LoRA Implementation and Domain Adaptation on Chronos models

Li Zepeng
s327196@studenti.polito.it

Mendoza Federica
s332190@studenti.polito.it

Miamoto Felipe
s321003@studenti.polito.it

Xu Yan
s326331@studenti.polito.it

***Abstract*—This work investigates the domain adaptation capabilities of Chronos, a family of pre-trained time series forecasting models. We explore how effectively Chronos models, trained on diverse datasets, can be adapted to specific target domains with limited data. Our experiments focus on evaluating performance across varying degrees of domain shift. Furthermore, we implement Low-Rank Adaptation (LoRA), a parameter-efficient fine-tuning technique, to enhance adaptation while minimizing computational cost. We compare the performance of Chronos models fine-tuned with LoRA against traditional fine-tuning methods. Our results demonstrate the potential of Chronos for rapid and effective domain adaptation in time series forecasting, highlighting the benefits of LoRA in reducing training overhead and improving performance in low-data regimes. This study contributes to a deeper understanding of Chronos' transfer learning capabilities and provides practical guidelines for adapting these powerful models to real-world forecasting challenges.**

***Index Terms*—Chronos, forecasting, timeseries, LoRA, fine-tuning**

Link to GitHub repository

I. PROBLEM STATEMENT

Accurate time series forecasting is a critical task with broad implications across diverse fields, from financial markets and healthcare to environmental modeling and resource management. While numerous approaches exist, ranging from classical statistical methods like ARIMA and Exponential Smoothing to sophisticated machine learning algorithms such as XGBoost and Random Forests, each presents its own set of challenges. Statistical models often fall short when faced with the non-linear dynamics characteristic of many real-world time series. Machine learning techniques, while more flexible, typically require substantial feature engineering, a process that can be both time-consuming and prone to subjective bias. Deep learning architectures, including LSTMs and Transformers, offer powerful tools for capturing complex temporal dependencies, but their performance is often contingent on the availability of large, high-quality datasets and significant computational resources, limiting their practicality in some scenarios. Moreover, the need to retrain these models extensively for each new time series domain hinders the development of a truly generalizable forecasting framework.

To address this, Chronos has emerged as a promising solution—a pre-trained transformer model for universal time series forecasting. Inspired by large language models, Chronos tokenizes time series, enabling it to handle variable-length inputs across domains. Initial results suggest strong performance with minimal fine-tuning, making it a scalable alternative to domain-specific models. However, two critical questions persist: (1) How well does Chronos transfer pre-trained knowledge to specialized domains with unique temporal patterns? (2) Can fine-tuning be optimized for computational efficiency without compromising accuracy?

This study tackles these questions through two research thrusts. First, we evaluate Chronos in Bitcoin price forecasting—a highly volatile domain influenced by market sentiment and regulatory shifts. Existing models often fail to generalize across regimes, underscoring the need for adaptable frameworks. [4] By fine-tuning Chronos on Bitcoin data, we assess its ability to balance domain-specific learning with retained generalization.

Second, we explore Low-Rank Adaptation (LoRA) to optimize fine-tuning. Traditional fine-tuning updates all parameters, incurring high computational costs. LoRA mitigates this by injecting trainable low-rank matrices while freezing pre-trained weights, significantly reducing resource demands. For financial applications requiring rapid model updates, LoRA could enable near real-time adaptation. We systematically compare LoRA-based fine-tuning with full fine-tuning, evaluating trade-offs in accuracy, training time, and efficiency, providing insights into scalable time series adaptation.

