A Language Model-Guided Framework for Mining Time Series with Distributional Shifts

Haibei Zhu, Yousef El-Laham, Elizabeth Fons, Svitlana Vyetrenko
J.P. Morgan AI Research
{haibei.zhu, yousef.el-laham, elizabeth.fons,
svitlana.s.vyetrenko}@jpmchase.com

Abstract

Effective utilization of time series data is often constrained by the scarcity of data quantity that reflects complex dynamics, especially under distributional shifts. This paper presents an approach that utilizes large language models and data source software interfaces to collect time series datasets. This approach enlarges the data quantity and diversity when the original data is limited or lacks essential properties. We demonstrate the effectiveness of the collected datasets through utility examples and show how time series forecasting foundation models fine-tuned on these datasets achieve better performance than those without fine-tuning.

1 Introduction

Time series analysis is important in various domains, including healthcare, finance, and environmental science [1, 2, 3, 4]. Both recent advanced machine learning models and traditional statistical analyses rely heavily on the availability of time series datasets that capture the underlying dynamics of the systems to perform downstream tasks [5, 6]. However, the scarcity of high-quality time series data, especially those reflecting distributional shifts, brings significant challenges. Also, privacy concerns and data accessibility issues further restrict the availability of real-world datasets [7]. Time series distributional shifts caused by special events alter the statistical properties of the data. Such statistical properties exacerbate the data scarcity issue.

One emerging solution for addressing the data scarcity and distributional shift issues is exploring and utilizing alternative time series datasets [8, 9]. Alternative datasets can be generated via synthetic data generation techniques, such as generative adversarial networks (GANs) [10]. These synthetic datasets aim to augment real datasets to enhance the model's robustness and allow models to generalize to unseen data. Alternative datasets can also be real-world data that share similar properties but are sourced from other domains. Understanding that recent advanced large language models (LLMs), like GPT-4 [11] and Gemini [12], have demonstrated the ability to understand human language and provide empirical knowledge [13], we leverage them to identify relevant data sources and retrieve data samples to construct alternative datasets.

This paper proposes an approach that leverages LLMs and data source application programming interfaces (APIs) to explore and collect time series datasets. We leverage the empirical knowledge provided by LLMs to optimize the data collection process. This approach can enhance the quantity and diversity of the time series datasets and collect data that meet specific requirements, such as distributional shifts. Our main contributions are: 1) We introduce a novel framework that leverages LLMs and data APIs to collect alternative time series datasets exhibiting distributional shifts efficiently, 2) We demonstrate the effectiveness of this approach by curating a diverse dataset collection across various domains, and 3) We showcase the utility of these datasets by fine-tuning time series forecasting foundation models and achieving comparable performance to models without fine-tuning, even in the presence of distributional shifts.

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

2 Dataset Mining Pipeline

The proposed method utilizes LLMs and data source APIs to collect alternative time series datasets that exhibit distributional shifts. This pipeline is structured into the following steps: 1) data source exploration, 2) data collection, 3) data pruning, 4) data augmentation, and 5) data evaluation.

Figure 1: Prompt example for LLMs to explore and collect time series data with distributional shifts.

Prompt to list potential data sources and APIs

I want to use general-purpose LLMs such as GPT4 to assist in constructing time series datasets, focusing on datasets that suffer from distribution shifts. For example, S&P500 data suffered a distribution shift during COVID-19. I want an LLM to generate query terms and data sources to build a heterogeneous time series dataset from different domains with distributional shifts. Please provide a list of open time series datasets from different contexts that can be used to query and extract time series with distribution shifts. Provide the list in latex tabular format with the following columns: Domain, Name of dataset, Description, API (yes/no), Link, Licence.

2.1 Data Exploration and Collection

Our pipeline's initial stage involves identifying and selecting suitable data sources. This process leverages LLMs as extensive knowledge repositories. We craft specific prompts (Figure 1) to elicit information regarding publicly available time series datasets, including their domains, descriptions, licenses, API availability, and their potential to exhibit distributional shifts due to significant events (e.g., economic crises, pandemics, policy changes). Four datasets, including Yahoo Finance, Fred Economics, EIA Energy, and Google Search Trend, have been identified by LLMs with detailed data sample information. We focus on these datasets to collect samples with distributional shifts.

For identified datasets, we engage the LLM in two steps. First, we provide the LLM with the dataset's API documentation or a structure description. The LLM then generates code snippets tailored to interact with the specific API. For example, the generated code may include logic for handling rate limits by pacing requests or incorporating retry mechanisms. Second, we harness the LLM's understanding of historical events and their potential impact on time series data to construct queries for each API, each containing a unique identifier for the specific time series within the dataset, the start and end dates for the period of interest, and a comment justifying the selection. This justification explains why this particular time series and period are hypothesized to exhibit distributional shifts.

2.2 Data Pruning and Augmentation

After acquiring the time series in the data collection step, we want to discard samples we suspect will not be useful in our downstream use cases. The data pruning in our pipeline takes all samples collected in the data collection step as input. It outputs the subset of time series whose statistical properties satisfy a pre-defined set of requirements. In the use cases in this work, we require that our collected data samples exhibit distributional shifts. Therefore, the data pruning in this work utilizes offline change point detection (CPD) as a means to pruning the time series samples. Change point detection algorithms identify points in time series samples where the statistical properties, such as mean and variance, significantly change. In this work, we utilize the Ruptures Python library [14]. By leveraging this library, we ensure the resulting data samples contain time series value distributional shifts for downstream analyses.

After data pruning, our approach may result in a smaller dataset than the entire database. Therefore, data augmentation becomes essential to increasing the quantity and diversity of our collected data, improving the robustness and generalizability of downstream models. We apply three data augmentation methods with a focus on warping the time dimension to the pruned data samples: *time warping* [15], *window warping* [16] and *window slicing* [16]. These methods create new samples that retain the statistical properties of the original data while introducing variations. They primarily affect the lower frequencies of data samples, corresponding to the trend and seasonality components, which are often more relevant for capturing the underlying dynamics and distributional shifts in the data.

Table 1: Collected Datasets

Name	Domain	Description	Length		Sample Quantity		
			Min	Max	Original	After Pruning	After Augmentation
FRED	Economics & Finance	Macroeconomic and financial time series data	25	457	241	77	2310
World Cup search trends	Google search	Time series data of the popularity of World Cup 2022-related search queries on Google	120	120	173	67	2010
EIA Daily	Energy	Time series data related to electricity generation, demand, etc.	32	254	3750	1194	35820
Yahoo Finance	Finance	Financial market data, including stock prices, commodities, and foreign exchange	41	252	369	91	2730
COVID search trends	Google search	Time series data of the popularity of COVID-related search queries on Google	120	120	144	68	2040

2.3 Collected Datasets

The resulting datasets from this pipeline include time series data from various domains, exhibiting distributional shifts in data samples. Table 1 summarizes the collected datasets, detailing their domains, descriptions, lengths, and sample quantities at each pipeline stage.

3 Utility Examples

We demonstrate the effectiveness of the collected datasets by fine-tuning time series forecasting foundation models, Lag-Llama [17] and Chronos [18], and comparing their zero-shot prediction performance versus after fine-tuning. The collected datasets mentioned in Section 2 are utilized for foundation model fine-tuning and evaluation. Specifically, the FRED, World Cup search trends, and EIA Daily energy datasets are used for fine-tuning as in-sample datasets. Once the in-sample datasets have been collected and pruned, we split them into training and testing sets following the commonly used 80-20 ratio. The training sets are augmented to let the models learn diverse patterns and scenarios in the data.

We evaluate the performance of the foundation models in predictions in both zero-shot and after fine-tuning. The zero-shot scenario evaluates the model's ability to generalize from its existing knowledge. The fine-tuning process will first train the models on the collected datasets, which will help them adapt to distributional shifts. The evaluation metrics for model prediction performance include the average mean square error (MSE), the variance of MSE across prediction samples, and the mean absolute error (MAE) coverage. The MSE measures the average squared difference between the predicted and actual time series. Lower MSE values indicate better prediction performance. The variance of MSE captures the variability of the prediction errors, showing the consistency of the prediction outcome. The MAE coverage measures the mean absolute error between the observed coverage and the target quantile levels, with lower MAE coverage values indicating more accurate and reliable prediction intervals.

