LTSM-Bundle: A Toolbox and Benchmark on Large Language Models for Time Series Forecasting

Yu-Neng Chuang *
Rice University
yc146@rice.edu

Songchen Li *
Rice University
sl237@rice.edu

Jiayi Yuan *
Rice University
jy101@rice.edu

Guanchu Wang *
Rice University
gw22@rice.edu

Kwei-Herng Lai *
Rice University
khlai@rice.edu

Joshua Han
Rice University
jh273@rice.edu

Zihang Xu
Rice University
zx57@rice.edu

Songyuan Sui
Rice University
ss275@rice.edu

Leisheng Yu
Rice University
ly50@rice.edu

Sirui Ding
University of California, San Francisco
sirui.ding@ucsf.edu

Chia-Yuan Chang
Texas A&M University
cychang@tamu.edu

Alfredo Costilla Reyes
Rice University
acostillar@rice.edu

Daochen Zha
Rice University
Daochen.Zha@rice.edu

Xia Hu
Rice University
xia.hu@rice.edu

Abstract

Time Series Forecasting (TSF) has long been a challenge in time series analysis. Inspired by the success of Large Language Models (LLMs), researchers are now developing Large Time Series Models (LTSMs), universal transformer-based models that use autoregressive prediction to improve TSF. However, training LTSMs on heterogeneous time series data poses unique challenges, including diverse frequencies, dimensions, scalability, and patterns across datasets. Though recent efforts have studied and evaluated various design choices aimed at enhancing LTSM training and generalization capabilities, these design choices are typically studied and evaluated in isolation and are not compared collectively. In this work, we introduce `LTSM-Bundle`, a comprehensive toolbox and benchmark for training LTSMs, spanning pre-processing techniques, model configurations, and dataset configurations. Modularized and benchmarked LTSMs from multiple dimensions, encompassing prompting strategies, tokenization approaches, training paradigms, base model selection, data quantity, and dataset diversity. Our findings provide practical guidance for configuring effective LTSMs in real-world settings. The source code is available at `https://anonymous.4open.science/r/LTSM-bundle-5B70/`

1 Introduction

Time series forecasting (TSF) aims to predict future values from historical observations. Recent advances in deep learning, especially Transformers [25], and the emergence of Large Time Series Models (LTSMs) [26, 10, 8, 23, 7, 11, 3, 33, 14] promise strong performance across tasks and domains. In particular, Transformer-based approaches have shown benefits for long-term horizons [29, 31, 15].

*These authors contributed equally to this research.

NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT^{2}S).

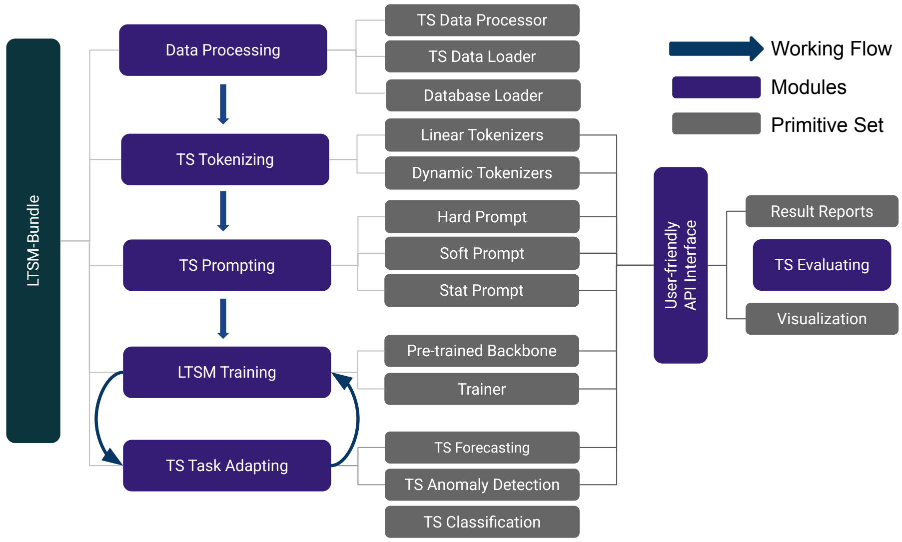

Figure 1: System Overview of LTSM-Bundle library. LTSM-Bundle provides an end-to-end training and evaluation pipeline constructed from data preprocessing to visualization.

However, unlike text, where tokens carry shared semantics, time series vary widely in frequency, dimensionality, and temporal patterns, making universal training and generalization difficult. Prior work has explored strategies in pre-processing (e.g., prompting [14] and tokenization [33, 1]), model configuration (e.g., reusing or adapting LLM backbones [33]), and dataset design [10, 33, 3], but these choices are typically evaluated in isolation. This fragmentation makes it hard to select components and understand their interactions.

To address the challenge, we introduce `LTSM-Bundle`, a modular toolbox and benchmark that unifies pre-processing, model, and dataset configurations into reproducible pipelines, covering prompting, tokenization, training paradigms, backbone selection, and data settings. We provide a consistent experimental protocol and systematically evaluate these components across eight datasets, identifying effective configurations and trade-offs. Our findings indicate that full fine-tuning generally converges faster and is more robust than training-from-scratch or LoRA in our setup; model size does not monotonically improve forecasting—smaller or medium backbones can match or exceed larger ones depending on horizon; and using roughly 5% of the training data often approaches full-data accuracy while reducing cost, with dataset diversity remaining important.

2 `LTSM-Bundle` Package Design

We present the system overview of `LTSM-Bundle` in Figure 1. `LTSM-Bundle` is a modular and extensible toolkit that supports the complete life cycle of large time series models (LTSMs), covering raw data ingestion, model training, evaluation, and deployment. The framework is organized into four tightly integrated subsystems, unified under a single API to eliminate boilerplate engineering and accelerate experimentation. First, **TS Tokenizing** converts multivariate time series into token sequences using both linear and dynamic schemes, preserving global trends and local temporal dynamics to make the data directly consumable by Transformer-style backbones. Second, **TS Prompting** embeds task instructions and statistical context through hard, soft, and statistics-aware prompts, enabling zero-shot and few-shot adaptation for forecasting, anomaly detection, and classification tasks. Third, the **Data Processing** layer provides scalable data loaders, windowing utilities, and feature-engineering pipelines that abstract dataset idiosyncrasies and handle large-scale benchmarks seamlessly. Finally, **LTSM Training** offers a unified optimization interface for fine-tuning or training from scratch across diverse backbones and parameter scales, with built-in support for curriculum learning, transfer learning, and large-scale hyperparameter sweeps. These subsystems are orchestrated by a reproducible workflow engine that integrates tokenizing, prompting, processing, and training into end-to-end pipelines. The toolkit further includes a comprehensive library of loss functions, data-augmentation routines, evaluation metrics, and visualization utilities, automatically generating publication-ready reports and artifacts. Moreover, `LTSM-Bundle` seamlessly integrates with the time series vector

Table 1: Performance of learning from scratch, LoRA fine-tuning, and fully fine-tuning

	Metric	MSE				MAE			
	Predict length	**96**	**192**	**336**	**720**	**96**	**192**	**336**	**720**
TS prompt	From scratch	0.325	0.296	0.323	0.355	0.355	0.375	0.374	0.409
	LoRA fine-tuning	0.343	0.381	0.399	0.466	0.374	0.403	0.426	0.478
	Fully fine-tuning	0.229	0.260	0.297	0.351	0.301	0.324	0.354	0.397
Text prompt	From scratch	0.494	0.434	0.597	0.485	0.463	0.438	0.512	0.475
	LoRA fine-tuning	0.347	0.379	0.406	0.473	0.373	0.404	0.431	0.484
	Fully fine-tuning	0.294	0.286	0.353	0.358	0.330	0.338	0.378	0.429

Table 2: Average performance of different sizes of LLM backbones

Metric	**MSE**				**MAE**			
	96	**192**	**336**	**720**	**96**	**192**	**336**	**720**
Small	0.252	0.306	0.316	0.352	0.313	0.363	0.367	0.400
Medium	0.229	0.260	0.297	0.351	0.301	0.324	0.354	0.397
Large	0.224	0.257	0.301	0.358	0.292	0.322	0.356	0.399
Extra Large	0.222	0.236	0.302	0.361	0.288	0.325	0.369	0.416

database TDengine, enabling a automated pipeline from raw data storage to result visualization and report. By reducing engineering overhead and simplifying experimentation, `LTSM-Bundle` lowers the barrier to reproducible research and accelerates industrial adoption.

3 Benchmarking LTSM Training

We benchmark existing LTSM components by involving and coordinating them into four fundamental components of `LTSM-Bundle` package. We aim to answer the following research questions: **(1) How do model sizes of LLMs impact time series forecasting performance?**; **2) How do training paradigms affect time series forecasting performance?**; **3) How does different training components benefit to time series forecasting performance?** and **4) How do different dataset configurations impact model generalization?** We follow the experimental settings outlined in Timesnet [28] and Time-LLM [14], employing the unified evaluation framework under 8 different datasets. The details of hyperparameter settings and datasets are in Appendix A and B, respectively.

3.1 Model Configuration: Training Paradigm

Different training paradigms exhibit unique characteristics that influence how well LLMs fit a specific training dataset. We explore three distinct training paradigms, fully fine-tuning, training from scratch, and LoRA [12], to identify the most effective approaches for training the LTSM framework.

Experimental Results. We assess the effectiveness of the training paradigm. Table 1 presents the results of various training strategies using GPT-2-Medium as the backbone. In general, the experimental results indicate that full fine-tuning is the most effective strategy for training the LTSM framework, whether leveraging time series prompts or text prompts. Based on the results, we summarize the observations as follows. 1 Although training-from-scratch achieves competitive performance compared to full fine-tuning, the large number of trainable model parameters may lead to overfitting, ultimately degrading performance. 2 Fully fine-tuning paradigm leads to the best performance with up to 11% improvement in MSE and up to 17% of improvement on MAE under the length of {96, 192, 336}, and performance competitive under the length of 720. Training `LTSM-bundle` under the full fine-tuning paradigm is recommended, as it converges twice as fast as training from scratch, ensuring efficient and effective forecasting.

