Extending Chronos: Fine-Tuning for Financial Forecasting and Consistency Calibration

Erfan Bayat
Politecnico di Torino
s328390
s328390@studenti.polito.it

Andrea Vasco Grieco
Politecnico di Torino
s334015
s334015@studenti.polito.it

Mehdi Nickzamir
Politecnico di Torino
s323959
s323959@studenti.polito.it

S.Mohammad S.Ahmadi
Politecnico di Torino
s327914
s327914@studenti.polito.it

***Abstract*—Chronos is a transformer model based on T5, that has demonstrated strong predictive capabilities. In this work, we extend Chronos with two enhancements to improve its performance in a specific domain, for which we chose financial forecasting, and to improve the overall calibration, in order to obtain accurate probability estimates. Firstly, we fine-tuned Chronos using a financial dataset that includes real stock market data and synthetic time series,generated using Geometric Brownian Motion (GBM).This fine-tuning achieved improvements of up to 15% improvement in forecastingaccuracy, as measured by the Mean Absolute Scaled Error (MASE). Secondly, to improve the reliability of the probability estimates, we applied a perturbation-based consistency calibration method (C3). To our knowledge, this method had not previously been applied in this context.This technique consists in directly applying controlled perturbations to the logits and aggregating multiple perturbed predictions, to improve the quality of these probabilities, as measured by the Expected Calibration Error (ECE). We found that it produces more stable and reliable confidence estimates. Our findings highlight the benefits of fine-tuning LLM-based forecasting models for domain-specific applications and demonstrate the effectiveness of consistency calibration in improving prediction reliability.**

***Index Terms*—Time series forecasting, Chronos, fine-tuning, consistency calibration, perturbation, financial modeling, large language models**

I. Introduction

Time series forecasting plays a vital role in fields like finance, energy, and healthcare. While traditional models such as LSTMs and CNNs often require retraining for new datasets, Large Language Models (LLMs) like Chronos [1] offer greater adaptability by modeling time series as token sequences and leveraging large-scale pretraining for improved generalization.

In this paper, we extend Chronos, a T5-based [2] forecasting model, with two key enhancements aimed at improving performance and reliability in the financial domain:

1) **Fine-tuning Chronos on a financial dataset**, enabling it to learn domain-specific market patterns for improved forecasting accuracy.
2) **Applying a perturbation-based consistency calibration method (C3)**, which enhances prediction stability and confidence calibration by aggregating multiple logit-level perturbations.

The implementation and experiments were conducted using our open-source code, available at our GitHub repository.

II. Extension 1: Finetuning Chronos on a financial dataset

A. Overview

This extension investigates how effectively Chronos can be adapted to a specific financial domain through fine-tuning. Our goal is to evaluate how much targeted fine-tuning can improve forecasting accuracy in this domain. We fine-tuned the *chronos-t5-tiny* model (8M parameters) using a curated dataset of stock price time series, augmented with both mixed sequences via TSMixup [1] and synthetic data generated using Geometric Brownian Motions (GBMs), a standard stochastic processes widely used in finance to model time series. Model performance was evaluated using Mean Absolute Scaled Error (MASE) and Weighted Quantile Loss (WQL), allowing us to quantify gains over the original model in a structured, domain-specific setting.

B. Dataset

1) Nifty 100 Index: We used 5-minute interval closing prices from the 100 stocks in the *Nifty 100* index. Though limited in number, the time series are long and dense, ranging from 23,089 to 131,504 steps, with 88% exceeding 130,000 steps, making the dataset well-suited for fine-tuning. To enhance training diversity, we applied the TSMixup strategy from [1], which linearly combines pairs of input and target sequences. This yielded 45,000 augmented samples with lengths ranging between 128 and 1024 steps.