As shown in Table 2, we present the prediction performance of the Lag-Llama and Chronos models in both zero-shot and fine-tuned scenarios. We fine-tuned the models using the three in-sample datasets (FRED, World Cup search trends, and EIA Daily) and evaluated them on all five collected datasets. These results indicate a significant improvement in in-sample datasets, with both MSE and variance measures lower than those zero-shot measures. However, improvement in the MAE coverage measures is limited. While the degree of improvement varies across different models, such improvements support the idea that models can be effectively fine-tuned to adapt to time series with distributional shifts. The testing results on the two out-sample datasets (Yahoo and COVID search trends) demonstrate the generalizability of the fine-tuned model on distributional shift data. Although the improvements are more modest than the in-sample datasets, they still represent a notable enhancement over the zero-shot performance. The fine-tuning process significantly enhanced the performance of both models. The results demonstrate the utility of the collected distributional shift datasets in improving model accuracy and consistency.

Table 2: Model prediction results on in-sample datasets.

Model	Model size	Evaluation	Metrics	In-sample datasets			Out-sample datasets	
				FRED	World Cup	EIA	Yahoo	COVID
Lag-Llama	(2.5M)	Zero-shot	MSE	0.1959	0.0126	0.1147	0.0613	0.0496
			Variance	0.0110	0.0003	0.0082	0.0060	0.0020
			MAE coverage	0.2646	0.4643	0.3575	0.3346	0.3588
		After fine-tuning	MSE	0.0779	0.0105	0.0428	0.0488	0.0450
			Variance	0.0015	0.0003	0.0009	0.0021	0.0032
			MAE coverage	0.2910	0.3556	0.3860	0.2584	0.3611
Chronos	Tiny (8M)	Zero-shot	MSE	0.1403	0.0095	0.0781	0.0508	0.0514
			Variance	0.0060	0.0002	0.0045	0.0041	0.0041
			MAE coverage	0.2330	0.4857	0.2644	0.2427	0.3145
		After fine-tuning	MSE	0.0956	0.0054	0.0244	0.0445	0.0420
			Variance	0.0032	0.0002	0.0005	0.0030	0.0028
			MAE coverage	0.2667	0.4036	0.3582	0.2804	0.2908
	Mini (20M)	Zero-shot	MSE	0.1409	0.0108	0.0785	0.0483	0.0589
			Variance	0.0064	0.0002	0.0049	0.0040	0.0061
			MAE coverage	0.2330	0.4857	0.2664	0.2425	0.3548
		After fine-tuning	MSE	0.1039	0.0054	0.0194	0.0460	0.0421
			Variance	0.0048	0.0002	0.0004	0.0034	0.0028
			MAE coverage	0.2719	0.3722	0.3634	0.2825	0.2965
	Small (46M)	Zero-shot	MSE	0.1428	0.0113	0.0764	0.0519	0.0641
			Variance	0.0070	0.0002	0.0043	0.0045	0.0062
			MAE coverage	0.2365	0.4893	0.2683	0.2324	0.3388
		After fine-tuning	MSE	0.1013	0.0059	0.0158	0.0490	0.0392
			Variance	0.0036	0.0002	0.0004	0.0036	0.0028
			MAE coverage	0.2858	0.3694	0.3605	0.2712	0.3116
	Base (200M)	Zero-shot	MSE	0.1442	0.0173	0.0753	0.0474	0.0681
			Variance	0.0049	0.0003	0.0046	0.0039	0.0069
			MAE coverage	0.2458	0.4821	0.2732	0.2366	0.3440
		After fine-tuning	MSE	0.0937	0.0054	0.0163	0.0550	0.0374
			Variance	0.0021	0.0002	0.0008	0.0035	0.0027
			MAE coverage	0.2625	0.3750	0.3468	0.2652	0.3165

4 Discussion and Conclusion

The proposed approach for creating alternative datasets using LLMs and data source APIs demonstrates an advancement in addressing the challenges associated with time series analysis, particularly under data scarcity and distributional shifts. This methodology leverages the capabilities of LLMs to identify and retrieve time series data. This pipeline can be adaptive across various domains, where the availability of high-quality datasets can significantly affect downstream modeling. By explicitly targeting datasets that reflect distributional shifts, the proposed approach ensures that models trained on collected datasets are better equipped to handle scenarios where distributional shifts occur. Another critical aspect of this approach is the integration of data pruning and augmentation. Data pruning ensures the collected time series samples satisfy the requirements of specific statistical properties. Data augmentation enhances the diversity and quantity of collected datasets.

Variations in data sources, such as differences in sampling intervals and lengths, can introduce noise and biases that may affect the performance of downstream tasks. Ensuring the reliability of the collected data requires further validation and quality control measures. Our approach can be adapted to various time resolutions to collect time series data for downstream models that can analyze datasets with different temporal properties. Furthermore, the potential applications of this pipeline can extend beyond various domains and scenarios, not limited to distributional shift time series datasets. In summary, the proposed pipeline leverages the extensive capabilities of LLMs to explore and identify relevant datasets and collect data samples with distributional shifts. The experiments conducted with time series forecasting foundation models demonstrate the effectiveness of the collected datasets in enhancing model performance and generalization capability.

Acknowledgments and Disclosure of Funding

This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan") and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

[1] Robert B Penfold and Fang Zhang. Use of interrupted time series analysis in evaluating health care quality improvements. *Academic Pediatrics*, 13(6):S38–S44, 2013.

[2] Ruey S Tsay. *Analysis of Financial Time Series*. John Wiley & Sons, 2005.

[3] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020.

[4] C James Taylor, Diego J Pedregal, Peter C Young, and Wlodek Tych. Environmental time series analysis and forecasting with the captain toolbox. *Environmental Modelling & Software*, 22(6):797–814, 2007.

[5] Ms Aayushi Bansal, Dr Rewa Sharma, and Dr Mamta Kathuria. A systematic review on data scarcity problem in deep learning: Solution and applications. *ACM Computing Surveys (CSUR)*, 54(10s):1–29, 2022.

[6] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[7] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, Ahmed Shihab Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H Al-Timemy, et al. A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10(1):46, 2023.

[8] Bilgi Yilmaz and Ralf Korn. Synthetic demand data generation for individual electricity consumers: Generative adversarial networks (gans). *Energy and AI*, 9:100161, 2022.

[9] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. In *2018 IEEE International Conference on Big Data*, pages 1367–1376. IEEE, 2018.

[10] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report, 2023.

[12] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: A family of highly capable multimodal models, 2023.

[13] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[14] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

[15] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 216–220, 2017.

[16] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2016.

[17] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024.

[18] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

A Code and Dataset Releasing

We have prepared an anonymous repository containing the code and the dataset, which can be accessed at the following link: `https://anonymous.4open.science/r/alternative_time_series-44C8/`

B Experiment Information

In this section, we provide detailed information for the Experiment Section of the main paper, including the dataset information, model fine-tuning details, and visualizations of data samples.

Table 3: Dataset and model licenses

	Name	Domain	Description	License
Datasets	FRED	Economics & Finance	Macroeconomic and financial time series data	Link
	World Cup search trends	Google search	Time series data of the popularity of World Cup 2022-related search queries on Google	MIT licensed
	EIA Daily	Energy	Time series data related to electricity generation, demand, etc.	Link
	Yahoo Finance	Finance	Financial market data, including stock prices, commodities, and foreign exchange	Link
	COVID search trends	Google search	Time series data of the popularity of COVID-related search queries on Google	MIT licensed
Models	Lag-Llama	Time series	An open-source foundation model for time series forecasting	Link
	Chronos	Time series	A family of pretrained time series forecasting models based on language model architectures	Link

B.1 Datasets

Table 3 lists the licenses of data source APIs and time series foundation models. All data source APIs and both foundation models are publicly accessible.

Here, we present visualizations for time series data samples after the data pruning process from the five datasets we used for the main paper. As shown in Figure 2, 3, 4, 5, and 6, nine random time series samples have been plotted for each dataset. The X-axis shows the time step for each sample, and the Y-axis shows the original values of the sample. Each sub-plot in these figures illustrates distributional shifts determined by the change point detection algorithm mentioned in the main paper. Different colors, either blue or red, indicate specific time series value distributions, and transitions between colors indicate the presence of change points or distributional shifts. In general, distributional shifts indicated by the change points in data samples align with visual observations.

Figure 2: Time series data samples with distributional shifts from the FRED dataset.