3.2 Model Configuration: Model Size

Model Candidates To evaluate the impact of model size, we adopt four pre-trained LLM backbones: GPT-2-small, GPT-2-medium, GPT-2-large [22], and Phi-2 [13]. GPT-2 follows a transformer architecture with up to 48 layers and model sizes of 124M, 355M, and 774M parameters. Phi-2, also

Table 3: Performance of linear and time series tokenization

Metric	Tokenizer	ETTh1	ETTh2	ETTm1	ETTm2	Traffic	Weather	Exchange	Electricity	Avg.
MSE	Linear tokenizer	0.301	0.228	0.261	0.149	0.300	0.163	0.058	0.140	0.214
	Time series tokenizer	1.798	0.855	1.671	0.625	2.199	0.983	3.729	2.206	1.663
MAE	Linear tokenizer	0.372	0.319	0.346	0.265	0.268	0.230	0.173	0.241	0.281
	Time series tokenizer	1.057	0.606	0.991	0.488	1.083	0.619	1.495	1.108	0.895

Table 4: Average performance with different downsampling under different domain `LTSM-Bundle`

(MAE/MSE)	1 dataset	2 datasets	4 datasets	8 datasets
2.5%	0.446/0.450	**0.435/0.485**	0.396/0.357	0.352/0.293
5%	0.416/0.380	**0.415/0.436**	0.383/0.341	0.344/0.283
10%	0.414/0.375	**0.415/0.440**	0.394/0.355	0.348/0.288

transformer-based, contains 2.7B parameters and is trained on high-quality ("textbook-quality") data with improved scaling strategies. Unlike GPT-2's absolute positional encoding, Phi-2 employs relative positional encoding, capturing pairwise token distances for more robust position representations. Following [33], we utilize the top three self-attention layers from each pre-trained model as the backbone in the `LTSM-bundle` framework.

Experimental Results We investigate the effect of different pre-trained backbones on LTSM models for time series forecasting. Results are summarized in Table 2. Under a full fine-tuning paradigm, we observe the following: ③ GPT-2-small achieves up to **2% higher accuracy** than GPT-2-large on long-term forecasting tasks (336, 720). ④ GPT-2-medium outperforms GPT-2-large on short-term forecasting tasks (96, 192), suggesting that larger models are more prone to overfitting, which degrades performance. While Table 2 shows that parameter count within the same architecture has limited impact, we further compare Phi-2 with GPT-2 models of varying sizes (small, medium, large) under different prompting strategies. As detailed in Table 15 and Table 16 in Appendix H, GPT-2-small and GPT-2-medium consistently outperform Phi-2 across both time series prompts and textual instruction prompts.

3.3 Model Configuration: Tokenizations

In addition to leveraging instructional prompts to enhance generalization in LTSM training, we conduct a detailed analysis to identify the most effective tokenization strategy for LTSMs. We compare two distinct approaches—*linear tokenization* [33] and *time series tokenization* [1], to determine which method better supports LTSM training across complex and multi-domain datasets.

Experimental Results We evaluate the impact of the two tokenization strategies using pre-trained GPT-2-medium backbones and time series prompts, as reported in Table 3. Empirically, *linear tokenization* leads to more effective LTSM training than *time series tokenization*, especially when operating under limited-data regimes. This performance gap arises because time series tokenization relies on a pre-trained vocabulary derived from a specific LTSM architecture and dataset, which constrains its transferability across architectures and domains. In contrast, linear tokenization offers a more architecture-agnostic and adaptive representation, enabling better generalization under diverse LTSM configurations and low-resource settings. In summary, ⑤ *linear tokenization* emerges as a more effective and robust strategy for LTSM training, particularly when training data is scarce.

3.4 Dataset configuration: Quantity

The quantity of datasets often plays a key role in LLM performance. In this section, we investigate whether more training data consistently improves LTSMs. We apply down-sampling strategies to study the impact of the quantity of data on prediction performance. Each time series in the training set is periodically down-sampled along the timestamps to reduce granularity while maintaining general patterns. We include 2.5%, 5%, and 10% of the training data.

Experimental Results Table 4 reports the results across datasets. ⑥ We observe that increasing the amount of data does not always correlate with improved performance. Specifically, 5% of the data achieves a favorable balance between performance and computational cost: while 10% of the data slightly improves forecasting, it nearly doubles training time, and 2.5% loses too much information. Thus, optimal performance requires carefully balancing data quantity and diversity.

4 Conclusion

We propose `LTSM-Bundle`, a unified toolbox and benchmark for large time series models (LTSMs). Our benchmarking reveals four key insights: (1) **full fine-tuning** converges faster and is more robust than training from scratch or LoRA; (2) **forecasting accuracy does not scale monotonically**: smaller backbones can match or exceed larger ones depending on the forecasting horizon; (3) **linear tokenization** offers more reliable cross-architecture transfer than time series tokenization, especially under low-data regimes; and (4) **Data efficiency is achievable**: training with only $\sim$5% of the data can nearly match full-data performance when diversity is maintained.

References

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

[2] David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S Jensen. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2):1–27, 2023.

[3] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*, 2023.

[4] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12270–12280, 2021.

[5] Yu-Neng Chuang, Guanchu Wang, Chia-Yuan Chang, Ruixiang Tang, Shaochen Zhong, Fan Yang, Mengnan Du, Xuanting Cai, and Xia Hu. Faithlm: Towards faithful explanations for large language models. *arXiv preprint arXiv:2402.04678*, 2024.

[6] Yu-Neng Chuang, Leisheng Yu, Guanchu Wang, Lizhe Zhang, Zirui Liu, Xuanting Cai, Yang Sui, Vladimir Braverman, and Xia Hu. Confident or seek stronger: Exploring uncertainty-based on-device llm routing from benchmarking to generalization. *arXiv preprint arXiv:2502.04428*, 2025.

[7] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

[8] Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. Evaluating large language models on time series feature understanding: A comprehensive taxonomy and benchmark. *arXiv preprint arXiv:2404.16563*, 2024.

[10] Azul Garza and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

[11] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[13] Mojan Javaheripi and Sébastien Bubeck. Phi-2: The surprising power of small language models, December 2023.

[14] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[15] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

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

[17] Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Qingsong Wen, et al. Foundts: Comprehensive and unified benchmarking of foundation models for time series forecasting. *arXiv preprint arXiv:2410.11802*, 2024.

[18] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933, 2025.

[19] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

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

[21] Xiangfei Qiu, Xiuwen Li, Ruiyang Pang, Zhicheng Pan, Xingjian Wu, Liu Yang, Jilin Hu, Yang Shu, Xuesong Lu, Chengcheng Yang, et al. Easytime: Time series forecasting made easy. *arXiv preprint arXiv:2412.17603*, 2024.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[23] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *Preprint*, 2024.

[24] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

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

[26] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[27] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

[28] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.

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

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

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

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

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

Appendix

A Details of Datasets

In this paper, the training datasets include ETT (Electricity Transformer Temperature) [31][2], Traffic[3], Electricity[4], Weather[5], and Exchange-Rate[16]. ETT[6] [31] comprises four subsets: two with hourly-level data (ETTh) and two with 15-minute-level data (ETTm). Each subset includes seven features related to oil and load metrics of electricity transformers, covering the period from July 2016 to July 2018. The traffic dataset includes hourly road occupancy rates from sensors on San Francisco freeways, covering the period from 2015 to 2016. The electricity dataset contains hourly electricity consumption data for 321 clients, spanning from 2012 to 2014. The weather data set comprises 21 weather indicators, such as air temperature and humidity, recorded every 10 minutes throughout 2020 in Germany. Exchange-Rate[16] contains daily exchange rates for eight countries, spanning from 1990 to 2016. We first train our framework on the diverse time series data collection and then assess the abilities of `LTSM-Bundle` on jointly learning and zero-shot transfer learning to different domains of time series knowledge.

B Hyper-parameter Settings of Experiments

The hyper-parameter settings of `LTSM-Bundle` training for all experiments are shown in Table 5. Other training hyper-parameters follow the default values in the `TrainingArguments` class[7] of the huggingface transformers package.

Table 5: Hyperparameter settings of `LTSM-Bundle` training

Hyperparameter name	**Value**
Number of Transformer layers N	3
Training / evaluation / testing split	0.7 / 0.1 / 0.2
Gradient accumulation steps	64
Learning rate	0.001
Optimizer	Adam
LR scheduler	CosineAnnealingLR
Number of epochs	10
Number of time steps per token	16
Stride of time steps per token	8
Dimensions of TS prompt	133
Transformer architectures	GPT-2-{small, medium, large}, Phi-2
Length of prediction	96, 192, 336, 720
Length of input TS data	336
Data type	`torch.bfloat16`
Downsampling rate of training data	20

C Computation Infrastructure

All experiments described in this paper are conducted using a well-defined physical computing infrastructure, the specifics of which are outlined in Table 6. This infrastructure is essential for ensuring the reproducibility and reliability of our results, as it details the exact hardware environments used during the testing phases.

[2] `https://github.com/zhouhaoyi/ETDataset`
[3] `http://pems.dot.ca.gov`
[4] `https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014`
[5] `https://www.bgc-jena.mpg.de/wetter/`
[6] `https://github.com/laiguokun/multivariate-time-series-data`
[7] `https://github.com/huggingface/transformers/blob/main/src/transformers/training_args.py`

Table 6: Computing infrastructure for the experiments

Device attribute	Value
Computing infrastructure	GPU
GPU model	Nvidia A5000 / Nvidia A100
GPU number	8 × A5000 / 4 × A100
GPU memory	8 × 24GB / 4 × 80GB

D Comparison with Other LTSM Packages

In this section, we highlight the difference and advantages of `LTSM-Bundle` comparing to other existing open source LTSM packages, including OpenLTM[8], Time-LLM [9], and LLM-Time[10]. Our package involve more industrial-oriented and user-friendly features, such as database integration and report visualization.

