Dual Adaptation of Time-Series Foundation Models for Financial Forecasting

Fatemeh Chitsaz [*1] **Saman Heratizadeh** [*1]

Abstract

Recent progress in time-series foundation models has expanded forecasting capabilities across domains. However, their application to finance remains constrained by data scarcity, volatility, and overfitting. We present Dual Adaptation, a lightweight adaptation of the TimesFM foundation model for financial forecasting, featuring a dual-module design: a Generalizer Adapter that learns broad temporal patterns across assets and an Identity Signature module that captures asset-specific signals, forming a lightweight layer tuned on top of a frozen foundation model. The method is evaluated in both in-domain and zero-shot settings, showing improved forecasting performance compared to the frozen model and common tuning baselines. To enhance generalization, the Identity Signature is removed during inference, allowing the Generalizer Adapter to apply the shared knowledge it has learned to unseen assets. This design improves both stability and cross-asset generalization, offering a practical solution for adapting large models to noisy, low-resource financial forecasting tasks.

1. Introduction

Time-series forecasting is a major challenge in finance. Financial data are often noisy, nonstationary, and limited in volume, which makes generalization across assets difficult. Both classical models and modern deep learning approaches frequently overfit and perform poorly on unseen instruments(Han et al., 2019).

Foundation models, which are large architectures pretrained on massive time-series corpora, have shown strong few-shot and zero-shot capabilities on structured forecasting benchmarks(Cao et al., 2024). TimesFM, a decoder-style attention-based model, is one of the most prominent examples(Das et al., 2024). However, its training corpus contains limited exposure to financial data, and its performance in financial settings remains largely untested. As a result, its ability to model noisy and irregular price trajectories cannot be assumed.

Full fine-tuning of TimesFM on financial datasets such as the SP100 can improve in-domain performance, but it is computationally intensive, sensitive to initialization, and offers only marginal improvements in generalization to unseen assets. Lightweight alternatives like LoRA, bias tuning, and layer-norm tuning are far more efficient and work well within the training domain, but they tend to exhibit reduced performance when evaluated on out-of-domain financial instruments.

To address these limitations, we introduce Dual Adaptation, a lightweight dual-module adaptation strategy for financial forecasting. It consists of a Generalizer Adapter that learns temporal patterns shared across assets, and an Identity Signature module that encodes asset-specific signals. The Identity Signature is used only during training to help the Generalizer Adapter focus on robust, generalizable patterns rather than overfitting to instance-level noise. At inference time, the Identity Signature is removed, and the Generalizer Adapter alone is used to forecast both seen and unseen assets.

2. Related Work

2.1. Foundation Models for Time Series

Inspired by the success of large language models (LLMs) in NLP, time-series foundation models (TSFMs) have been developed to learn general-purpose temporal representations from large-scale data(Zhou et al., 2021). TimesFM is a recent and prominent example of this trend. It is a decoder-only transformer model with 200 million parameters, pretrained on a mixture of synthetic and real-world time-series data covering over 100 billion time points. TimesFM demonstrates strong zero-shot performance on structured forecasting benchmarks such as Darts and Monash, requiring little or no fine-tuning to generalize across tasks.

However, its effectiveness in financial forecasting remains

[*]Equal contribution [1]School of Intelligent Systems, University of Tehran, Tehran, Iran. Correspondence to: Fatemeh Chitsaz <fatemeh.chitsaz@ut.ac.ir>, Saman Heratizadeh <haratizadeh@ut.ac.ir>.

Proceedings of the 1st ICML Workshop on Foundation Models for Structured Data, Vancouver, Canada. 2025.

limited. Financial time series are typically noisy, lack strong seasonal patterns, and are often limited in quantity(Jia et al., 2024). These characteristics make it difficult for general-purpose TSFMs to perform well out-of-the-box(Nie et al., 2023). The PFNet model addresses this challenge by fine-tuning TimesFM on curated financial datasets and adjusting loss functions to better suit price prediction(Fu et al., 2025). While PFNet achieves improved results, it relies on full-model fine-tuning, which is computationally expensive and prone to overfitting.

Other recent foundation models, such as Lag-Llama, MOIRAI, Chronos, and TimeGPT, explore varied architectural designs including encoder-decoder formats, patch-based attention, and probabilistic outputs(Rasul et al., 2024; Zeng et al., 2023; Fan & Shen, 2024; Cao et al., 2024).

Despite their architectural diversity, most of these models have not been evaluated in the context of real-world financial markets, and many are proprietary, limiting their reproducibility and accessibility(Liu et al., 2023; Salinas et al., 2020).