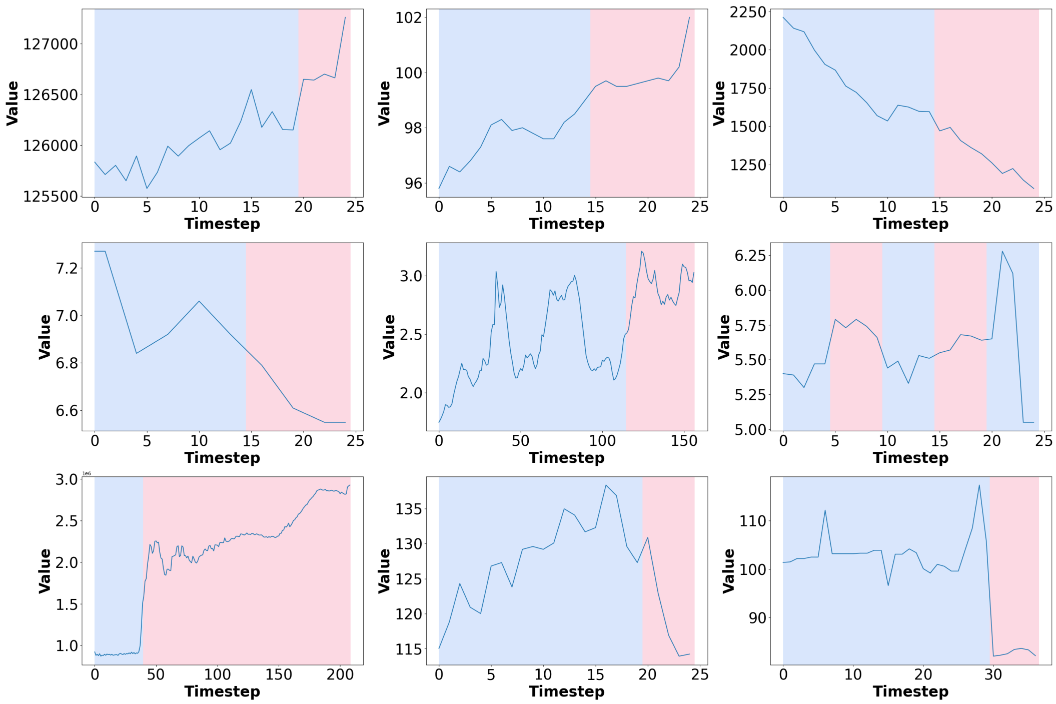


Figure 3: Time series data samples with distributional shifts from the World Cup search trend dataset.

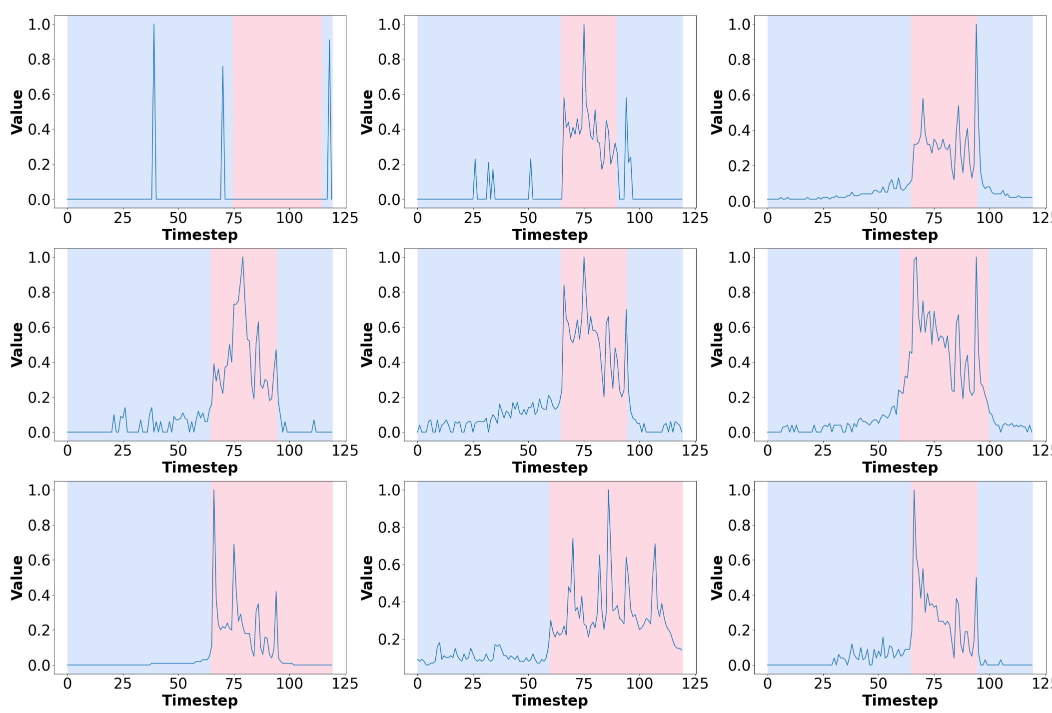


Figure 4: Time series data samples with distributional shifts from the EIA Daily Energy dataset.

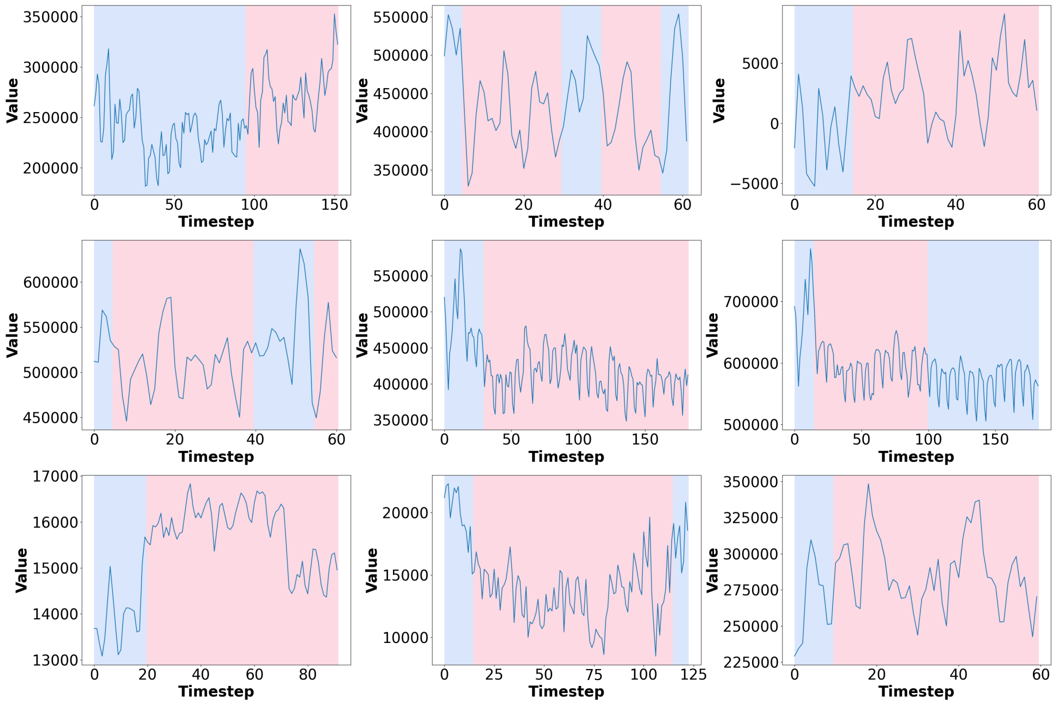


Figure 5: Time series data samples with distributional shifts from the Yahoo Finance dataset.