II. RELATED WORK

A. Bitcoin dataset

Given the high volatility of the cryptocurrency market and its reliance on high-frequency time series analysis for trading strategies and risk assessment, we selected from Kaggle a dataset capturing Bitcoin's stock values from 2018 to 2022 [3]. Specifically, the following features were included: open, high, low, close prices, and trading volume. These features exhibit strong intercorrelation due to their inherent relationship. Indeed, each of them offers a slightly different perspective on the Bitcoin's price trend, hence providing minimal unique information for model training. This high degree of redundancy and correlation can negatively impact model performance by hindering the identification of salient features, potentially leading to overfitting on less relevant variations and slower convergence. Furthermore, the dimensionality resulting from including all four features increases the computational complexity and memory requirements, extending training times. In light of these observations, we focused our task on a single feature, namely, the "close" feature. The dataset provides multiple sampling frequencies, ranging from three-minute intervals to six-hour intervals. For our experiments, we adopt a 15-minute sampling interval, which offers a balance between granularity and computational efficiency. While higher frequencies (e.g., three-minute intervals) introduce excessive noise and computational overhead, lower frequencies (e.g., hourly intervals) risk missing short-term arbitrage opportunities critical in cryptocurrency trading. To ensure robust evaluation, the dataset is temporally partitioned, with 104,732 samples from January 2018 to December 2020 used for training, and 37,878 samples from January 2021 to January 2022 reserved for testing. This partitioning reflects real-world deployment scenarios, exposing the model to unseen market conditions and enabling a more rigorous assessment of its generalization capability under dynamic market regimes. Though stock values dataset were not included in the Chronos training phase, nevertheless we expect good performance from the models, due to the presence of economy and finance related training datasets.

B. Chronos model

The Chronos model, introduced by Cheng et al. (2023) [1], revolutionizes time series forecasting by unifying diverse temporal patterns within a single pre-trained transformer architecture. Inspired by large language models (LLMs), Chronos tokenizes time series into numerical tokens via scaling and quantization, reframing forecasting as a sequence modeling task. This allows it to process heterogeneous datasets—ranging from sensor readings to financial metrics—without domain-specific preprocessing. Pre-trained on synthetic and real-world time series through parametric sampling and dataset perturbation, Chronos captures broad temporal dependencies. It outperforms classical methods (e.g., Theta) and deep learning models (e.g., N-BEATS, TFT) in low-data settings, leveraging pre-trained representations for strong zero-shot generalization across domains like healthcare, energy, and retail. However, its adaptation to domain-specific shifts, such as financial market volatility, remains underexplored. Additionally, its fine-tuning efficiency, a key constraint for transformers, necessitates optimization, motivating the integration of parameter-efficient techniques like LoRA.

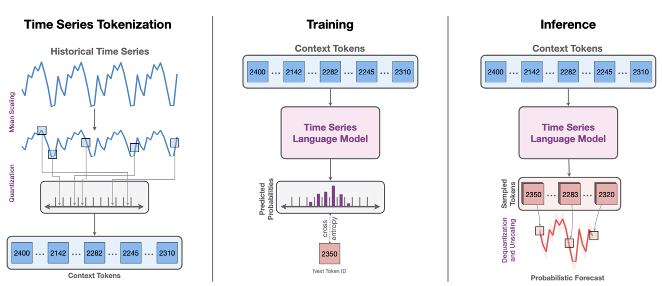

Fig. 1: High-level depiction of Chronos. [1]

C. LoRA technique

Low-Rank Adaptation (LoRA), introduced by Hu et al. (2021) [2], is a key technique for parameter-efficient domain adaptation in large-scale pre-trained models. Initially developed for natural language processing (NLP), LoRA freezes model weights while injecting trainable low-rank matrices into transformer layers, significantly reducing trainable parameters while maintaining near full fine-tuning performance. Though underexplored in time series analysis, LoRA holds promise for balancing pre-trained temporal patterns with domain-specific dynamics, such as Bitcoin's volatility. In NLP and computer vision, LoRA accelerates fine-tuning and mitigates catastrophic forgetting, a crucial advantage for adapting models to sparse or shifting distributions. In financial time series, where market shifts demand rapid updates, LoRA enables efficient retraining without erasing foundational knowledge. Preliminary applications, such as fault detection and demand forecasting, achieve competitive accuracy with only 10–20% of the trainable parameters required for full fine-tuning. However, its effectiveness in tokenized time series models like Chronos remains unvalidated, particularly in latency-sensitive domains like cryptocurrency markets. Integrating LoRA into Chronos bridges this gap, demonstrating how parameter-efficient adaptation can enhance domain precision while optimizing computational costs.