Table 7: Feature comparison with other LTSM open source packages

	LTSM-Bundle	OpenLTM	Time-LLM	LLM-Time
Support for multiple model architectures and prompting strategies	**Yes**	Yes	No	No
Integration with database	**Yes**	No	No	No
Data preprocessing and pipeline integration	**Yes**	No	No	No
Zero-shot	**Yes**	Yes	Yes	Yes
Visualization	**Yes**	No	No	Yes

E Comparison with Baselines

Based on the observations in Section 3, we identify a strong combination using `LTSM-Bundle` with the settings as follows: (1) Base model backbone: GPT-2-Medium, (2) Instruction prompts: the time series prompts, (3) Tokenization: linear tokenization, and (4) Training paradigm: fully fine-tuning. We compare this combination against SoTA TSF models on zero-shot and few-shot settings.

E.1 Experimental Settings

We follow the same settings as in Time-LLM [14]. Specifically, for zero-shot experiments, we test the model's cross-domain adaptation under the long-term forecasting scenario and evaluate it on various cross-domain scenarios utilizing the ETT datasets. The hyperparameter settings of training `LTSM-Bundle` are in Appendix B. For the few-shot setting, we train our `LTSM-Bundle` on 5% of the data and compare it with other baselines under the 5% as well. We cite the performance of other models when applicable [33]. Furthermore, we compare `LTSM-Bundle` trained on 5% training data against baselines trained on the full training set. Our findings in Appendix H indicate that `LTSM-Bundle` achieves comparable results, further underscoring its superiority.

The baseline methods consist of various Transformer-based methods, including PatchTST [20], ETSformer [27], Non-Stationary Transformer [19], FEDformer [32], Autoformer [4], Informer [31], and Reformer [15]. Additionally, we evaluate our model against recent competitive models like Time-LLM [14], TEST [24], LLM4TS [3], GPT4TS [33], DLinear [30], TimesNet [28], and LightTS [2]. More details of the baseline methods can be found in Section G.

F Zero-shot and Few-shot Results of LTSM-bundle

Zero-shot Performance In the zero-shot learning experiments shown in Table 8 shows that the best component combination from benchmarking `LTSM-Bundle` consistently delivers superior perfor-

[8] `https://github.com/thuml/OpenLTM`
[9] `https://github.com/KimMeen/Time-LLM`
[10] `https://github.com/ngruver/llmtime`

Table 8: Zero-shot performance. "LTSM-Bundle" denotes the best combination of LTSM training components

	LTSM-Bundle		TIME-LLM		GPT4TS		LLMTime		DLinear		PatchTST		TimesNet	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1 → ETTh2	0.319	0.402	0.353	0.387	0.406	0.422	0.992	0.708	0.493	0.488	0.380	0.405	0.421	0.431
ETTh1 → ETTm2	0.312	0.406	0.273	0.340	0.325	0.363	1.867	0.869	0.415	0.452	0.314	0.360	0.327	0.361
ETTm1 → ETTh2	0.306	0.391	0.381	0.412	0.433	0.439	0.992	0.708	0.464	0.475	0.439	0.438	0.457	0.454
ETTm1 → ETTm2	0.217	0.319	0.268	0.320	0.313	0.348	1.867	0.869	0.335	0.389	0.296	0.334	0.322	0.354
ETTm2 → ETTh2	0.314	0.393	0.354	0.400	0.435	0.443	1.867	0.869	0.455	0.471	0.409	0.425	0.435	0.443
ETTm2 → ETTm1	0.403	0.430	0.414	0.438	0.769	0.567	1.933	0.984	0.649	0.537	0.568	0.492	0.769	0.567

Table 9: The average performance of Few-shot under (5% training data)

Dataset	LTSM-Bundle		**TIME-LLM**		**DLinear**	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.338	0.403	0.627	0.543	0.750	0.611
ETTh2	0.303	0.387	0.382	0.418	0.694	0.577
ETTm1	0.362	0.416	0.425	0.434	0.400	0.417
ETTm2	0.239	0.335	0.274	0.323	0.399	0.426
Weather	0.251	0.305	0.260	0.309	0.263	0.308
Electricity	0.175	0.276	0.179	0.268	0.173	0.266
Traffic	0.323	0.285	0.423	0.298	0.450	0.317
1^{st} Count	6		1		0	

mance across various cross-domain scenarios using the ETT datasets. For example, in the ETTh1 to ETTh2 dataset transfer task, LTSM-Bundle achieves an MSE of 0.319 and an MAE of 0.402, outperforming all other methods, including TIME-LLM, GPT4TS, and DLinear. Similarly, in the ETTm1 to ETTm2 dataset transfer scenario, LTSM-Bundle records the lowest MSE and MAE scores of 0.217 and 0.319, showing its strong generalization capability across different domains. The consistent improvements across transfer tasks of LTSM-Bundle in zero-shot learning.

Few-shot Performance Table 9 presents the performance of the best component combination from benchmarking compared to the baseline models in the few-shot setting, utilizing 5% of the training data. Notably, LTSM-Bundle exhibits a significant advantage over both traditional baselines and existing LTSMs. Across the 7 datasets, LTSM-Bundle outperforms all baselines regarding MSE in 5 datasets and regarding MAE in 4 datasets. Moreover, LTSM-Bundle achieves the top rank 40 times among the reported results. These findings underscore the effectiveness of our model in few-shot scenarios, where it demonstrates high accuracy even with limited training data. Its capability to excel with minimal data not only highlights its adaptability but also its potential for practical applications, particularly in contexts where data availability is constrained. All further and full versions of results on the full datasets are provided in Appendix H.

G Related Works

In this work, we focus on benchmarking the training paradigms of LTSMs on top of decoder-only single models. The other related works, as well as our benchmarking baselines, are illustrated as follows. PatchTST [20] employs a patch-based technique for time series forecasting, leveraging the self-attention mechanism of transformers. ETSformer [27] integrates exponential smoothing with transformer architectures to improve forecast accuracy. The Non-Stationary Transformer [19] addresses non-stationarity by adapting to changes in statistical properties over time. FEDformer [32] incorporates information in the frequency domain to handle periodic patterns. Autoformer [4] introduces an autocorrelation mechanism to capture long-term dependencies and seasonality patterns. Informer [31] optimizes transformers for long sequence forecasting with an efficient self-attention mechanism. Reformer [15] uses locality-sensitive hashing and reversible layers to improve memory and computational efficiency. Additionally, we also consider several competitive models in the pursuit of time series foundation models. Time-LLM [14] leverages large language models for time series forecasting, treating data as a sequence of events. TEST [24] handles complex temporal dependencies with an enhanced transformer architecture. LLM4TS [3] uses large language models adapted for time series forecasting. GPT4TS [33] adapts the Frozen Pretrained Transformer (FPT) for generating future predictions. DLinear [30] that focuses on capturing linear trends with a linear layer model. TimesNet [28] integrates neural network architectures to capture complex patterns. LightTS [2] provides efficient and fast forecasting solutions suitable for real-time applications.

To the best of our knowledge, no prior art has provided a comprehensive benchmark to analyze the effectiveness of each component in training LTSMs. Some works [17, 18] maintain a fair platform to compare different time series forecasting methods. The others [9, 21] analyze the forecasting performance from the perspectives of time series patterns. This benchmark provides an accessible and modular pipeline for evaluating a diverse set of training components in LTSM development, leveraging a time series database and user-friendly visualization. With the debates on whether LLMs can benefit from time series forecasting tasks, our toolbox offers a scikit-learn-like API interface to efficiently explore each component's effectiveness in training LTSMs.

H Additional Experimental Results on `LTSM-Bundle`

In this section, we show additional results regarding comparing `LTSM-Bundle` with other baselines in Tables 13 and 14, results of zero-shot transfer learning in Table 11, results of different training paradigms in Table 15, results of different backbones in Table 16, results of different downsampling ratios in Table 17.

H.1 Impacts of different prompts for LTSM training

Instruction prompts enhance the effectiveness of LTSM training by providing auxiliary information. This prompt helps the model adjust its internal state and focus more on relevant features in different domains of the dataset, thereby improving learning accuracy. With the aid of prompts, LTSM aims to optimize forecasting ability across diverse dataset domains. We explore two types of prompts: the *Text Prompts* [14] written in task-specific information, and the *time series prompts* developed by global features of time series data. This comparison determines the most effective prompt type for LTSM training.

Time Series Prompts Time series prompts aim to capture the comprehensive properties of time series data. Unlike instruct prompts, they are derived from a diverse set of global features extracted from the entire training dataset. This approach ensures a robust representation of the underlying dynamics, crucial for enhancing model performance. The time series prompts are generated by extracting global features from each variate of the time series training data. The extracted global features are specified in Appendix **??**. After extracting the global features, we proceed to standardize their values across all varieties and instances within the dataset. This standardization is crucial to prevent the overflow issue during both training and inference stages. Let $\mathbf{P} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_M\}$ denote the global features of $\boldsymbol{Z}$ after the standardization, where $\boldsymbol{p}_t \in \mathbb{R}^d$. Subsequently, $\mathbf{P}$ serves as prompts, being concatenated with each timestamp $\mathbf{X}$ derived from the Time series data. Consequently, the large Time series models take the integrated vector $\tilde{\mathbf{X}} = \mathbf{P} \cup \mathbf{X} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_M, \mathbf{z}_{t_1}, \mathbf{z}_{t_2}, ..., \mathbf{z}_{t_P}\}$ as input data throughout both training and inference phases, as illustrated in Figure **??**. The time series prompts are generated separately for the training and testing datasets, without leaking the testing data information to the training process. We leverage the package[11] to generate the time series prompts.