2.2. Parameter-Efficient Fine-Tuning

As foundation models grow in size, full fine-tuning becomes increasingly resource-intensive, particularly in domains with limited labeled data. Parameter-efficient fine-tuning (PEFT) strategies have emerged as practical alternatives. LoRA introduces low-rank trainable matrices into attention layers, allowing for compact adaptation with minimal memory and compute requirements(Hu et al., 2022). Similarly, adapter modules, bias-only tuning, and layer-norm adaptation freeze most of the model while updating a small subset of parameters(Ben Zaken et al., 2022; Qi et al., 2022).

While parameter-efficient fine-tuning (PEFT) is widely adopted in NLP and vision, existing PEFT variants for time-series foundation models are scarce, and in our experiments they lose substantial accuracy on out-of-domain (OOD) financial instruments. We therefore introduce Dual Adaptation, a PEFT approach that explicitly separates general temporal structure from asset-specific information. By disentangling shared and specific signals during training, Dual Adaptation helps the general component avoid overfitting to instance-level noise and enables more robust generalization to unseen assets.

3. Methodology

We consider the task of forecasting future values of a financial time series using a pre-trained foundation model. Let $\mathbf{x}_{1:C}$ denote a sequence of historical asset prices and $\mathbf{x}_{C+1:C+H}$ the target prediction window. We use daily historical price data from the S&P100 as training data and evaluate performance on both S&P100 test stocks and a disjoint zero-shot set consisting of 100 randomly selected S&P500 stocks.

3.1. Base Model: TimesFM

TimesFM is a decoder-only transformer with three key components: (1) an input residual block that projects raw time-series patches to the model dimension, (2) a stack of causal self-attention layers, and (3) an output residual block that maps hidden states to forecast values. We use the public TimesFM weights and keep all backbone parameters frozen during adaptation.

3.2. Dual Adaptation Architecture

To enable efficient domain adaptation with minimal computation, we introduce a lightweight fine-tuning module composed of two components:

- **Generalizer Adapter:** A residual two-layer feedforward network applied after the input projection layer, shared across all stocks.
- **Identity Signature:** A learnable embedding that encodes asset-specific information via a trainable linear projection from one-hot identity vectors.

Let $\mathbf{x} \in \mathbb{R}^{B\times N\times D}$ be the output of TimesFM's residual input block, where B is the batch size, $N = C/P$ is the number of patches obtained by splitting a context window of length C into non-overlapping chunks of size $P = 32$, and $D = 1280$ is the model's hidden dimension.

Suppose there are S distinct stocks in the training set, each assigned a unique index $i \in \{0, 1, ..., S-1\}$. We represent the identity of stock i using a one-hot vector $\mathbf{z}_i \in \mathbb{R}^S$, where:

$$\mathbf{z}_i[j] = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

We then define a learnable identity Signature matrix $\mathbf{W}_{\text{sig}} \in \mathbb{R}^{S\times D}$ that maps one-hot identity vectors to D-dimensional Signatures. The identity vector $\mathbf{e}_i \in \mathbb{R}^D$ for stock i is computed as:

$$\mathbf{e}_i = \mathbf{z}_i \mathbf{W}_{\text{sig}}$$

This formulation is equivalent to selecting the i-th row of the Signature matrix $\mathbf{W}_{\text{sig}}$, but makes explicit that the Identity Signature is a learned linear transformation of one-hot stock identifiers.

To inject this asset identity into the model, we broadcast $\mathbf{e}_i$ across the patch dimension of the token sequence:

$$\mathbf{x}' = \mathbf{x} + \text{Broadcast}(\mathbf{e}_i)$$

The enriched representation $\mathbf{x}'$ is then passed through the Generalizer Adapter, a residual two-layer feedforward network that is shared across all stocks.

$$\text{GeneralizerAdapter}(\mathbf{x}') = \mathbf{x}' + f_{\text{up}}\left(\sigma\left(f_{\text{down}}(\mathbf{x}')\right)\right)$$

Here, $f_{\text{down}} : \mathbb{R}^D \to \mathbb{R}^d$ projects to a bottleneck dimension $d = 64$, σ is the Swish activation function, and $f_{\text{up}} : \mathbb{R}^d \to \mathbb{R}^D$ restores the original dimensionality. An overview of the full architecture, including the integration of the Identity Signature and Generalizer Adapter into TimesFM, is illustrated in Figure 1.

3.3. Fine-Tuning and Inference

During training, we update only the parameters of the Generalizer Adapter and the Identity Signature, using data from the S&P100. The Identity Signature provides stock-specific identity information, allowing the Generalizer Adapter to separate idiosyncratic patterns from shared temporal structure.