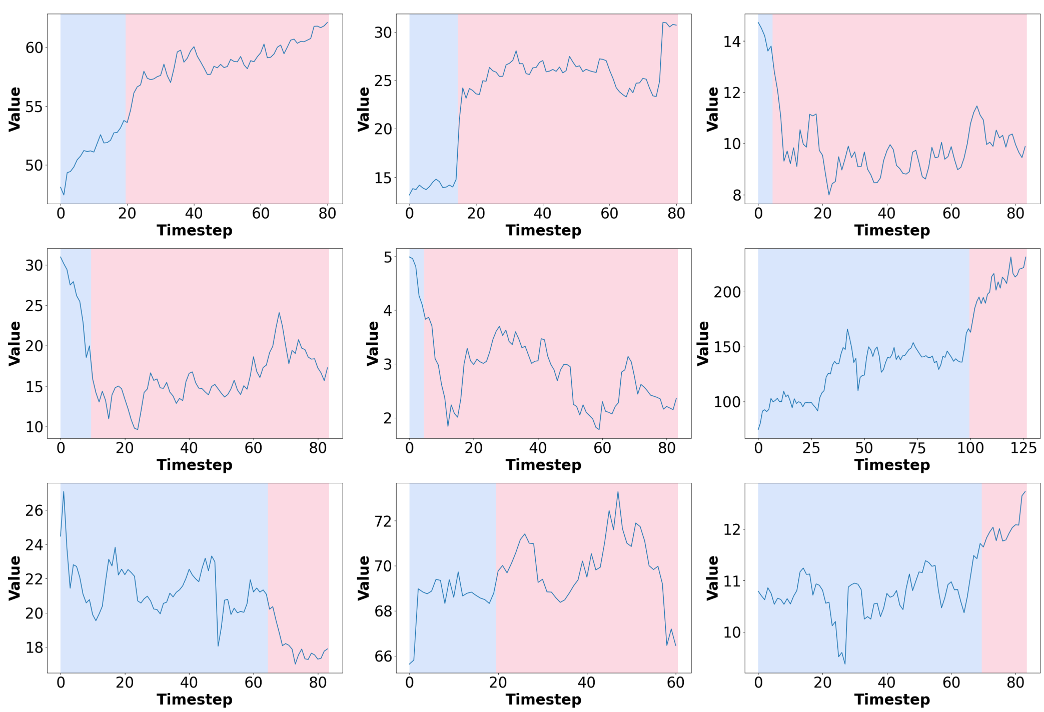
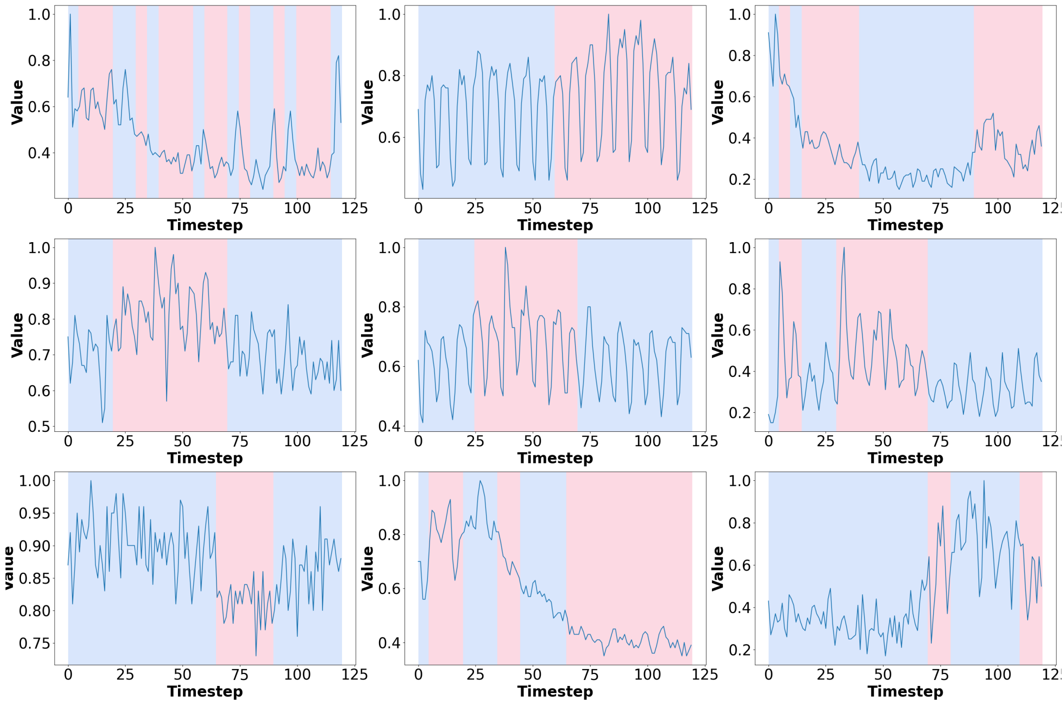


Figure 6: Time series data samples with distributional shifts from the COVID search trend dataset.



B.2 Model Fine-tuning Configuration

Here, we list detailed configurations for fine-tuning the Lag-Llama and Chronos foundation models.

Table 4: Training and Testing Data Samples

Dataset type	Dataset name	Sample quantity		Training and testing split	
		After pruning	After augmentation	Number of samples for fine-tuning	Number of samples for testing
In-sample datasets	FRED	77	2310	1848	16
	World Cup search trends	67	2010	1608	14
	EIA Daily	3750	35820	28656	239
Out-sample datasets	Yahoo Finance	91	N/A	N/A	91
	COVID search trends	68	N/A	N/A	68

Table 4 illustrates the data sample quantity from each dataset for fine-tuning and testing time series foundation models. Once the data pruning processing is completed, we augmented the three in-sample datasets by utilizing the three augmentation methods mentioned in the main paper with ten randomized iterations. Here, we define the in-sample datasets as the datasets used for fine-tuning and testing the foundation models, and out-sample datasets are those only used for testing models' prediction performance. Thus, the quantity of augmented data samples is 30 times of the pruned samples. Given the augmented datasets, we split them following the commonly-used 80-20 rule that we randomly selected 80% of the data samples as the training or fine-tuning data. In testing, we utilize the original data samples without augmentation for both in and out-sample datasets.

We fine-tuned foundation models and ran prediction tasks on an AWS g4dn.2xlarge instance, which has 8 vCPUs, 32 GB memory, and one NVIDIA T4 Tensor Core GPU.

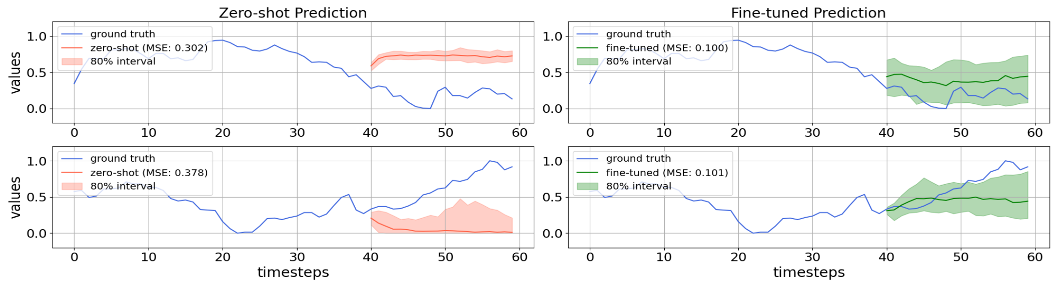
For fine-tuning the Lag-Llama model, we followed the configurations in the default training function provided by the developers. Specifically, we set the prediction length to 20 and the context length to 100. So that the model takes time series samples with various lengths and utilizes the first part of a sample (the segment from the last 120th timestamp to the last 20th timestamp if the sample is longer than 120 timesteps, or the segment from the beginning of the sample to the last 20th timestamp if the sample is shorter than 120 timesteps) as the context or the forecasting input and the second part (the last 20 timesteps of the sample) as the forecasting ground truth to fine-tune the model or evaluate prediction performance. We used the default learning rate of $1e^{-4}$, batch size of 64, and set the training epoch to $10k$.

We also followed the default settings, which are constructed in YAML files, to fine-tune the Chronos model with four different sizes (tiny, mini, small, and base). Similar to the Lag-Llama fine-tuning setting, we configured the context length to 100 and the prediction length to 20. We increased the training steps to $100k$ and kept the training rate to $1e^{-3}$ and batch size to 32. We utilized the tokenizer provided by the developers and left the number of tokens to the default value of 4096.

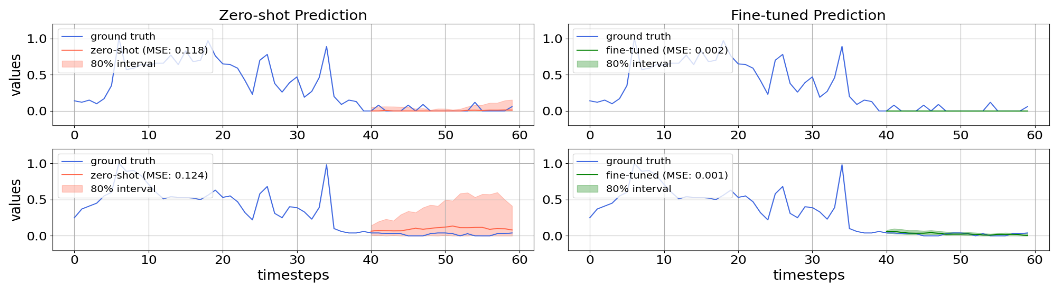
B.3 Experiment Result Visualizations

The experiment presented in the main paper compares foundation model prediction performance between zero-shot and after fine-tuning. Here, we present visualization examples of predicted results in both zero-shot and after fine-tuning scenarios. As shown in Figure 7, 8, 9, 10, and 11, left side of these figures present the zero-shot prediction examples, while right side show the prediction from the fine-tuned models. Blue lines in all figures indicate the normalized ground truth time series value. The colored areas indicate the 80% prediction interval across 100 prediction runs.