After extracting the global features, we proceed to standardize their values across all varieties and instances within the dataset. This standardization is crucial to prevent the overflow issue during both training and inference stages. Let $\mathbf{P} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_M\}$ denote the global features of $\boldsymbol{Z}$ after the standardization, where $\boldsymbol{p}_t \in \mathbb{R}^d$. Subsequently, $\mathbf{P}$ serves as prompts, being concatenated with each timestamp $\mathbf{X}$ derived from the Time series data. Consequently, the LTSMs take the integrated vector $\tilde{\mathbf{X}} = \mathbf{P} \cup \mathbf{X} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_M, \mathbf{z}_{t_1}, \mathbf{z}_{t_2}, ..., \mathbf{z}_{t_P}\}$ as input data throughout both training and inference phases, depicted in Figure **??**. The time series prompts are generated separately for the training and testing datasets without leaking the testing data information to the training process.

[11] `https://github.com/thuml/Time-Series-Library`

Table 10: Performance of different prompting strategies

Metric	Input	ETTh1	ETTh2	ETTm1	ETTm2	Traffic	Weather	Electricity	Avg.
MSE	No prompt	0.308	0.237	0.367	0.157	0.306	0.177	0.148	0.243
	TS prompt	0.307	0.234	0.285	0.155	0.305	0.172	0.145	0.229
	Text prompt	0.319	0.241	0.490	0.190	0.345	0.212	0.185	0.283
MAE	No prompt	0.375	0.325	0.411	0.258	0.272	0.232	0.246	0.303
	TS prompt	0.377	0.326	0.369	0.266	0.279	0.242	0.247	0.301
	Text prompt	0.386	0.329	0.476	0.289	0.326	0.269	0.299	0.339

Experimental Results We begin by evaluating the effectiveness of instruction prompts. Specifically, we assess two distinct types of instruction prompts, both initialized by the same pre-trained GP2-Medium weights within the context of commonly used linear tokenization. The experimental results are shown in Table 10. Our observations suggest that 1 statistical prompts outperform traditional text prompts in enhancing the training of LTSM models with up to 8% lower MAE scores. Additionally, 2 it is observed that the use of statistical prompts results in superior performance compared to scenarios where no prompts are employed, yielding up to 3% lower MSE scores. The superiority of statistical prompt is evident in the more effective leveraging of LTSM capabilities, leading to improved learning outcomes across various datasets. Based on the above observations, we select time series prompts as the focus in the following analysis and incorporate them into `LTSM-bundle`.

H.2 Performance Comparison with Additional Baselines

Extending the analysis presented in Section F, we introduce full performance comparison with new baselines. We evaluate the proposed `LTSM-Bundle` in zero-shot and few-shot settings to highlight its efficacy and robustness in Table 13 and 14.

H.3 Zero-shot Transfer Learning Comparisons

In addition to the results in Section F, this section introduces the full zero-shot transfer learning comparisons. We evaluate the proposed `LTSM-Bundle` in the zero-shot transfer scenarios, detailed in shown in Table 11.

H.4 Training Paradigm Comparisons

Expanding upon the results in Section 3.1, this section presents the full experimental results for the training paradigms analysis, including different backbones and prompting strategies. The analytic results are detailed in Table 15.

H.5 Backbone Architecture Comparisons

We provide all the numbers of analytics on different backbone architectures. Results are in different language model backbones, including GPT-2-Small, GPT-2-Medium, GPT-2-Large, and Phi-2, shown in Table 16.

H.6 Down-sampling Ratio Comparisons

We here present the full version of our experimental results on the different down-sampling ratios in Section A. We test `LTSM-Bundle` with GPT-Medium as backbones with the proposed TS prompt under a fully tuning paradigm. The results in the ratio of {40, 20, 10} (i.e., downsample rate in {2.5%, 5%, 10% }) are all demonstrated in Table 17.

H.7 Different Numbers of Layer Adaptation Comparisons

We compared the average performance among all datasets of a 3-layer model and a full 24-layer model using GPT-medium as the backbone. Our results (in Table 12) show that the 24-layer model performs worse when trained with the same number of iterations. We believe this suggests that the 3-layer configuration is a reasonable strategy for benchmarking at this stage.

Table 11: Results of zero-shot transfer learning. A time-series model is trained on a source dataset and transferred to the target dataset without adaptation.

Methods		`LTSM-Bundle`		TIME-LLM		LLMTime		GPT4TS		DLinear		PatchTST		TimesNet		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 → ETTh2	96	0.229	0.326	0.279	0.337	0.510	0.576	0.335	0.374	0.347	0.400	0.304	0.350	0.358	0.387	0.469	0.486
	192	0.310	0.395	0.351	0.374	0.523	0.586	0.412	0.417	0.417	0.460	0.386	0.400	0.427	0.429	0.634	0.567
	336	0.336	0.414	0.388	0.415	0.640	0.637	0.441	0.444	0.515	0.505	0.414	0.428	0.449	0.451	0.655	0.588
	720	0.401	0.474	0.391	0.420	2.296	1.034	0.438	0.452	0.665	0.589	0.419	0.443	0.448	0.458	0.570	0.549
	Avg	0.319	0.402	0.353	0.387	0.992	0.708	0.406	0.422	0.493	0.488	0.380	0.405	0.421	0.431	0.582	0.548
ETTh1 → ETTm2	96	0.197	0.318	0.189	0.293	0.646	0.563	0.236	0.315	0.255	0.357	0.215	0.304	0.239	0.313	0.352	0.432
	192	0.314	0.420	0.237	0.312	0.934	0.654	0.287	0.342	0.338	0.413	0.275	0.339	0.291	0.342	0.413	0.460
	336	0.313	0.405	0.291	0.365	1.157	0.728	0.341	0.374	0.425	0.465	0.334	0.373	0.342	0.371	0.465	0.489
	720	0.425	0.483	0.372	0.390	4.730	1.531	0.435	0.422	0.640	0.573	0.431	0.424	0.434	0.419	0.599	0.551
	Avg	0.312	0.406	0.273	0.340	1.867	0.869	0.325	0.363	0.415	0.452	0.314	0.360	0.327	0.361	0.457	0.483
ETTh2 → ETTh1	96	0.390	0.439	0.450	0.452	1.130	0.777	0.732	0.577	0.689	0.555	0.485	0.465	0.848	0.601	0.693	0.569
	192	0.417	0.460	0.465	0.461	1.242	0.820	0.758	0.559	0.707	0.568	0.565	0.509	0.860	0.610	0.760	0.601
	336	0.462	0.501	0.501	0.482	1.382	0.864	0.759	0.578	0.710	0.577	0.581	0.515	0.867	0.626	0.781	0.619
	720	0.568	0.588	0.501	0.502	4.145	1.461	0.781	0.597	0.704	0.596	0.628	0.561	0.887	0.648	0.796	0.644
	Avg	0.459	0.497	0.479	0.474	1.961	0.981	0.757	0.578	0.703	0.574	0.565	0.513	0.865	0.621	0.757	0.608
ETTh2 → ETTm2	96	0.200	0.316	0.174	0.276	0.646	0.563	0.253	0.329	0.240	0.336	0.226	0.309	0.248	0.324	0.263	0.352
	192	0.250	0.359	0.233	0.315	0.934	0.654	0.293	0.346	0.295	0.369	0.289	0.345	0.296	0.352	0.326	0.389
	336	0.327	0.416	0.291	0.337	1.157	0.728	0.347	0.376	0.345	0.397	0.348	0.379	0.353	0.383	0.387	0.426
	720	0.573	0.563	0.392	0.417	4.730	1.531	0.446	0.429	0.432	0.442	0.439	0.427	0.471	0.446	0.487	0.478
	Avg	0.337	0.413	0.272	0.341	1.867	0.869	0.335	0.370	0.328	0.386	0.325	0.365	0.342	0.376	0.366	0.411
ETTm1 → ETTh2	96	0.246	0.342	0.321	0.369	0.510	0.576	0.353	0.392	0.365	0.415	0.354	0.385	0.377	0.407	0.435	0.470
	192	0.290	0.374	0.389	0.410	0.523	0.586	0.443	0.437	0.454	0.462	0.447	0.434	0.471	0.453	0.495	0.489
	336	0.326	0.406	0.408	0.433	0.640	0.637	0.469	0.461	0.496	0.464	0.481	0.463	0.472	0.484	0.470	0.472
	720	0.363	0.440	0.406	0.436	2.296	1.034	0.466	0.468	0.541	0.529	0.474	0.471	0.495	0.482	0.480	0.485
	Avg	0.306	0.391	0.381	0.412	0.992	0.708	0.433	0.439	0.464	0.475	0.439	0.438	0.457	0.454	0.470	0.479
ETTm1 → ETTm2	96	0.144	0.257	0.169	0.257	0.646	0.563	0.217	0.294	0.221	0.314	0.195	0.271	0.222	0.295	0.385	0.457
	192	0.193	0.302	0.227	0.318	0.934	0.654	0.277	0.327	0.286	0.359	0.258	0.311	0.288	0.337	0.433	0.469
	336	0.240	0.342	0.290	0.338	1.157	0.728	0.331	0.360	0.357	0.406	0.317	0.348	0.341	0.367	0.476	0.477
	720	0.292	0.379	0.375	0.367	4.730	1.531	0.429	0.413	0.476	0.476	0.416	0.404	0.436	0.418	0.582	0.535
	Avg	0.217	0.320	0.268	0.320	1.867	0.869	0.313	0.348	0.335	0.389	0.296	0.334	0.322	0.354	0.469	0.484
ETTm2 → ETTh2	96	0.257	0.346	0.298	0.356	0.510	0.576	0.360	0.401	0.333	0.391	0.327	0.367	0.360	0.401	0.353	0.393
	192	0.309	0.382	0.359	0.397	0.523	0.586	0.434	0.437	0.441	0.456	0.411	0.418	0.434	0.437	0.432	0.437
	336	0.341	0.413	0.367	0.412	0.640	0.637	0.460	0.459	0.505	0.503	0.439	0.447	0.460	0.459	0.452	0.459
	720	0.350	0.432	0.393	0.434	2.296	1.034	0.485	0.477	0.543	0.534	0.459	0.470	0.485	0.477	0.453	0.467
	Avg	0.314	0.393	0.354	0.400	0.992	0.708	0.435	0.443	0.455	0.471	0.409	0.425	0.435	0.443	0.423	0.439
ETTm2 → ETTm1	96	0.364	0.410	0.359	0.397	1.179	0.781	0.747	0.558	0.570	0.490	0.491	0.437	0.747	0.558	0.735	0.576
	192	0.405	0.432	0.390	0.420	1.327	0.846	0.781	0.560	0.590	0.506	0.530	0.470	0.781	0.560	0.753	0.586
	336	0.413	0.433	0.421	0.445	1.478	0.902	0.778	0.578	0.706	0.567	0.565	0.497	0.778	0.578	0.750	0.593
	720	0.432	0.446	0.487	0.488	3.749	1.408	0.769	0.573	0.731	0.584	0.686	0.565	0.769	0.573	0.782	0.609
	Avg	0.403	0.430	0.414	0.438	1.933	0.984	0.769	0.567	0.649	0.537	0.568	0.492	0.769	0.667	0.755	0.591

Table 12: Performance comparison between 3-layer and 24-layer of `LTSM-Bundle`.