At inference time, we adopt a selective strategy:

- **Seen stocks (in-domain test):** We keep the corresponding Identity Signature active, allowing the model to leverage asset-specific information it learned during training.
- **Unseen stocks (zero-shot):** We remove the Identity Signature and route the input only through the Generalizer Adapter. Since the adapter was trained with a variety of identities, it learns general temporal dynamics that transfer effectively to unseen stocks.

This design enables a flexible balance between specialization and generalization. The Generalizer Adapter captures temporal patterns shared across assets, while the Identity Signature assigns each stock a trainable identity vector.

3.4. Weight Update Analysis Metrics

We conduct two diagnostic tests to evaluate the effect of the Identity Signature on the training dynamics and generalization behavior of the Generalizer Adapter. These tests are designed to measure two key factors: *Stability* and *Transferability*.

1. Stability Factor (Intra-Domain Consistency)

To assess consistency across runs, we compute the cosine similarity between the final weights of the Generalizer Adapter obtained from five independent training runs on the S&P 100 dataset. Let $W_\ell^{(r)}$ denote the weight vector of adapter layer ℓ from run r. For every pair of runs $i \neq j$, we compute:

$$\text{CosSim}_\ell(i, j) = \frac{\langle W_\ell^{(i)}, W_\ell^{(j)} \rangle}{\|W_\ell^{(i)}\| \cdot \|W_\ell^{(j)}\|}$$

The resulting similarity scores reflect how consistently the adapter converges across random seeds.

2. Transferability Factor (Cross-Domain Similarity)

To evaluate generalization across asset groups, we fine-tune the Generalizer Adapter independently on two disjoint stock sets: (i) the S&P 100 training universe, and (ii) a randomly sampled subset of 100 S&P 500 stocks not seen during training. For each domain, five separate runs are performed. Cosine similarity is then computed between all pairs of adapter weight vectors across the two domains, resulting in 25 inter-domain comparisons.This analysis measures the degree to which the adapter captures representations that are structurally similar across different stock universes.

4. Experiment Setup

We use historical daily stock data for the SP100 and SP500, obtained from Yahoo Finance. The forecasting task is formulated as one-step-ahead prediction using a context window of 64 days. The dataset is split chronologically: data from 2000 to 2021 is used for training, 2022 for validation, and data from 2023 to 2024 is reserved for testing, which we refer to as the in-domain (ID) evaluation period.

All models are trained and validated exclusively on SP100 stocks. To evaluate out-of-domain (OOD) generalization, we construct a disjoint set of 100 stocks randomly sampled from the SP500 (excluding all SP100 members). These OOD stocks are evaluated only over the 2023–2024 test window, with no exposure during training or validation.

Model performance is reported as the mean squared error (MSE), averaged across all stocks in each split. To ensure robustness, all experiments are repeated using five different random seeds, and the final metrics are computed as the average across these runs.

5. Results and Discussion

We evaluate our approach against a range of fine-tuning strategies on the SP100 and SP500 datasets using the TimesFM foundation model. The compared strategies include LoRA (with rank 8), a parameter-efficient technique that integrates low-rank trainable matrices into the attention mechanism; Residual-Only Tuning, which updates only the

input and output feedforward layers while leaving the transformer layers frozen; Bias Tuning, also known as BitFit, which modifies only the model's bias parameters; and Layer Norm Tuning, which updates solely the scale and shift parameters within the normalization layers. The forecasting performance and parameter efficiency of all methods are summarized in Table 1.

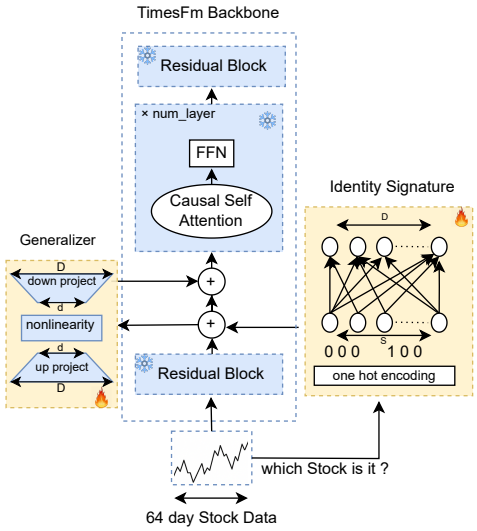

Figure 1. Illustration of the proposed Dual Adaptation architecture. The input stock time series is enriched with a stock-specific Identity Signature and then passed through the Generalizer Adapter. Only these modules are fine-tuned; the TimesFM backbone remains frozen.