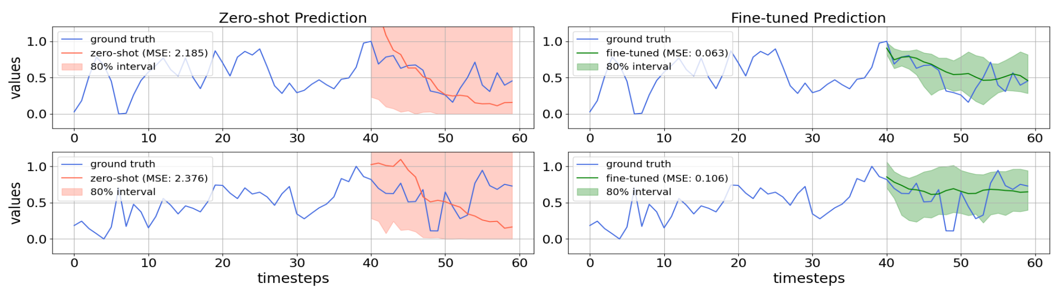
Figure 7: Lag-Llama model prediction examples.



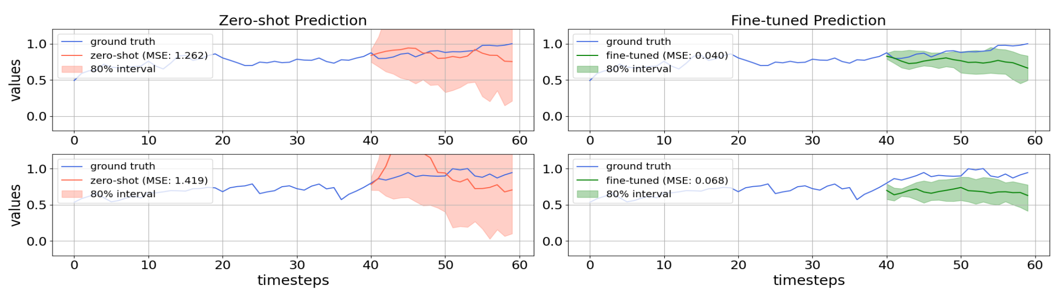
(a) Prediction examples on the FRED dataset.



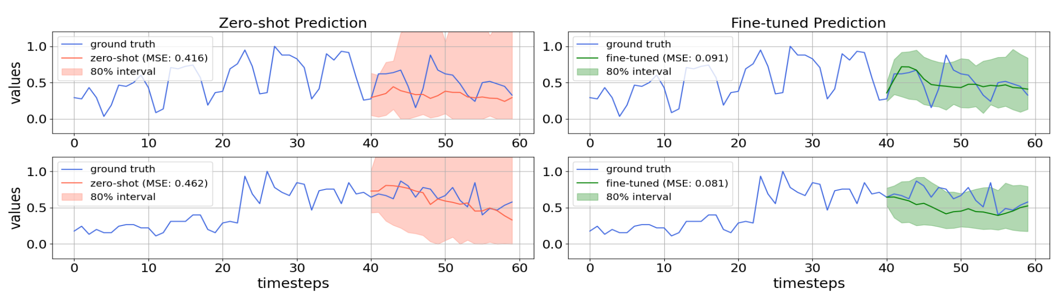
(b) Prediction examples on the World Cup Trend dataset.



(c) Prediction examples on the EIA Daily Energy dataset.

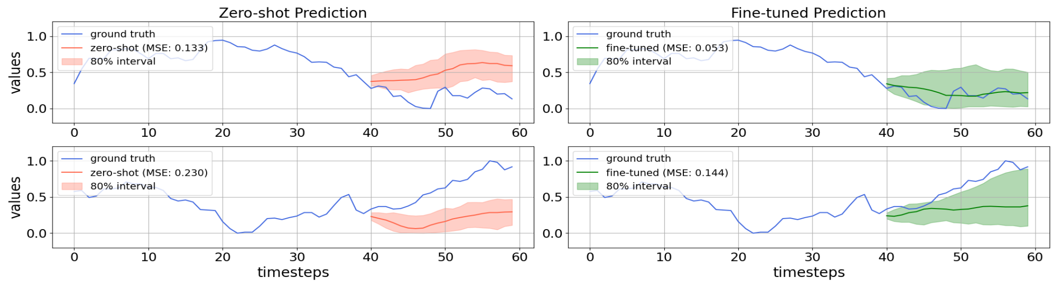


(d) Prediction examples on the Yahoo dataset.

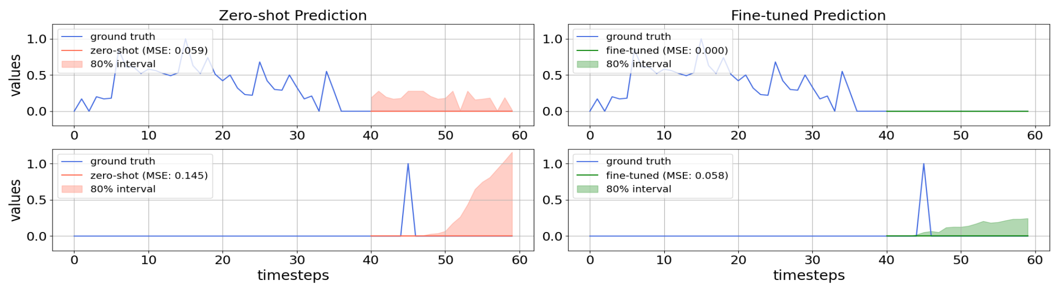


(e) Prediction examples on the COVID Trend dataset.

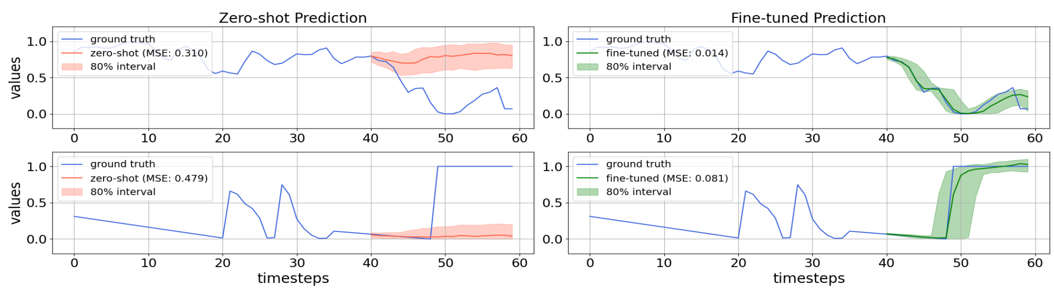
Figure 8: Chronos Tiny model prediction examples.



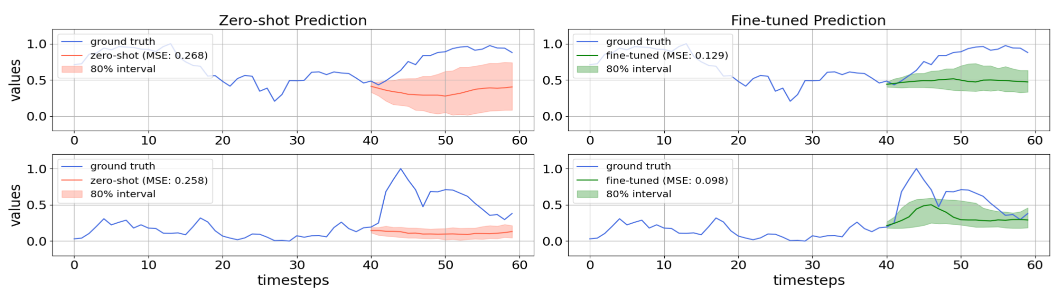
(a) Prediction examples on the FRED dataset.



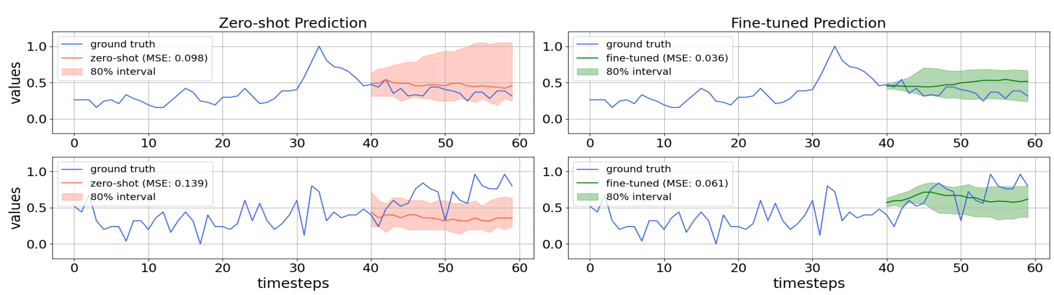
(b) Prediction examples on the World Cup Trend dataset.



(c) Prediction examples on the EIA Daily Energy dataset.