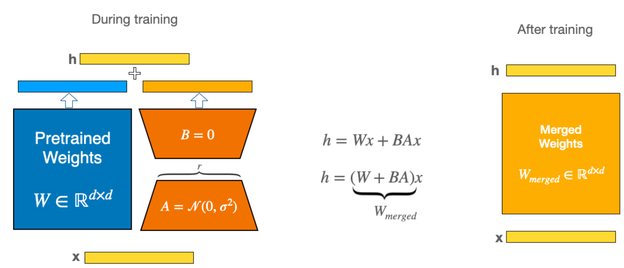

Fig. 2: Parameter-Efficient Fine-Tuning (LoRA) [2]

LoRA (Low-Rank Adaptation) introduces two low-rank matrices, A and B, to modify the pretrained weights W without directly updating them. During training, only A and B are optimized to learn a low-rank adaptation BA. After training, the adapted weights are merged as

$$W_{merged} = W + BA$$

enabling efficient fine-tuning with reduced parameter overhead while maintaining computational efficiency during inference. .

III. EXPERIMENTS

A. Zero-Shot Domain Adaptation

Chronos is a pre-trained transformer model for universal time series forecasting, demonstrating strong zero-shot accuracy on unseen forecasting tasks. In this experiment, we aim to evaluate *Chronos-T5-Tiny* model's performance on our Bitcoin transaction test dataset without fine-tuning, using key metrics such as Mean Absolute Scaled Error (MASE) and Weighted Quantile Loss (WQL). We selected this version due to its lower computational demands, making it feasible for evaluation on Google Colab. Specifically, the evaluation was performed using the following configurations:

- `batch size=32` to ensure efficient computation while fitting within Colab's GPU memory constraints.
- `device`: The cuda:0 device is utilized for inference, leveraging Colab's free T4 GPU for optimal performance.

- number of samples: 20 for evaluation.
- `offset = 96` : given that the Bitcoin transaction dataset is sampled at a 15-minute interval, we configure the historical window size (offset) to 96, allowing the model to capture both short-term trends and long-term patterns. The prediction length is set to 24, ensuring the forecast aligns with the dataset's temporal structure.

This setup provides a baseline performance assessment of the pre-trained model on our dataset, helping us understand the generalization capability of the model.

B. Full-Parameter Fine-Tuning for Domain Adaptation

In the second experiment, we fine-tuned the Chronos-T5-Tiny model on our dataset before evaluating its performance. The training process was conducted using the following configurations:

- `training steps: 1000`; given our dataset's size, a prolonged training period could lead to overfitting. Limiting training to 1000 steps ensures the model learns useful representations without excessive memorization. Colab's limited computational resources also restrict us from training for an extended number of steps.
- `learning rate: 0.001`; this learning rate balances convergence speed and stability, ensuring that the model learns effectively without causing training instability.

Evaluation after fine-tuning allows us to compare the model's performance before and after fine-tuning to quantify the impact of domain adaptation.

C. Parameter-Efficient Fine-Tuning (LoRA) for Domain Adaptation

In the previous experiment, we investigated the impact of fine-tuning Chronos on the model's performance when applied to the Bitcoin transaction test dataset. In this experiment, our objective shifts towards accelerating the fine-tuning process using Low-Rank Adaptation (LoRA) while aiming to achieve comparable performance to full fine-tuning.