	3-layer	24-layer
MAE	0.2003	0.2439
MSE	0.2770	0.3162

Table 13: Performance comparison with additional baselines (Full data)

Methods		LTSM-Bundle		TIME-LLM		TEST		LLM4TS		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer		Non-Stationary		ETSformer		LightTS		Informer		Reformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.307	0.377	0.362	0.392	0.372	0.400	0.371	0.394	0.376	0.397	0.375	0.399	0.370	0.399	0.384	0.402	0.376	0.419	0.449	0.459	0.513	0.491	0.494	0.479	0.424	0.432	0.865	0.713	0.837	0.728
	192	0.329	0.391	0.398	0.418	0.414	0.422	0.403	0.412	0.416	0.418	0.405	0.416	0.413	0.421	0.436	0.429	0.420	0.448	0.500	0.482	0.534	0.504	0.538	0.504	0.475	0.462	1.008	0.792	0.923	0.766
	336	0.346	0.405	0.430	0.427	0.422	0.437	0.420	0.422	0.442	0.433	0.439	0.443	0.422	0.436	0.491	0.469	0.459	0.465	0.521	0.496	0.588	0.535	0.574	0.521	0.518	0.488	1.107	0.809	1.097	0.835
	720	0.370	0.441	0.442	0.457	0.447	0.467	0.422	0.444	0.477	0.456	0.472	0.490	0.447	0.466	0.521	0.500	0.506	0.507	0.514	0.512	0.643	0.616	0.562	0.535	0.547	0.533	1.181	0.865	1.257	0.889
	Avg	0.338	0.403	0.408	0.423	0.414	0.431	0.404	0.418	0.465	0.455	0.422	0.437	0.413	0.430	0.458	0.450	0.440	0.460	0.496	0.487	0.570	0.537	0.542	0.510	0.491	0.479	1.040	0.795	1.029	0.805
ETTh2	96	0.235	0.326	0.268	0.328	0.275	0.338	0.269	0.332	0.285	0.342	0.289	0.353	0.274	0.336	0.340	0.374	0.358	0.397	0.346	0.388	0.476	0.458	0.340	0.391	0.397	0.437	3.755	1.525	2.626	1.317
	192	0.283	0.365	0.329	0.375	0.340	0.379	0.328	0.377	0.354	0.389	0.383	0.418	0.339	0.379	0.402	0.414	0.429	0.439	0.456	0.452	0.512	0.493	0.430	0.439	0.520	0.504	5.602	1.931	11.12	2.979
	336	0.320	0.401	0.368	0.409	0.329	0.381	0.353	0.369	0.373	0.407	0.448	0.465	0.329	0.380	0.452	0.452	0.496	0.487	0.482	0.486	0.552	0.551	0.485	0.479	0.626	0.559	4.721	1.835	9.323	2.769
	720	0.378	0.456	0.372	0.420	0.381	0.423	0.383	0.425	0.406	0.441	0.605	0.551	0.379	0.422	0.462	0.468	0.463	0.474	0.515	0.511	0.562	0.560	0.500	0.497	0.863	0.672	3.647	1.625	3.874	1.697
	Avg	0.304	0.387	0.334	0.383	0.331	0.380	0.333	0.376	0.381	0.412	0.431	0.446	0.330	0.379	0.414	0.427	0.437	0.449	0.450	0.459	0.526	0.516	0.439	0.452	0.602	0.543	4.431	1.729	6.736	2.191
ETTm1	96	0.285	0.369	0.272	0.334	0.293	0.346	0.285	0.343	0.292	0.346	0.299	0.343	0.290	0.342	0.338	0.375	0.379	0.419	0.505	0.475	0.386	0.398	0.375	0.398	0.374	0.400	0.672	0.571	0.538	0.528
	192	0.319	0.393	0.310	0.358	0.332	0.369	0.324	0.366	0.332	0.372	0.335	0.365	0.332	0.369	0.374	0.387	0.426	0.441	0.553	0.496	0.459	0.444	0.408	0.410	0.400	0.407	0.795	0.669	0.658	0.592
	336	0.378	0.425	0.352	0.384	0.368	0.392	0.353	0.385	0.366	0.394	0.369	0.386	0.366	0.392	0.410	0.411	0.445	0.459	0.621	0.537	0.495	0.464	0.435	0.428	0.438	0.438	1.212	0.871	0.898	0.721
	720	0.464	0.477	0.383	0.411	0.418	0.420	0.408	0.419	0.417	0.421	0.425	0.421	0.416	0.420	0.478	0.450	0.543	0.490	0.671	0.561	0.585	0.516	0.499	0.462	0.527	0.502	1.166	0.823	1.102	0.841
	Avg	0.362	0.416	0.329	0.372	0.353	0.382	0.343	0.378	0.388	0.403	0.357	0.378	0.351	0.380	0.400	0.406	0.448	0.452	0.588	0.517	0.481	0.456	0.429	0.425	0.435	0.437	0.961	0.734	0.799	0.671
ETTm2	96	0.156	0.266	0.161	0.253	-	-	0.165	0.254	0.173	0.262	0.167	0.269	0.165	0.255	0.187	0.267	0.203	0.287	0.255	0.339	0.192	0.274	0.189	0.280	0.209	0.308	0.365	0.453	0.658	0.619
	192	0.203	0.307	0.219	0.293	-	-	0.220	0.292	0.229	0.301	0.224	0.303	0.220	0.292	0.249	0.309	0.269	0.328	0.281	0.340	0.280	0.339	0.253	0.319	0.311	0.382	0.533	0.563	1.078	0.827
	336	0.255	0.349	0.271	0.329	-	-	0.268	0.326	0.286	0.341	0.281	0.342	0.274	0.329	0.321	0.351	0.325	0.366	0.339	0.372	0.334	0.361	0.314	0.357	0.442	0.466	1.363	0.887	1.549	0.972
	720	0.342	0.417	0.352	0.379	-	-	0.350	0.380	0.378	0.401	0.397	0.421	0.362	0.385	0.408	0.403	0.421	0.415	0.433	0.432	0.417	0.413	0.414	0.413	0.675	0.587	3.379	1.338	2.631	1.242
	Avg	0.239	0.335	0.251	0.313	-	-	0.251	0.313	0.284	0.339	0.267	0.333	0.255	0.315	0.291	0.333	0.305	0.349	0.327	0.371	0.306	0.347	0.293	0.342	0.409	0.436	1.410	0.810	1.479	0.915
Weather	96	0.172	0.242	0.147	0.201	0.150	0.202	0.147	0.196	0.162	0.212	0.176	0.237	0.149	0.198	0.172	0.220	0.217	0.296	0.266	0.336	0.173	0.223	0.197	0.281	0.182	0.242	0.300	0.384	0.689	0.596
	192	0.218	0.278	0.189	0.234	0.198	0.246	0.191	0.238	0.204	0.248	0.220	0.282	0.194	0.241	0.219	0.261	0.276	0.336	0.307	0.367	0.245	0.285	0.237	0.312	0.227	0.287	0.598	0.544	0.752	0.638
	336	0.276	0.329	0.262	0.279	0.245	0.286	0.241	0.277	0.254	0.286	0.265	0.319	0.245	0.282	0.280	0.306	0.339	0.380	0.359	0.395	0.321	0.338	0.298	0.353	0.282	0.334	0.578	0.523	0.639	0.596
	720	0.339	0.373	0.304	0.316	0.324	0.342	0.313	0.329	0.326	0.337	0.333	0.362	0.314	0.334	0.365	0.359	0.403	0.428	0.419	0.428	0.414	0.410	0.352	0.288	0.352	0.386	1.059	0.741	1.130	0.792
	Avg	0.251	0.305	0.225	0.257	0.229	0.271	0.223	0.260	0.237	0.270	0.248	0.300	0.225	0.264	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.261	0.312	0.634	0.548	0.803	0.656
Electricity	96	0.145	0.247	0.131	0.224	0.132	0.223	0.128	0.223	0.139	0.238	0.140	0.237	0.129	0.222	0.168	0.272	0.193	0.308	0.201	0.317	0.169	0.273	0.187	0.304	0.207	0.307	0.274	0.368	0.312	0.402
	192	0.159	0.259	0.152	0.241	0.158	0.241	0.146	0.240	0.153	0.251	0.153	0.249	0.157	0.240	0.184	0.289	0.201	0.315	0.222	0.334	0.182	0.286	0.199	0.315	0.213	0.316	0.296	0.386	0.348	0.433
	336	0.180	0.284	0.160	0.248	0.163	0.260	0.163	0.258	0.169	0.266	0.169	0.267	0.163	0.259	0.198	0.300	0.214	0.329	0.231	0.338	0.200	0.304	0.212	0.329	0.230	0.333	0.300	0.394	0.350	0.433
	720	0.215	0.317	0.192	0.298	0.199	0.291	0.200	0.292	0.206	0.297	0.203	0.301	0.197	0.290	0.220	0.320	0.246	0.355	0.254	0.361	0.222	0.321	0.233	0.345	0.265	0.360	0.373	0.439	0.340	0.420
	Avg	0.175	0.276	0.158	0.252	0.162	0.253	0.159	0.253	0.167	0.263	0.166	0.263	0.161	0.252	0.192	0.295	0.214	0.327	0.227	0.338	0.193	0.296	0.208	0.323	0.229	0.329	0.311	0.397	0.338	0.422
Traffic	96	0.305	0.279	0.362	0.248	0.407	0.282	0.372	0.259	0.388	0.282	0.410	0.282	0.360	0.249	0.593	0.321	0.587	0.366	0.613	0.388	0.612	0.338	0.607	0.392	0.615	0.391	0.719	0.391	0.732	0.423
	192	0.313	0.274	0.374	0.247	0.423	0.287	0.391	0.265	0.407	0.290	0.423	0.287	0.379	0.256	0.617	0.336	0.604	0.373	0.616	0.382	0.613	0.340	0.621	0.399	0.601	0.382	0.696	0.379	0.733	0.420
	336	0.326	0.287	0.385	0.271	0.430	0.296	0.405	0.275	0.412	0.294	0.436	0.296	0.392	0.264	0.629	0.336	0.621	0.383	0.622	0.337	0.618	0.328	0.622	0.396	0.613	0.386	0.777	0.420	0.742	0.420
	720	0.346	0.301	0.430	0.288	0.463	0.315	0.437	0.292	0.450	0.312	0.466	0.315	0.432	0.286	0.640	0.350	0.626	0.382	0.660	0.408	0.653	0.355	0.632	0.396	0.658	0.407	0.864	0.472	0.755	0.423
	Avg	0.323	0.285	0.388	0.264	0.430	0.295	0.401	0.273	0.414	0.294	0.433	0.295	0.390	0.263	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.622	0.392	0.764	0.416	0.741	0.422