5.1. In-Domain Performance (ID)

As a baseline, frozen TimesFM yields 39.76 MSE on SP100. Full fine-tuning reduces this to 32.83, while our lightweight method achieves 32.36. Though LoRA performs slightly better, our approach is more parameter-efficient and better suited to resource-limited settings.

5.2. Generalization to Unseen Stocks (OOD)

To assess generalization, we test on 100 SP500 stocks excluding all SP100 members. TimesFM yields 24.12 MSE, full fine-tuning improves to 20.79, and our method achieves the best result at 20.65. While LoRA performs better in-domain, it overfits and underperforms on excluded stocks. Our compact design offers stronger generalization.

5.3. Generalizer Consistency and Cross-Stock Generalization

To assess the effect of the Identity Signature, we report two cosine similarity metrics. First, we measure stability

Table 1. Forecasting error and number of trainable parameters on ID (in-domain S&P100) and OOD (out-of-domain S&P500) test sets. Lower values indicate better performance.

METHOD	ID MSE	OOD MSE	PARAMS
OURS	32.36	**20.65**	293K
LORA	**31.68**	20.97	1.64M
LAYERNORM TUNE	32.58	21.05	77K
BIAS TUNE	32.67	21.04	187K
FULL TUNE	32.83	20.79	203M
RESIDUAL-ONLY	33.03	21.02	6.73M
ONLY GENERALIZER	33.16	20.88	165K
ONLY SIGNATURE	38.64	23.77	128K
PFNET	38.80	24.35	203M
ZERO-SHOT	39.76	24.12	0

across five training runs on SP100. The average similarity of the Generalizer Adapter's final weights increases from 91% without the Identity Signature to 97% with it. This result highlights that the added identity information leads to more stable and consistent optimization behavior.

Second, to evaluate whether the shared adapter captures truly transferable structure, we compare weights from five SP100 runs with five runs on a disjoint SP500 subset. The average similarity improves from 88% to 90% when using the Identity Signature, suggesting that Dual Adaptation is not merely overfitting to the training distribution but is learning generalizable patterns that extend across asset domains. The detailed results are shown in Table 2.

Table 2. Cosine similarity of the Generalizer Adapter under different training setups.

Metric	**Without Signature**	**With Signature**
Stability Factor	91%	97%
Transferability Factor	88%	90%

6. Conclusion and Future Work

We proposed **Dual Adaptation**, a lightweight adaptation framework for financial time-series foundation models. By combining a shared Generalizer Adapter and an Identity Signature module, our method improves stability and generalization while tuning a small number of parameters.

Future work includes applying Dual Adaptation to multivariate and high-frequency data, and evaluating its adaptability to other time-series foundation models.

References

Ben Zaken, E., Goldberg, Y., and Ravfogel, S. Bitfit: Simple parameter-efficient fine-tuning for transformer-based

masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Cao, D., Jia, F., Ö. Arik, S., Pfister, T., Zheng, Y., Ye, W., and Liu, Y. TEMPO: Prompt-based generative pre-trained transformer for time-series forecasting. In *International Conference on Learning Representations*, 2024.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *Proceedings of the International Conference on Machine Learning*, 2024.

Fan, J. and Shen, Y. Stockmixer: A simple yet strong MLP-based architecture for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Fu, X., Hirano, M., and Imajo, K. Financial fine-tuning a large time series model. In *2025 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CiFer)*, Trondheim, Norway, March 2025. IEEE. doi: 10.1109/CiFer64978.2025.10975735. URL https://ieeexplore.ieee.org/document/10975735.

Han, Z., Zhao, J., Leung, H., Ma, K. F., and Wang, W. A review of deep learning models for time-series prediction. *IEEE Sensors Journal*, 19(20):9329–9348, 2019. doi: 10.1109/JSEN.2019.2922045.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR 2022)*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jia, F., Wang, K., Zheng, Y., Cao, D., and Liu, Y. GPT4MTS: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Liu, X.-Y., Wang, G., Yang, H., and Zha, D. FinGPT: Democratizing internet-scale data for financial large language models. In *NeurIPS Instruction-Following Workshop*, 2023.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Qi, W., Ruan, Y.-P., Zuo, Y., and Li, T. Parameter-efficient tuning on layer normalization for pre-trained language models. *arXiv preprint arXiv:2211.08682*, 2022. URL https://arxiv.org/abs/2211.08682.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., Biloš, M., Garg, S., Schneider, A., and Chapados, N. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024. arXiv preprint.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time-series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10984–10992, 2023.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long-sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11113, 2021.