, suggesting that overly strong compression may discard task-relevant temporal information. Therefore, we

set $\lambda = 5 \times 10^{-3}$ as the default value, which provides a favorable balance between efficiency and reasoning performance.

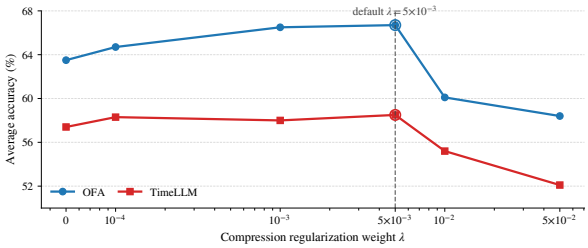


Figure 9: Sensitivity analysis of the compression regularization weight λ on reasoning tasks.

I.2 Effect of Local Affinity Radius

We further analyze the effect of the local affinity radius r_{loc}, which controls the neighborhood size used to construct the sparse local graph for TS token merging. As shown in Table 10, a small r_{loc} restricts the searchable merging candidates, making the compression overly conservative and limiting the benefit of redundancy removal. Increasing r_{loc} to a moderate value improves the average reasoning performance, indicating that a proper local neighborhood helps identify spectrally similar tokens and produces more reliable merged representatives. However, when r_{loc} becomes too large, the performance starts to decline. This is because a large neighborhood may connect tokens with different spectral structures, increasing the group-wise spectral diameter and introducing larger merging errors. Therefore, we set $r_{\text{loc}} = 4$ by default, which provides a stable balance between local spectral consistency and compression effectiveness.

Table 10: Sensitivity analysis of the local affinity radius r_{loc} on reasoning tasks.

r_{loc}	OFA	TimeLLM
1	64.1	57.6
2	65.8	58.1
4	**66.7**	**58.5**
8	65.9	57.9
16	62.8	55.6

J Prompt Templates

To ensure a consistent input format across different time-series tasks, we provide the prompt templates used in our experiments. Fig. 10–14 illustrate the templates for five tasks: forecasting, imputation, classification, anomaly detection, and reasoning. For pattern-centric tasks, including forecasting, imputation, classification, and anomaly detection, the prompts mainly specify the input sequence, task objective, and expected output format. For the reasoning task, the prompt further incorporates contextual information and candidate answers, requiring the model to jointly interpret temporal evidence and domain-specific semantics. These templates standardize the textual interface while allowing each task to preserve its own output structure.

Example of the Imputation task

Instruction: You are an expert in time series imputation. Given an incomplete time series, please infer the missing values based on the observed temporal evidence.
Input Time Series: [0.541, 0.682, 1.322, [MASK], ..., 1.087, 0.936, [MASK], 0.952]
Task Description: Given a time series with missing observations, recover the missing values by using the surrounding temporal context, local continuity, and global temporal patterns.
Missing Pattern: The missing values are marked as [MASK] in the input sequence.
Output: Please output the imputed values for all masked positions in their original order.

Figure 10: Prompt template for the Imputation task.

Example of the Classification task

Instruction: You are an expert in time series classification. Given an input time series, please assign it to the correct predefined class according to its temporal dynamics and discriminative patterns.
Input Time Series: [[0.312, -0.184, 0.527, ..., 0.091], [0.286, -0.207, 0.498, ..., 0.116], [0.241, -0.263, 0.462, ..., 0.154], ..., [0.105, -0.391, 0.336, ..., 0.218]]
Task Description: Given a multivariate time series instance, identify its class label from the predefined label set by analyzing temporal evolution, variable interactions, and class-specific dynamic patterns.
Candidate Classes:
(A) Class 1
(B) Class 2
(C) Class 3
(D) Class 4
(E) Class 5
Output: Please output the correct class label.

Figure 11: Prompt template for the Classification task.

Example of the Anomaly Detection task

Instruction: You are an expert in time series anomaly detection. Given an input time series, please identify whether each time step is normal or anomalous.
Input Time Series: [0.318, 0.336, 0.351, 0.329, ..., 0.374, 2.186, 0.392, 0.365]
Task Description: Given the input sequence, perform point-wise anomaly detection by assigning an anomaly label to each time step. A label of 0 indicates a normal point, while a label of 1 indicates an anomalous point. Anomalies may correspond to sudden spikes, abrupt drops, abnormal fluctuations, distribution shifts, or deviations from normal temporal patterns.
Label Definition:
0: Normal time point
1: Anomalous time point
Output: Please output a binary label sequence with the same length as the input time series, where each position corresponds to the anomaly label of the corresponding time step.

Figure 12: Prompt template for the Anomaly Detection task.

Example of the Forecasting task

Instruction: You are an expert in time series forecasting. Given a historical time series, please predict its future values based on the observed temporal evidence.
Input Time Series: Past 512 time steps: [0.214, 0.287, 0.356, 0.421, ..., 0.764, 0.812, 0.857, 0.903]
Task Description: Given the observed historical sequence of length 512, forecast the future sequence of length 192 by considering temporal continuity, trend, periodicity, and recent fluctuations.
Prediction Horizon: Please predict the next 192 time steps.
Output: Please output the predicted future time series values in chronological order: $[\hat{x}_{513}, \hat{x}_{514}, \ldots, \hat{x}_{704}]$.

Figure 13: Prompt template for the Forecasting task.

Example of the Reasoning task

Instruction: You are an expert in time series analysis and temporal reasoning. Given a time series and its contextual information, please answer the multiple-choice question based on the temporal evidence, domain context, and candidate answers. Only one answer is correct.
Input Time Series: [0.124, 0.139, 0.151, 0.148, ..., 0.632, 1.847, 0.914, 0.386]
Context: The sequence records a physical signal collected from a transient electromagnetic event. A sharp radiation spike followed by subsequent signal variations may indicate a positive cloud-to-ground discharge event.
Task Description: Given the temporal pattern and contextual information, infer the most plausible event type. The answer should be derived from both the observed time series dynamics and the provided domain knowledge, rather than from the numerical pattern alone.
Question: Which event type is most consistent with the observed temporal signal and the provided context?
Candidate Answers:
(A) CG Positive discharge
(B) Background noise without transient event
(C) Periodic sensor drift
(D) Long-duration stable radiation source
Output: Please output the correct candidate answer.

Figure 14: Prompt template for the Reasoning task.