To achieve this, we adopt the LoRA technique, which enables parameter-efficient fine-tuning by freezing the pre-trained Chronos model and updating only a small set of additional trainable parameters. The training configuration remains consistent with the previous experiment, except that only the LoRA adapter parameters are updated, while the original model parameters remain unchanged. The selected LoRA adapter configuration is as follows:

- `LoRA rank` ($r = 8$): Controls the dimension of the low-rank matrices. A moderate rank of 8 provides a balance between expressiveness and computational efficiency.
- `Scaling factor` ($\alpha = 16$): Scales the LoRA-adapted transformations, ensuring effective adaptation without significantly increasing computational cost.
- `Dropout probability` (dropout = 0.1): Introduces a 10% dropout rate to prevent overfitting by reducing reliance on specific neurons during training.
- `Target layers`: Applies LoRA to the query (Q) and value (V) projection layers, as they play a pivotal role in attention by guiding focus (Q) and transmitting information (V), making them the most effective points for adaptation.
- `Bias setting` (bias = none): Indicates that biases remain unmodified, ensuring a more stable and computationally efficient fine-tuning process.

By leveraging LoRA, we significantly reduce the number of trainable parameters, enabling faster adaptation with lower memory overhead while preserving the core knowledge embedded in Chronos.

IV. Results

A. Zero-Shot vs. Full Fine-Tuning

The zero-shot performance serves as a crucial baseline, reflecting the pre-trained model's inherent capability to generalize to Bitcoin's close price data without any specific adaptation. The reported Mean Absolute Scaled Error (MASE) of 0.2155 and Weighted Quantile Loss (WQL) of 0.0063 indicate a moderate level of predictive accuracy. This level of performance likely stems from a mismatch between the pre-training data characteristics and the target data's high-frequency nature. Indeed, during its training phase the Chronos-T5-tiny model has not encountered a substantial proportion of high-frequency data. In particular, since the economic and financial time series in the training set were primarily at monthly or daily frequencies, the model may struggle to handle potentially increased noise and short-term volatility present in 15-minute Bitcoin data.

Upon employing full fine-tuning, a significant improvement in performance is observed. The MASE decreases to 0.1730, representing a reduction of approximately 19.7% from the zero-shot baseline. Similarly, the WQL improves to 0.0039, a decrease of roughly 38.1% . This substantial reduction in error metrics highlights the effectiveness of adapting the entire model to the specific characteristics of the Bitcoin price time series. By training all model parameters on the target data, the model can learn to capture the specific nuances of Bitcoin price dynamics at this granular level, filter out noise, identify relevant short-term fluctuations, and potentially even capture the impact of market events or news sentiment reflected in the high-frequency data learn the intricate dynamics of high-frequency trading, which it likely missed in the zero-shot scenario.

The substantial performance gains compared to the zero-shot approach highlight the importance of targeted training for capturing the specific dynamics of the target timeseries.

B. Full vs. LoRA Fine-Tuning

The comparative assessment of full-parameter fine-tuning and LoRA-based parameter-efficient fine-tuning reveals nuanced trade-offs among performance, computational efficiency, and generalization capability. As illustrated in Table I, full fine-tuning achieves a Mean Absolute Scaled Error (MASE) of 0.173 and a Weighted Quantile Loss (WQL) of 0.0039, whereas LoRA fine-tuning attains a slightly lower MASE of 0.1599 but a marginally higher WQL of 0.0042, with a 6%

reduction in training time (14 minutes 49 seconds vs. 15 minutes 39 seconds). This contrast highlights the fundamental differences in how these approaches update model parameters, influencing their ability to generalize and capture distributional nuances.

The observed improvement in MASE aligns with LoRA's intrinsic regularization effect, which primarily stems from its parameter-freezing mechanism. By constraining updates to low-rank adaptations in the query and value projections, LoRA effectively preserves Chronos' pre-trained temporal representations, leading to better generalization across unseen Bitcoin price regimes. However, its slightly elevated WQL suggests a minor trade-off in capturing extreme quantile behaviors, likely due to its reduced capacity to adjust the full distributional structure of model predictions.