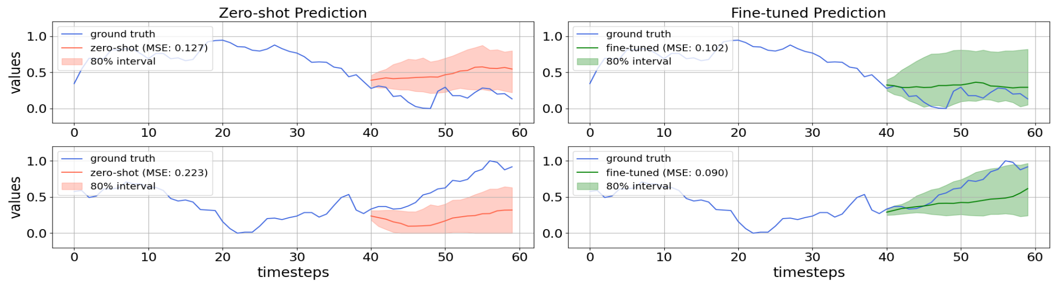


(d) Prediction examples on the Yahoo dataset.

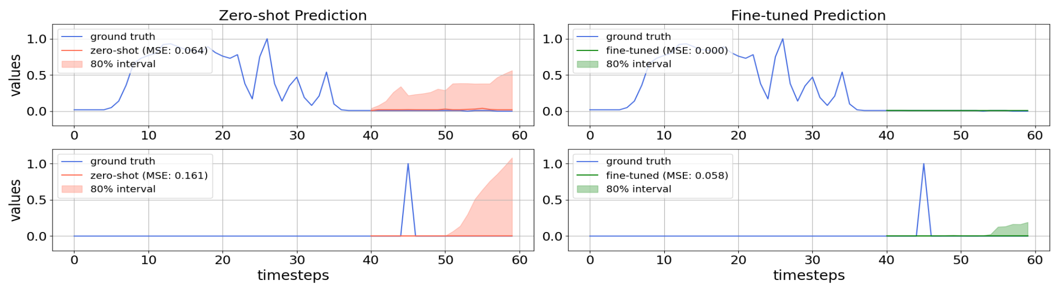


(e) Prediction examples on the COVID Trend dataset.

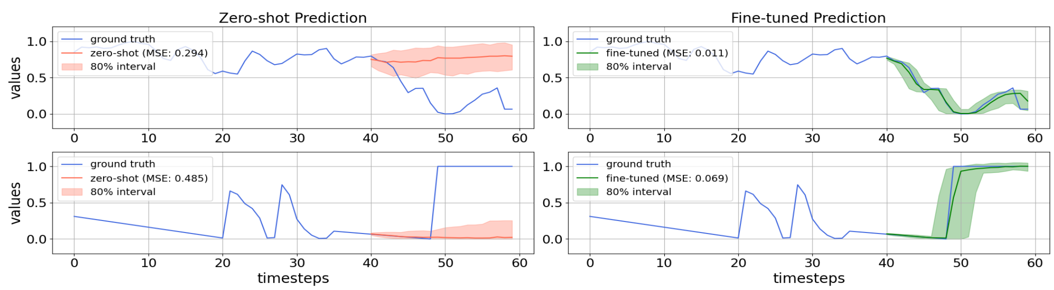
Figure 9: Chronos Mini model prediction examples.



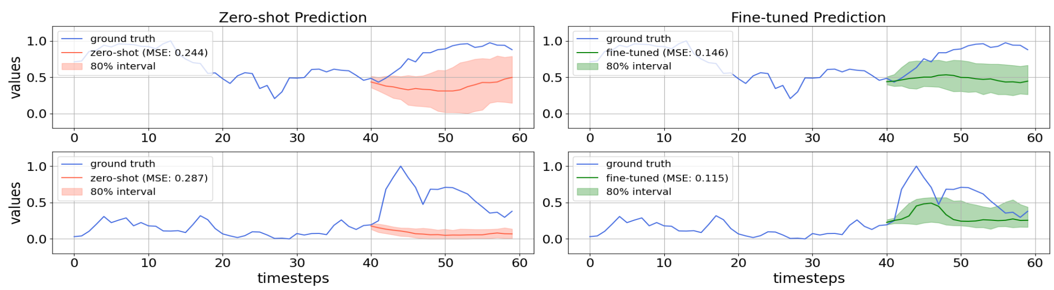
(a) Prediction examples on the FRED dataset.



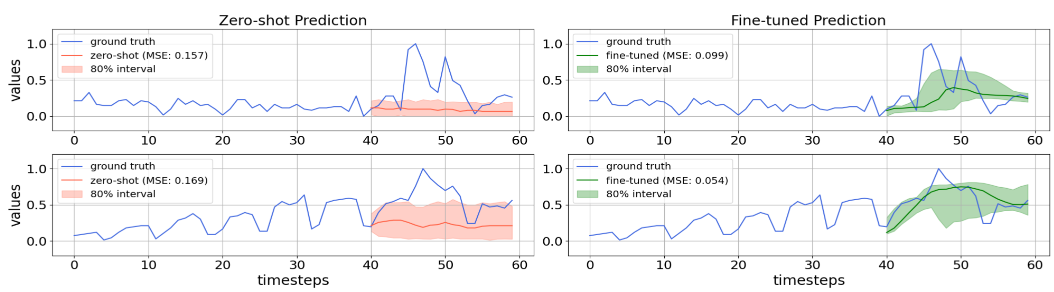
(b) Prediction examples on the World Cup Trend dataset.



(c) Prediction examples on the EIA Daily Energy dataset.

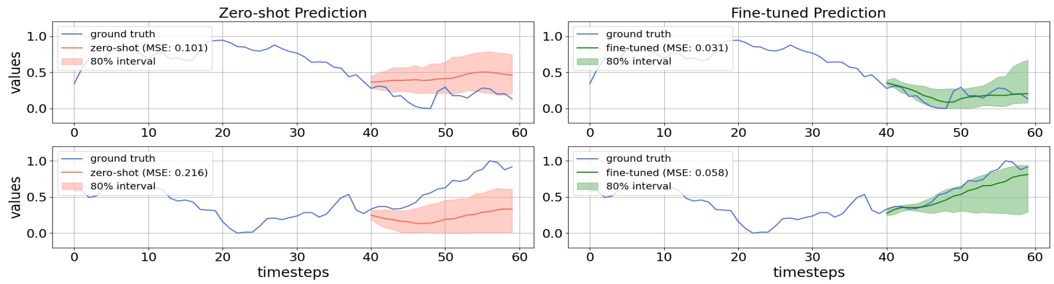


(d) Prediction examples on the Yahoo dataset.

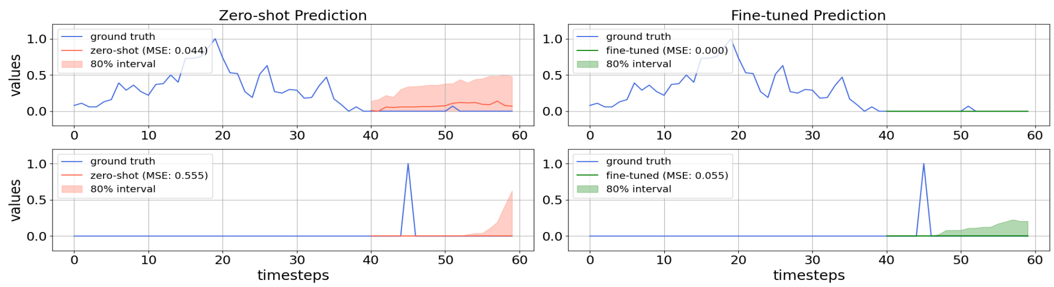


(e) Prediction examples on the COVID Trend dataset.

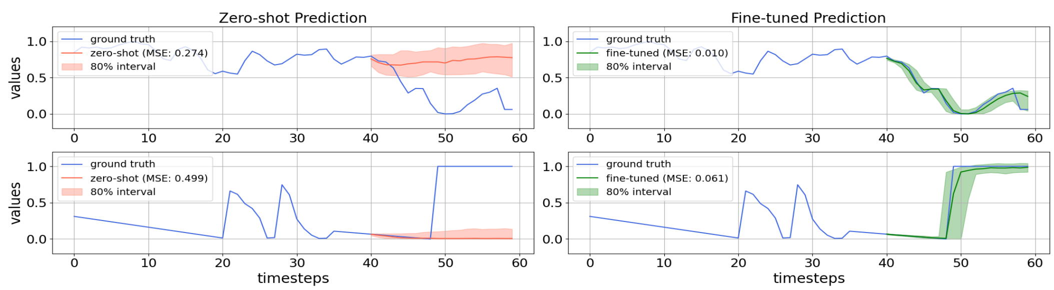
Figure 10: Chronos Small model prediction examples.