Table 14: Performance comparison with additional baselines (5% Few Shot data)

Methods		LTSM-Bundle		TIME-LLM		LLM4TS		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer		Non-Stationary		ETSformer		LightTS		Informer		Reformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.307	0.377	0.483	0.464	0.509	0.484	0.543	0.506	0.547	0.503	0.557	0.519	0.892	0.62	5 0.593	0.529	0.681	0.570	0.952	0.650	1.169	0.832	1.483	0.91	1.225	0.812	1.198	0.795
	192	0.329	0.391	0.629	0.540	0.717	0.581	0.748	0.580	0.720	0.604	0.711	0.570	0.940	0.665	0.652	0.563	0.725	0.602	0.943	0.645	1.221	0.853	1.525	0.93	1.249	0.828	1.273	0.853
	336	0.346	0.405	0.768	0.626	0.728	0.589	0.754	0.595	0.984	0.727	0.816	0.619	0.945	0.653	0.731	0.594	0.761	0.624	0.935	0.644	1.179	0.832	1.347	0.87	1.202	0.811	1.254	0.857
	720	0.370	0.441	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	0.338	0.403	0.627	0.543	0.651	0.551	0.681	0.560	0.750	0.611	0.694	0.569	0.925	0.647	0.658	0.562	0.722	0.598	0.943	0.646	1.189	0.839	1.451	0.903	1.225	0.817	1.241	0.835
ETTh2	96	0.235	0.326	0.336	0.397	0.314	0.375	0.376	0.421	0.442	0.456	0.401	0.421	0.409	0.420	0.390	0.424	0.428	0.468	0.408	0.423	0.678	0.619	2.022	1.006	3.837	1.508	3.753	1.518
	192	0.283	0.365	0.406	0.425	0.365	0.408	0.418	0.441	0.617	0.542	0.452	0.455	0.483	0.464	0.457	0.465	0.496	0.504	0.497	0.468	0.845	0.697	3.534	1.348	3.975	1.933	3.516	1.473
	336	0.320	0.401	0.405	0.432	0.398	0.432	0.408	0.439	1.424	0.849	0.464	0.469	0.499	0.479	0.477	0.483	0.486	0.496	0.507	0.481	0.905	0.727	4.063	1.451	3.956	1.520	3.312	1.427
	720	0.378	0.456	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	0.304	0.387	0.382	0.418	0.359	0.405	0.400	0.433	0.694	0.577	0.439	0.448	0.439	0.448	0.463	0.454	0.441	0.457	0.470	0.489	0.809	0.681	3.206	1.268	3.922	1.653	3.527	1.472
ETTm1	96	0.285	0.369	0.316	0.377	0.349	0.379	0.386	0.405	0.332	0.374	0.399	0.414	0.606	0.518	0.628	0.544	0.726	0.578	0.823	0.587	1.031	0.747	1.048	0.733	1.130	0.775	1.234	0.798
	192	0.319	0.393	0.450	0.464	0.374	0.394	0.440	0.438	0.358	0.390	0.441	0.436	0.681	0.539	0.666	0.566	0.750	0.591	0.844	0.591	1.087	0.766	1.097	0.756	1.150	0.788	1.287	0.839
	336	0.378	0.425	0.450	0.424	0.411	0.417	0.485	0.459	0.402	0.416	0.499	0.467	0.786	0.597	0.807	0.628	0.851	0.659	0.870	0.603	1.138	0.787	1.147	0.775	1.198	0.809	1.288	0.842
	720	0.464	0.477	0.483	0.471	0.516	0.479	0.577	0.499	0.511	0.489	0.767	0.587	0.796	0.593	0.822	0.633	0.857	0.655	0.893	0.611	1.245	0.831	1.200	0.799	1.175	0.794	1.247	0.828
	Avg	0.362	0.416	0.425	0.434	0.412	0.417	0.472	0.450	0.400	0.417	0.526	0.476	0.717	0.561	0.730	0.592	0.796	0.620	0.857	0.598	1.125	0.782	1.123	0.765	1.163	0.791	1.264	0.826
ETTm2	96	0.156	0.266	0.174	0.261	0.192	0.273	0.199	0.280	0.236	0.326	0.206	0.288	0.220	0.299	0.229	0.320	0.232	0.322	0.238	0.316	0.404	0.485	1.108	0.772	3.599	1.478	3.883	1.545
	192	0.203	0.307	0.215	0.287	0.249	0.309	0.256	0.316	0.306	0.373	0.264	0.324	0.311	0.361	0.394	0.361	0.291	0.357	0.298	0.349	0.479	0.521	1.317	0.850	3.578	1.475	3.553	1.484
	336	0.255	0.349	0.273	0.330	0.301	0.342	0.318	0.353	0.380	0.423	0.334	0.367	0.338	0.366	0.378	0.427	0.478	0.517	0.353	0.380	0.552	0.555	1.415	0.879	3.561	1.473	3.446	1.460
	720	0.342	0.417	0.433	0.412	0.402	0.405	0.460	0.436	0.674	0.583	0.454	0.432	0.509	0.465	0.523	0.510	0.553	0.538	0.475	0.445	0.701	0.627	1.822	0.984	3.896	1.533	3.445	1.460
	Avg	0.239	0.335	0.274	0.323	0.286	0.332	0.308	0.346	0.399	0.426	0.314	0.352	0.344	0.372	0.381	0.404	0.388	0.433	0.341	0.372	0.534	0.547	1.415	0.871	3.658	1.489	3.581	1.487
Weather	96	0.172	0.242	0.172	0.263	0.173	0.227	0.175	0.230	0.184	0.242	0.171	0.224	0.207	0.253	0.229	0.309	0.227	0.299	0.215	0.252	0.218	0.295	0.230	0.285	0.497	0.497	0.406	0.435
	192	0.218	0.278	0.224	0.271	0.218	0.265	0.227	0.276	0.228	0.283	0.230	0.277	0.272	0.307	0.265	0.317	0.278	0.333	0.290	0.307	0.294	0.331	0.274	0.323	0.620	0.545	0.446	0.450
	336	0.276	0.329	0.282	0.321	0.276	0.310	0.286	0.322	0.279	0.322	0.294	0.326	0.313	0.328	0.353	0.392	0.351	0.393	0.353	0.348	0.359	0.398	0.318	0.355	0.649	0.547	0.465	0.459
	720	0.339	0.373	0.366	0.381	0.355	0.366	0.366	0.379	0.364	0.388	0.384	0.387	0.400	0.385	0.391	0.394	0.387	0.389	0.452	0.407	0.461	0.461	0.401	0.418	0.570	0.522	0.471	0.468
	Avg	0.251	0.305	0.260	0.309	0.251	0.292	0.263	0.301	0.263	0.308	0.269	0.303	0.298	0.318	0.309	0.353	0.310	0.353	0.327	0.328	0.333	0.371	0.305	0.345	0.584	0.527	0.447	0.453
Electricity	96	0.145	0.247	0.147	0.242	0.139	0.235	0.143	0.241	0.150	0.251	0.145	0.244	0.315	0.389	0.235	0.322	0.297	0.367	0.484	0.518	0.697	0.638	0.639	0.609	1.265	0.919	1.414	0.855
	192	0.159	0.259	0.158	0.241	0.155	0.249	0.159	0.255	0.163	0.263	0.163	0.260	0.318	0.396	0.247	0.341	0.308	0.375	0.501	0.531	0.718	0.648	0.772	0.678	1.298	0.939	1.240	0.919
	336	0.180	0.284	0.178	0.277	0.174	0.269	0.179	0.274	0.175	0.278	0.183	0.281	0.340	0.415	0.267	0.356	0.354	0.411	0.574	0.578	0.758	0.667	0.901	0.745	1.302	0.942	1.253	0.921
	720	0.215	0.317	0.224	0.312	0.222	0.310	0.233	0.323	0.219	0.311	0.233	0.323	0.635	0.613	0.318	0.394	0.426	0.466	0.952	0.786	1.028	0.788	1.200	0.871	1.259	0.919	1.249	0.921
	Avg	0.175	0.276	0.179	0.268	0.173	0.266	0.178	0.273	0.176	0.275	0.181	0.277	0.402	0.453	0.266	0.353	0.346	0.404	0.627	0.603	0.800	0.685	0.878	0.725	1.281	0.929	1.289	0.904
Traffic	96	0.305	0.279	0.414	0.291	0.401	0.285	0.419	0.298	0.427	0.304	0.404	0.286	0.854	0.492	0.670	0.421	0.795	0.481	1.468	0.821	1.643	0.855	1.157	0.636	1.557	0.821	1.586	0.841
	192	0.313	0.274	0.419	0.291	0.418	0.293	0.434	0.305	0.447	0.315	0.412	0.294	0.894	0.517	0.653	0.405	0.837	0.503	1.509	0.838	1.856	0.928	1.688	0.848	1.596	0.834	1.602	0.844
	336	0.326	0.287	0.437	0.314	0.436	0.308	0.449	0.313	0.478	0.333	0.439	0.310	0.853	0.471	0.707	0.445	0.867	0.523	1.602	0.860	2.080	0.999	1.826	0.903	1.621	0.841	1.668	0.868
	720	0.346	0.301	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	0.323	0.285	0.423	0.298	0.418	0.295	0.434	0.305	0.450	0.317	0.418	0.296	0.867	0.493	0.676	0.423	0.833	0.502	1.526	0.839	1.859	0.927	1.557	0.795	1.591	0.832	1.618	0.851