While full fine-tuning theoretically allows for complete model adaptation, in practice, it risks capturing transient noise rather than meaningful temporal patterns, particularly under the limited 1000 training steps imposed by computational constraints. This susceptibility to overfitting may explain its diminished robustness when applied to the volatile 2021–2022 period. Conversely, LoRA's rank-8 adaptation provides a balanced approach, enabling localized adjustments to attention mechanisms while maintaining the model's global temporal priors. The comparable training durations—despite LoRA's parameter efficiency—suggest that its sparse updates facilitate faster convergence, offsetting the computational overhead introduced by low-rank matrix injections. These findings underscore LoRA's practicality for financial forecasting under real-time constraints. Although the slightly elevated WQL warrants further refinement—potentially through adaptive rank selection or targeted bias tuning—these results affirm that parameter-efficient fine-tuning not only achieves competitive accuracy but also ensures enhanced scalability, a crucial factor for real-time cryptocurrency trading.

TABLE I: Performance Comparison of Different Fine-Tuning Methods

Method	**MASE** $\downarrow$	**WQL** $\downarrow$	**Training Time** $\downarrow$
Zero-Shot	0.2155	0.0063	-
Full Fine-Tuning	0.1730	**0.0039**	15m 39s
LoRA Fine-Tuning	**0.1599**	0.0042	**14m 49s**

Error assessment and training time of each experiment. Lower values indicate better performance. Best values are highlighted in bold.

V. Conclusion

In this work, we explored the domain adaptation capabilities of Chronos, a pre-trained time series forecasting model, and investigated the efficiency of Low-Rank Adaptation (LoRA) for fine-tuning. Our study evaluated Chronos' performance in Bitcoin price forecasting across three adaptation strategies: zero-shot inference, full fine-tuning, and LoRA-based fine-tuning. Our results demonstrate that while Chronos exhibits reasonable zero-shot performance, it struggles with high-frequency financial time series due to its pre-training on lower-frequency datasets. Fine-tuning significantly improves accuracy, with LoRA achieving comparable results to full fine-tuning while reducing computational costs. Notably, LoRA enhances generalization by preserving the pre-trained model's temporal representations while minimizing overfitting to short-term noise.

Our contributions are significant in two key aspects. First, we highlight the potential of pre-trained time series models for rapid domain adaptation, showing that fine-tuning enables effective transfer to high-volatility financial markets. This underscores the need for expanding pre-training datasets to include higher-frequency time series. Second, our findings demonstrate that parameter-efficient adaptation techniques like LoRA can achieve competitive performance with significantly lower computational overhead, paving the way for scalable, real-time financial forecasting.

However, our study was constrained by hardware limitations. With access to more powerful GPUs, we could have leveraged a larger Chronos variant, potentially improving accuracy and generalization further. Additionally, while we focused on Bitcoin price forecasting, exploring other financial assets, such as stock price movements of various companies, would have provided broader insights on Chronos' domain adaptation capability. Unfortunately, the lack of publicly available large-scale datasets limited the scope of our experiments. Future work can explore adaptive rank selection for LoRA, hybrid fine-tuning strategies, and integrating external financial signals to further enhance Chronos' robustness in dynamic market environments.

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, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, Yuyang Wang. Chronos: Learning the Language of Time Series.

[2] Edward J. Hu, Yelong Shen, David Du, Chia-Wei Liu, Zhuang Liu, Trevor Darrell, and Kilian Q. Weinberger. LoRA: Low-Rank Adaptation of Large Language Models.

[3] Shubham2703. Bitcoin Time Series Data (Jan 2018 - Jan 2022). *Kaggle Dataset*, 2022.

[4] Samuel Asante Gyamerah. Are Bitcoins price predictable? Evidence from machine learning techniques using technical indicators. *arXiv preprint arXiv:1909.01268*, 2019.