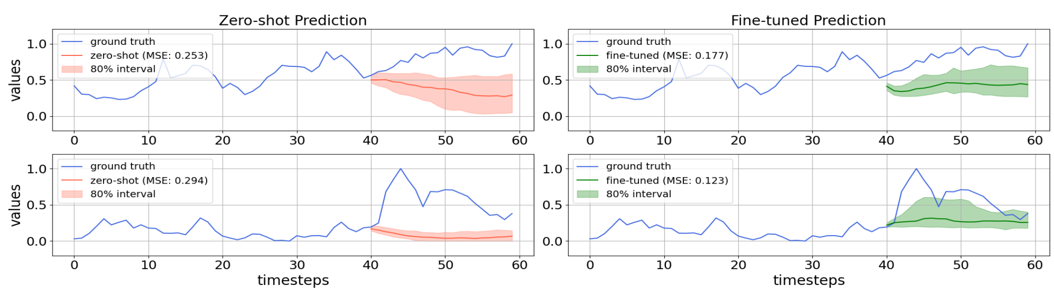
(a) Prediction examples on the FRED dataset.



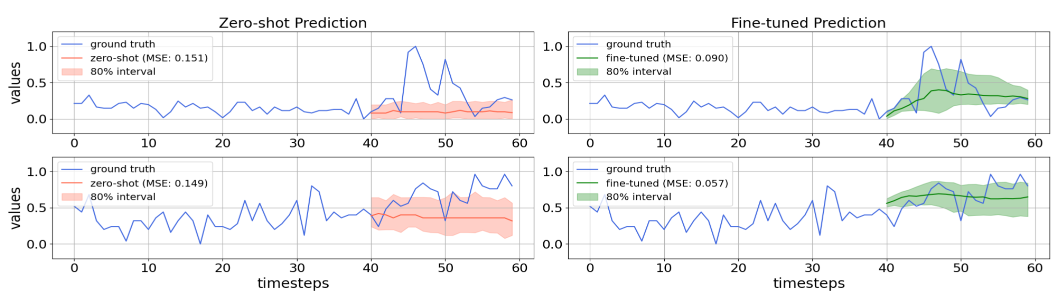
(b) Prediction examples on the World Cup Trend dataset.



(c) Prediction examples on the EIA Daily Energy dataset.

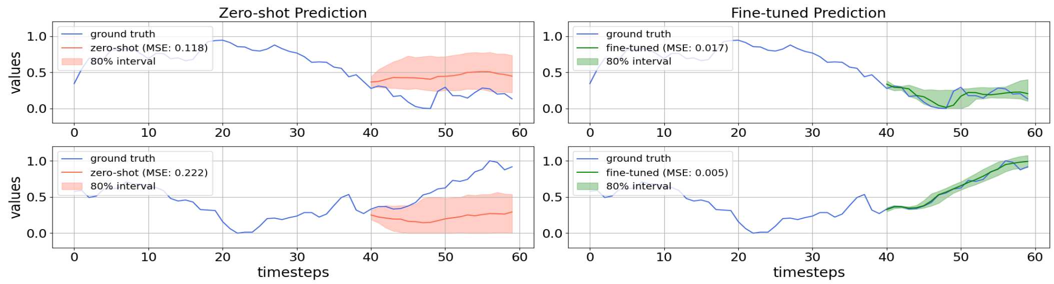


(d) Prediction examples on the Yahoo dataset.

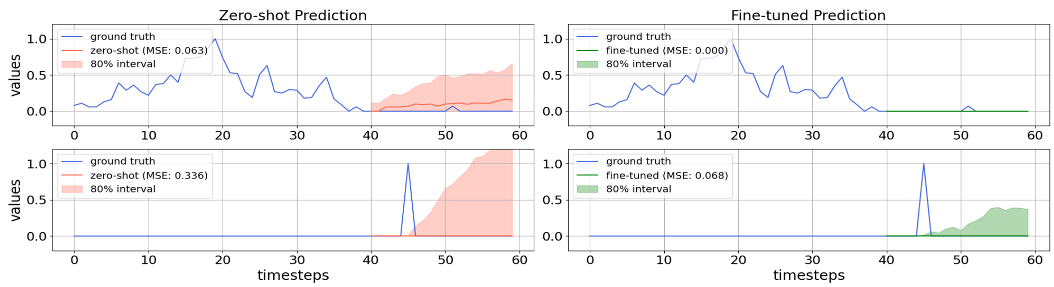


(e) Prediction examples on the COVID Trend dataset.

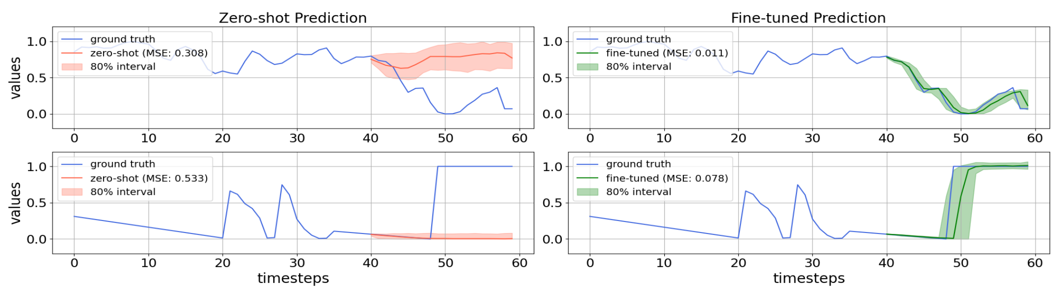
Figure 11: Chronos Base model prediction examples.



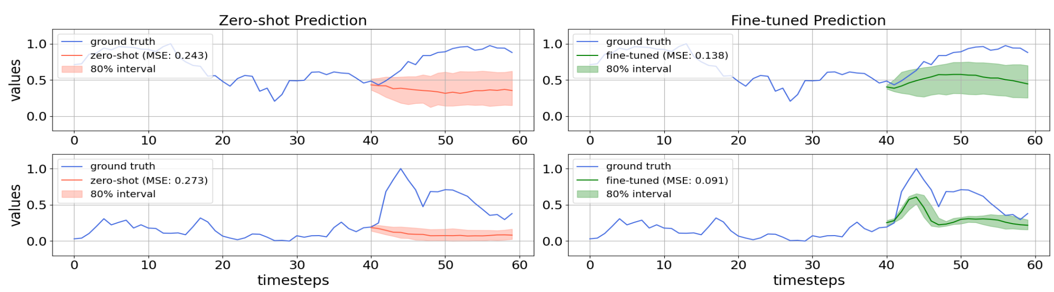
(a) Prediction examples on the FRED dataset.



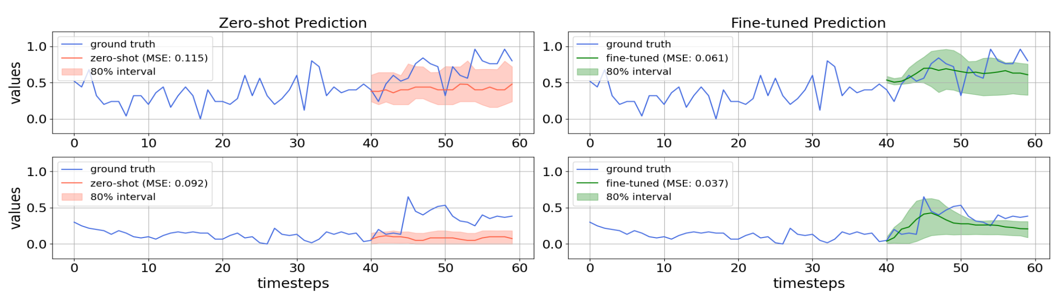
(b) Prediction examples on the World Cup Trend dataset.



(c) Prediction examples on the EIA Daily Energy dataset.



(d) Prediction examples on the Yahoo dataset.



(e) Prediction examples on the COVID Trend dataset.

C Ablation Studies with Synthetic Data

Here, we utilize the Ornstein-Uhlenbeck (OU) process to create synthetic time series data samples with specific distributional shifts for an ablation study. In this ablation study, we focus on evaluating 1) whether adding synthetic OU data would help with model fine-tuning and 2) whether different quantities of synthetic data would significantly improve model fine-tuning.

C.1 Synthetic Data Generation

We generated synthetic data using the Ornstein-Uhlenbeck (OU) process, which is a type of continuous-time stochastic process. OU process is often used to model mean-reverting behavior in time series data. Over time, the values of the OU process tend to drift towards a long-term mean. This process is characterized by three key parameters: the mean, which is the long-term average value to which the process reverts; the scale, which is the volatility or standard deviation of the process, determining the extent of fluctuations around the mean; and the reversion rate, which is the speed at which the process reverts to the mean. A higher reversion rate indicates a quicker reversion.

We defined two types of OU processes with different parameters for generating synthetic data: Fast Mean and Fast Variance. The Fast Mean configuration has a relatively high reversion rate (0.1), meaning that the process quickly reverts to its mean. The Fast Variance configuration also has a high reversion rate but with a higher scale (6), leading to larger fluctuations around the mean while still reverting quickly.