Table 15: Results of different backbones, training paradigms, and prompting strategies.

Datasets		ETTh1		ETTh2		ETTm1		ETTm2		Traffic		Weather		Exchange		ECL	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
	96	0.354	0.415	0.259	0.350	0.551	0.507	0.215	0.319	0.393	0.377	0.222	0.292	0.108	0.246	0.250	0.342
From scratch	192	0.364	0.421	0.537	0.505	0.231	0.331	0.235	0.335	0.373	0.356	0.246	0.309	0.143	0.288	0.231	0.331
+ GPT-Medium	336	0.359	0.420	0.321	0.402	0.423	0.454	0.267	0.360	0.357	0.329	0.283	0.335	0.207	0.344	0.217	0.323
+ TS prompt	720	0.357	0.430	0.372	0.449	0.398	0.444	0.360	0.434	0.347	0.311	0.342	0.388	0.358	0.461	0.211	0.317
	Avg	0.358	0.421	0.372	0.427	0.401	0.434	0.269	0.362	0.367	0.343	0.273	0.331	0.204	0.335	0.227	0.328
	96	0.453	0.483	0.350	0.422	0.757	0.613	0.338	0.420	0.659	0.552	0.343	0.398	0.223	0.353	0.605	0.507
From scratch	192	0.422	0.470	0.348	0.423	0.708	0.601	0.326	0.415	0.509	0.475	0.323	0.388	0.212	0.352	0.352	0.403
+ GPT-Medium	336	0.481	0.502	0.449	0.487	0.938	0.701	0.483	0.506	0.562	0.496	0.457	0.474	0.430	0.502	0.729	0.456
+ Text prompt	720	0.437	0.482	0.396	0.461	0.634	0.563	0.408	0.459	0.536	0.463	0.415	0.434	0.457	0.517	0.460	0.438
	Avg	0.448	0.484	0.385	0.448	0.759	0.619	0.389	0.450	0.566	0.496	0.384	0.423	0.330	0.431	0.537	0.451
	96	0.323	0.392	0.243	0.341	0.394	0.437	0.185	0.301	0.334	0.321	0.200	0.279	0.098	0.236	0.197	0.291
From scratch	192	0.332	0.399	0.275	0.362	0.369	0.426	0.204	0.313	0.333	0.306	0.219	0.286	0.127	0.268	0.195	0.290
+ GPT-Small	336	0.345	0.405	0.317	0.394	0.352	0.415	0.263	0.364	0.324	0.288	0.265	0.323	0.179	0.321	0.181	0.282
+ TS prompt	720	0.362	0.432	0.364	0.447	0.389	0.439	0.324	0.402	0.341	0.300	0.339	0.377	0.333	0.457	0.207	0.309
	Avg	0.340	0.407	0.300	0.386	0.376	0.429	0.244	0.345	0.333	0.304	0.256	0.316	0.184	0.320	0.195	0.293
	96	0.317	0.383	0.240	0.334	0.431	0.443	0.178	0.285	0.315	0.293	0.188	0.257	0.083	0.208	0.161	0.265
Fully tune	192	0.355	0.413	0.285	0.370	0.479	0.482	0.221	0.323	0.352	0.336	0.238	0.304	0.132	0.277	0.213	0.311
+ GPT-Small	336	0.357	0.414	0.302	0.388	0.486	0.486	0.252	0.346	0.339	0.310	0.275	0.326	0.196	0.336	0.203	0.302
+ TS prompt	720	0.363	0.434	0.361	0.442	0.479	0.483	0.345	0.420	0.350	0.313	0.345	0.384	0.426	0.496	0.224	0.326
	Avg	0.348	0.411	0.297	0.384	0.469	0.473	0.249	0.343	0.339	0.313	0.261	0.317	0.209	0.329	0.200	0.301
	96	0.305	0.377	0.226	0.320	0.276	0.360	0.143	0.253	0.305	0.279	0.162	0.227	0.060	0.178	0.144	0.246
Fully tune	192	0.335	0.397	0.278	0.359	0.314	0.389	0.191	0.295	0.315	0.283	0.212	0.275	0.118	0.253	0.161	0.261
+ GPT-Small	336	0.348	0.406	0.310	0.392	0.344	0.411	0.239	0.339	0.323	0.285	0.266	0.318	0.198	0.333	0.175	0.277
+ Text prompt	720	0.371	0.454	0.364	0.446	0.404	0.452	0.352	0.405	0.344	0.305	0.332	0.366	0.379	0.470	0.208	0.309
	Avg	0.340	0.409	0.294	0.379	0.334	0.403	0.231	0.323	0.322	0.288	0.243	0.296	0.189	0.308	0.172	0.273
	96	0.301	0.372	0.229	0.320	0.261	0.346	0.149	0.266	0.300	0.268	0.163	0.230	0.058	0.173	0.141	0.241
Fully tune	192	0.332	0.397	0.290	0.368	0.288	0.370	0.204	0.303	0.316	0.282	0.215	0.282	0.133	0.277	0.158	0.258
+ GPT-Medium	336	0.351	0.412	0.316	0.392	0.343	0.413	0.294	0.376	0.328	0.295	0.281	0.332	0.224	0.369	0.175	0.276
+ Text prompt	720	0.368	0.436	0.378	0.452	0.371	0.431	0.492	0.494	0.344	0.303	0.350	0.385	0.321	0.442	0.207	0.308
	Avg	0.338	0.404	0.303	0.383	0.316	0.390	0.285	0.360	0.322	0.287	0.252	0.307	0.184	0.315	0.170	0.271
	96	0.320	0.387	0.242	0.330	0.490	0.477	0.191	0.290	0.346	0.326	0.212	0.270	0.134	0.269	0.185	0.300
Fully tune	192	0.342	0.403	0.270	0.352	0.376	0.423	0.196	0.287	0.355	0.327	0.236	0.286	0.173	0.305	0.204	0.316
+ GPT-Medium	336	0.348	0.409	0.284	0.367	0.530	0.501	0.253	0.335	0.379	0.345	0.298	0.331	0.311	0.421	0.224	0.334
+ Text prompt	720	0.368	0.433	0.424	0.479	0.375	0.429	0.361	0.423	OOM	OOM	OOM	OOM	0.333	0.457	0.206	0.307
	Avg	0.344	0.408	0.305	0.382	0.443	0.458	0.250	0.334	0.360	0.333	0.249	0.295	0.238	0.363	0.205	0.314
	96	0.296	0.371	0.234	0.328	0.309	0.381	0.150	0.263	0.299	0.278	0.175	0.248	0.073	0.204	0.145	0.249
Fully tune	192	0.318	0.386	0.273	0.355	0.301	0.381	0.190	0.293	0.311	0.278	0.212	0.279	0.129	0.271	0.164	0.266
+ Phi-2	336	0.337	0.402	0.311	0.389	0.346	0.419	0.283	0.381	0.323	0.290	0.282	0.345	0.233	0.374	0.179	0.281
+ TS prompt	720	0.372	0.445	0.317	0.407	0.404	0.461	0.439	0.484	0.347	0.305	0.354	0.382	0.404	0.501	0.218	0.319
	Avg	0.331	0.401	0.284	0.370	0.340	0.411	0.265	0.355	0.320	0.288	0.256	0.313	0.210	0.337	0.176	0.279
	96	0.296	0.371	0.234	0.328	0.309	0.381	0.150	0.263	0.299	0.278	0.175	0.248	0.073	0.204	0.145	0.249
Fully tune	192	0.319	0.385	0.269	0.355	0.309	0.383	0.188	0.295	0.307	0.275	0.212	0.283	0.134	0.281	0.161	0.262
+ Pi-2	336	0.337	0.402	0.311	0.389	0.346	0.419	0.283	0.381	0.323	0.290	0.282	0.345	0.233	0.374	0.179	0.281
+ Text prompt	720	0.356	0.430	0.359	0.442	0.392	0.454	0.383	0.451	0.345	0.302	0.345	0.377	0.561	0.606	0.212	0.315
	Avg	0.327	0.397	0.293	0.378	0.339	0.409	0.251	0.347	0.318	0.286	0.254	0.313	0.250	0.366	0.174	0.277
	96	0.362	0.419	0.273	0.363	0.589	0.533	0.225	0.332	0.428	0.396	0.224	0.293	0.129	0.274	0.227	0.333
LoRA-dim-16	192	0.394	0.444	0.312	0.397	0.582	0.531	0.259	0.361	0.502	0.437	0.339	0.280	0.200	0.345	0.257	0.358
+ GPT-Medium	336	0.403	0.457	0.321	0.413	0.560	0.532	0.293	0.392	0.547	0.457	0.320	0.369	0.266	0.409	0.291	0.386
+ TS prompt	720	0.444	0.499	0.366	0.451	0.576	0.547	0.355	0.436	0.660	0.519	0.369	0.406	0.457	0.532	0.406	0.479
	Avg	0.401	0.455	0.318	0.406	0.577	0.536	0.283	0.380	0.534	0.452	0.313	0.337	0.263	0.390	0.295	0.389
	96	0.365	0.422	0.270	0.361	0.596	0.593	0.222	0.329	0.438	0.408	0.223	0.294	0.117	0.259	0.233	0.341
LoRA-dim-32	192	0.401	0.449	0.314	0.398	0.594	0.537	0.261	0.363	0.503	0.443	0.281	0.340	0.204	0.346	0.259	0.361
+ GPT-Medium	336	0.403	0.457	0.321	0.413	0.563	0.533	0.294	0.393	0.547	0.459	0.321	0.370	0.267	0.410	0.294	0.390
+ TS prompt	720	0.444	0.498	0.367	0.452	0.572	0.545	0.357	0.437	0.647	0.513	0.369	0.406	0.454	0.530	0.399	0.473
	Avg	0.403	0.457	0.318	0.406	0.581	0.552	0.283	0.380	0.534	0.456	0.298	0.352	0.260	0.386	0.296	0.391
	96	0.377	0.431	0.284	0.376	0.603	0.538	0.239	0.348	0.462	0.423	0.244	0.313	0.154	0.302	0.242	0.348
LoRA-dim-16	192	0.394	0.445	0.313	0.400	0.578	0.530	0.263	0.367	0.511	0.441	0.284	0.344	0.203	0.351	0.262	0.363
+ GPT-Medium	336	0.412	0.465	0.325	0.417	0.571	0.538	0.299	0.397	0.567	0.471	0.323	0.373	0.267	0.413	0.308	0.402
+ Word prompt	720	0.448	0.501	0.368	0.452	0.582	0.550	0.357	0.437	0.672	0.526	0.370	0.407	0.452	0.529	0.416	0.486
	Avg	0.408	0.461	0.322	0.411	0.583	0.539	0.289	0.387	0.553	0.465	0.305	0.359	0.269	0.399	0.307	0.400
	96	0.365	0.423	0.276	0.367	0.590	0.533	0.230	0.337	0.449	0.410	0.234	0.305	0.133	0.277	0.237	0.343
LoRA-dim-32	192	0.400	0.449	0.311	0.397	0.572	0.527	0.261	0.364	0.515	0.447	0.284	0.345	0.207	0.352	0.265	0.366
+ GPT-Medium	336	0.410	0.463	0.324	0.416	0.570	0.538	0.299	0.398	0.563	0.469	0.323	0.373	0.268	0.414	0.305	0.399
+ Word prompt	720	0.447	0.500	0.368	0.453	0.589	0.553	0.359	0.459	0.664	0.522	0.370	0.407	0.453	0.530	0.414	0.486
	Avg	0.406	0.459	0.320	0.408	0.580	0.538	0.287	0.389	0.547	0.462	0.303	0.357	0.265	0.393	0.305	0.398