We configured change points while generating time series samples to include distributional shifts. For each time step, the function computes the new value of the time series based on the previous value, the mean reversion, and a random fluctuation. If a change point is reached, the parameters of the OU process are updated to the new values (mean, scale, reversion rate) specified after the change point.

C.2 Ablation Study Results

The results of the ablation study are summarized in Tables 5 and 6. The objective of this study was to evaluate the impact of adding synthetic data generated using the OU process on the fine-tuning of time series foundation models. We focus on whether the addition of synthetic data improves prediction performance and how different quantities of synthetic data affect the performance.

Experiment results indicate that adding synthetic OU data does not significantly enhance the prediction performance of the models, according to the evaluation metrics. For the Lag-Llama model, it was observed that after fine-tuning with different quantities of synthetic data (ranging from 10K to 200K samples), the MSE, variance, and MAE coverage did not show consistent improvement. Similarly, the Chronos models with different model sizes showed negligible improvements in prediction performance after fine-tuning with synthetic data. One possible reason is that the generated synthetic time series data through OU processes do not share similar patterns and dynamics as in the testing datasets. Thus, adding the synthetic OU time series to the fine-tuning cannot improve the models' prediction performance.

Table 5: Ablation study of adding different quantity of synthetic data – summary 1.

Model (model size)	Evaluation type	Synthetic data added	Metrics	In-sample datasets			Out-sample datasets	
				FRED	World Cup	EIA	Yahoo	COVID
Lag-Llama (2.5M)	Zero-shot	N/A	MSE	0.1959	0.0126	0.1147	0.0613	0.0496
			Variance	0.0110	0.0003	0.0082	0.0060	0.0020
			MAE coverage	0.2646	0.4643	0.3575	0.3346	0.3588
	After fine-tuning	0	MSE	0.0779	0.0105	0.0428	0.0488	0.0450
			Variance	0.0015	0.0003	0.0009	0.0021	0.0032
			MAE coverage	0.2910	0.3556	0.3860	0.2584	0.3611
		10K	MSE	0.0951	0.0078	0.0399	0.0620	0.0503
			Variance	0.0021	0.0002	0.0013	0.0039	0.0030
			MAE coverage	0.2719	0.3083	0.3805	0.2394	0.3422
		20K	MSE	0.0894	0.0088	0.0404	0.0568	0.0505
			Variance	0.0015	0.0003	0.0014	0.0033	0.0037
			MAE coverage	0.2753	0.3611	0.3784	0.2513	0.3571
		40K	MSE	0.0906	0.0080	0.0435	0.0519	0.0500
			Variance	0.0018	0.0002	0.0007	0.0029	0.0037
			MAE coverage	0.2806	0.3202	0.3945	0.2632	0.3703
		100K	MSE	0.1101	0.0075	0.0535	0.0582	0.0521
			Variance	0.0031	0.0002	0.0018	0.0036	0.0036
			MAE coverage	0.2684	0.4179	0.3452	0.2329	0.3405
		200K	MSE	0.1113	0.0080	0.0613	0.0816	0.0552
			Variance	0.0028	0.0003	0.0025	0.0062	0.0052
			MAE coverage	0.2635	0.3556	0.2872	0.2252	0.2632
Chronos Tiny (8M)	Zero-shot	N/A	MSE	0.1403	0.0095	0.0781	0.0508	0.0514
			Variance	0.0060	0.0002	0.0045	0.0041	0.0041
			MAE coverage	0.2330	0.4857	0.2644	0.2427	0.3145
	After fine-tuning	0	MSE	0.0956	0.0054	0.0244	0.0445	0.0420
			Variance	0.0032	0.0002	0.0005	0.0030	0.0028
			MAE coverage	0.2667	0.4036	0.3582	0.2804	0.2908
		10K	MSE	0.1113	0.0062	0.0377	0.0431	0.0665
			Variance	0.0054	0.0002	0.0008	0.0024	0.0069
			MAE coverage	0.2531	0.4071	0.3691	0.2575	0.3434
		20K	MSE	0.1212	0.0058	0.0338	0.0487	0.0784
			Variance	0.0070	0.0002	0.0007	0.0028	0.0198
			MAE coverage	0.2736	0.3972	0.3666	0.2440	0.3382
		40K	MSE	0.1085	0.0061	0.0396	0.0442	0.0688
			Variance	0.0025	0.0002	0.0006	0.0024	0.0271
			MAE coverage	0.2615	0.4214	0.3439	0.2495	0.3365
		100K	MSE	0.1132	0.0092	0.0550	0.0457	0.0768
			Variance	0.0037	0.0003	0.0012	0.0026	0.0358
			MAE coverage	0.2684	0.4143	0.3098	0.2441	0.3067
		200K	MSE	0.1159	0.0095	0.0587	0.0478	0.0830
			Variance	0.0033	0.0003	0.0015	0.0026	0.0343
			MAE coverage	0.2458	0.4214	0.2947	0.2310	0.2761

Table 6: Ablation study of adding different quantity of synthetic data – summary 2.

Model (model size)	Evaluation type	Synthetic data added	Metrics	In-sample datasets FRED	In-sample datasets World Cup	In-sample datasets EIA	Out-sample datasets Yahoo	Out-sample datasets COVID
Chronos Mini (20M)	Zero-shot	N/A	MSE	0.1409	0.0108	0.0785	0.0483	0.0589
			Variance	0.0064	0.0002	0.0049	0.0040	0.0061
			MAE coverage	0.2330	0.4857	0.2664	0.2425	0.3548
	After fine-tuning	0	MSE	0.1039	0.0054	0.0194	0.0460	0.0421
			Variance	0.0048	0.0002	0.0004	0.0034	0.0028
			MAE coverage	0.2719	0.3722	0.3634	0.2825	0.2965
		10K	MSE	0.1029	0.0060	0.0240	0.0465	0.0579
			Variance	0.0037	0.0002	0.0005	0.0030	0.0052
			MAE coverage	0.2635	0.3556	0.3696	0.2548	0.3565
		20K	MSE	0.1176	0.0059	0.0300	0.0447	0.0518
			Variance	0.0037	0.0002	0.0008	0.0022	0.0033
			MAE coverage	0.2552	0.3667	0.3839	0.2566	0.2566
		40K	MSE	0.1069	0.0062	0.0378	0.0443	0.0724
			Variance	0.0039	0.0002	0.0007	0.0023	0.0224
			MAE coverage	0.2684	0.3944	0.3535	0.2553	0.3520
		100K	MSE	0.1262	0.0110	0.0497	0.0453	0.0778
			Variance	0.0059	0.0005	0.0015	0.0023	0.0517
			MAE coverage	0.2625	0.4143	0.3374	0.2416	0.3251
		200K	MSE	0.1048	0.0123	0.0573	0.0461	0.0811
			Variance	0.0028	0.0006	0.0016	0.0025	0.0539
			MAE coverage	0.2583	0.4107	0.3042	0.2406	0.2940
Chronos Small (46M)	Zero-shot	N/A	MSE	0.1428	0.0113	0.0764	0.0519	0.0641
			Variance	0.0070	0.0002	0.0043	0.0045	0.0062
			MAE coverage	0.2365	0.4893	0.2683	0.2324	0.3388
	After fine-tuning	0	MSE	0.1013	0.0059	0.0158	0.0490	0.0392
			Variance	0.0036	0.0002	0.0004	0.0036	0.0028
			MAE coverage	0.2858	0.3694	0.3605	0.2712	0.3116
		10K	MSE	0.1111	0.0060	0.0221	0.0491	0.0678
			Variance	0.0021	0.0002	0.0005	0.0028	0.0094
			MAE coverage	0.2719	0.3611	0.3561	0.2599	0.3577
		20K	MSE	0.0913	0.0060	0.0243	0.0455	0.0828
			Variance	0.0019	0.0002	0.0006	0.0024	0.0211
			MAE coverage	0.2583	0.3778	0.3714	0.2643	0.3525
		40K	MSE	0.1160	0.0063	0.0316	0.0432	0.0804
			Variance	0.0060	0.0002	0.0006	0.0022	0.0448
			MAE coverage	0.2510	0.3889	0.3738	0.2575	0.3474
		100K	MSE	0.1154	0.0097	0.0413	0.0438	0.0778
			Variance	0.0058	0.0003	0.0010	0.0023	0.0585
			MAE coverage	0.2583	0.3972	0.3479	0.2473	0.3302
		200K	MSE	0.1134	0.0070	0.0533	0.0453	0.0720
			Variance	0.0031	0.0002	0.0012	0.0024	0.0380
			MAE coverage	0.2552	0.4286	0.3172	0.2333	0.2944