Table 16: Results of different backbones.

Datasets		**ETTh1**		**ETTh2**		**ETTm1**		**ETTm2**		**Traffic**		**Weather**		**Exchange**		**ECL**	
Metric		**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
	96	0.297	0.368	0.224	0.319	0.277	0.360	0.147	0.259	0.304	0.280	0.168	0.240	0.069	0.192	0.148	0.251
Fully tune	192	0.328	0.391	0.279	0.362	0.300	0.377	0.207	0.311	0.315	0.279	0.212	0.277	0.121	0.264	0.159	0.258
+ GPT-Large	336	0.347	0.407	0.365	0.420	0.322	0.397	0.284	0.364	0.324	0.287	0.291	0.341	0.356	0.461	0.174	0.276
+ TS prompt	720	0.358	0.427	0.430	0.478	0.358	0.421	0.436	0.467	0.348	0.303	0.370	0.388	0.424	0.525	0.207	0.308
	Avg	0.333	0.398	0.325	0.395	0.314	0.389	0.269	0.350	0.323	0.287	0.260	0.311	0.242	0.360	0.172	0.273
	96	0.307	0.377	0.235	0.326	0.285	0.369	0.156	0.266	0.305	0.278	0.172	0.242	0.065	0.186	0.145	0.247
Fully tune	192	0.329	0.391	0.283	0.365	0.319	0.393	0.203	0.307	0.313	0.274	0.218	0.278	0.115	0.248	0.159	0.259
+ GPT-Medium	336	0.346	0.405	0.320	0.401	0.378	0.425	0.255	0.349	0.326	0.287	0.276	0.329	0.206	0.339	0.180	0.284
+ TS prompt	720	0.370	0.441	0.378	0.456	0.464	0.477	0.342	0.417	0.346	0.301	0.339	0.373	0.409	0.487	0.215	0.317
	Avg	0.338	0.403	0.304	0.387	0.362	0.416	0.239	0.335	0.323	0.285	0.251	0.305	0.199	0.315	0.175	0.276
	96	0.317	0.383	0.240	0.334	0.431	0.443	0.178	0.285	0.315	0.293	0.188	0.257	0.083	0.208	0.161	0.265
Fully tune	192	0.355	0.413	0.285	0.370	0.479	0.482	0.221	0.323	0.352	0.336	0.238	0.304	0.132	0.277	0.213	0.311
+ GPT-Small	336	0.357	0.414	0.302	0.388	0.486	0.486	0.252	0.346	0.339	0.310	0.275	0.326	0.196	0.336	0.203	0.302
+ TS prompt	720	0.363	0.434	0.361	0.442	0.479	0.483	0.345	0.420	0.350	0.313	0.345	0.384	0.426	0.496	0.224	0.326
	Avg	0.348	0.411	0.297	0.384	0.469	0.473	0.249	0.343	0.339	0.313	0.261	0.317	0.209	0.329	0.200	0.301
	96	0.296	0.371	0.234	0.328	0.309	0.381	0.150	0.263	0.299	0.278	0.175	0.248	0.073	0.204	0.145	0.249
Fully tune	192	0.318	0.386	0.273	0.355	0.301	0.381	0.190	0.293	0.311	0.278	0.212	0.279	0.129	0.271	0.164	0.266
+ Phi-2	336	0.337	0.402	0.311	0.389	0.346	0.419	0.283	0.381	0.323	0.290	0.282	0.345	0.233	0.374	0.179	0.281
+ TS prompt	720	0.372	0.445	0.317	0.407	0.404	0.461	0.439	0.484	0.347	0.305	0.354	0.382	0.404	0.501	0.218	0.319
	Avg	0.331	0.401	0.284	0.370	0.340	0.411	0.265	0.355	0.320	0.288	0.256	0.313	0.210	0.337	0.176	0.279

Table 17: Results of different down-sampling ratios. Experiments with GPT-Medium as backbones, TS prompt, and fully tuning paradigm.

Datasets		**ETTh1**		**ETTh2**		**ETTm1**		**ETTm2**		**Traffic**		**Weather**		**ECL**	
Metric		**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
	96	0.4536	0.4787	0.3395	0.4097	0.7441	0.6068	0.3229	0.4055	0.6619	0.5636	0.3538	0.3979	0.6118	0.499
Downsample	192	0.4648	0.4935	0.386	0.4507	0.7659	0.6336	0.3797	0.4604	0.6644	0.5537	0.4106	0.454	0.6756	0.4915
Ratio = 40	336	0.6629	0.6167	0.5982	0.5916	1.1366	0.8212	0.6904	0.6455	1.0706	0.7785	0.7359	0.65	1.6485	0.7188
	720	1.0518	0.8133	0.8802	0.7588	1.8609	1.1304	1.2011	0.9073	1.7276	1.062	1.3176	0.9268	3.9988	1.0223
	Avg	0.343	0.408	0.307	0.390	0.353	0.412	0.234	0.331	0.344	0.314	0.253	0.306	0.210	0.330
	96	0.307	0.3767	0.2349	0.3263	0.285	0.3687	0.1555	0.2657	0.3053	0.2778	0.1717	0.2416	0.1447	0.2468
Downsample	192	0.3288	0.3908	0.2826	0.3649	0.3193	0.3925	0.2032	0.3069	0.3132	0.2744	0.2177	0.2782	0.1587	0.2588
Ratio = 20	336	0.346	0.405	0.3198	0.4005	0.3782	0.4254	0.2549	0.3492	0.3263	0.2869	0.2761	0.3287	0.1803	0.2835
	720	0.3704	0.4405	0.378	0.456	0.4638	0.4773	0.3422	0.4172	0.3456	0.301	0.3386	0.3732	0.2151	0.3167
	Avg	0.338	0.404	0.303	0.383	0.316	0.390	0.285	0.360	0.322	0.287	0.252	0.307	0.184	0.315
	96	0.2975	0.3698	0.2268	0.3175	0.2633	0.3462	0.1463	0.2583	0.2961	0.2626	0.2583	0.2247	0.1406	0.2411
Downsample	192	0.3293	0.3896	0.2848	0.3674	0.3286	0.3991	0.1995	0.3014	0.3089	2673	0.2151	0.2786	0.1568	0.2564
Ratio = 10	336	0.3461	0.4039	0.3097	0.3938	0.3593	0.4198	0.259	0.3505	0.3206	0.2785	0.2651	0.3155	0.1744	0.2747
	720	0.3676	0.4333	0.4101	0.4738	0.4096	0.4505	0.3632	0.4242	0.3428	0.2983	0.3462	0.38	0.2099	0.3101
	Avg	0.335	0.402	0.321	0.392	0.341	0.399	0.235	0.332	0.312	0.274	0.252	0.308	0.232	0.360