2) Synthetic data: To further diversify the training data, we generated 5,000 synthetic time series using Geometric Brownian Motions (GBMs). In the field of finance, it is customary to assume that stock prices behave as GBMs. The Black-Scholes model, for instance, is based on this hypothesis and is considered one of the most important models in the field. For this reason, we chose to generate our synthetic data using GBMs. A Geometric Brownian Motion S_t is described by the following Stochastic Differential Equation:

$$\mathrm{d}S_t = \mu S_t \,\mathrm{d}\mu + \sigma S_t \,\mathrm{d}W_t \tag{1}$$

where μ is called *drift*, σ is the *volatility* and W_t is a Brownian motion (or Wiener process). The following equation is a well-known solution of the SDE(1):

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right)$$

where S_0 denotes the starting price. To approximate a variety of processes we sampled $\mu \sim \mathcal{U}(-0.1, 0.1)$ and $\sigma \sim \mathcal{U}(0, 1))$, with starting prices sampled from $\mathcal{U}(0, 2000)$. We then chose an appropriate timestep (1/105,120 i.e. one over the number of 5 minutes intervals in a year) in order to create appropriate time series. To approximate W_t we sampled from $\mathcal{N}(0, \sqrt{\text{timestep}})$.

Remark. It's worth noting that Brownian motions are Gaussian processes, and similar behavior to GBMs could be produced using KernelSynth. However, we opted for GBMs based on domain knowledge to focus on the most relevant process for our application.

This hybrid training set combines real, augmented, and synthetic data to expose the model to a broader range of temporal dynamics.

C. Fine-Tuning Setup

We fine-tuned the *chronos-t5-tiny* model on a combined dataset consisting of real financial time series of nifty 100, augmented samples generated via TSMixup, and synthetic sequences based on Geometric Brownian Motions (GBMs). From the 100 original time series, 70 were used for training and 30 for testing. The 70 training series were augmented using TSMixup to produce 45,000 samples, and an additional 5,000 GBM-generated sequences were added, resulting in 50,000 total training samples. The test set consisted of the 30 remaining, unseen time series.

Remark. Notice that our testing approach was different than the in-domain method adopted in the Chronos paper. In fact, the authors used the same time series chosen for evaluation during the training, simply removing the last tokens used for the prediction length. We did not test it in this way, but instead our model never saw the time series on which it was tested. Therefore, our results are to be regarded as a conservative evaluation of the performance, since no information of the test series was directly available to the model before testing.

D. Evaluation and Results

To evaluate the model, we split each of the 30 test time series into 10 segments, resulting in 300 evaluation samples. For each segment, all preceding timesteps were used as input, while the final n tokens, referred to as the *prediction length*, were withheld and used for evaluation. Forecasting accuracy was measured using Mean Absolute Scaled Error (MASE) and Weighted Quantile Loss (WQL), following the protocol in [1]. Unlike the in-domain evaluation setup in the original Chronos paper—where the same time series were used for both training and testing by holding out the last k steps for evaluation—our test series were entirely unseen during training. This provides a more conservative estimate of real-world performance.

We compared outcomes across various prediction lengths. As shown in Figure 1, performance improvements were minimal for shorter horizons but became more pronounced with longer ones. For example, MASE improved by only 3% at a prediction length of 6 steps, but the gain reached 13% at 96 steps. WQL followed a similar trend, improving from 5% to 15% across the same range.

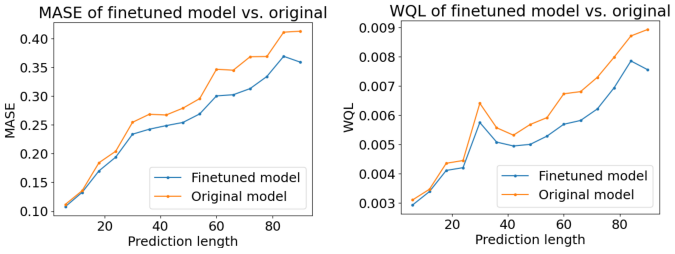


Fig. 1: Performance of the fine-tuned model relative to the original *chronos-t5-tiny* on the financial dataset, across varying prediction lengths.

E. Conclusion & Future Work

These results highlight that even with a relatively small number of time series from a different domain, fine-tuning can yield substantial performance gains, up to 15%, when adapting to a specialized context such as finance. Future work could explore alternative strategies for synthetic data generation and evaluate performance across both in-domain and zero-shot settings to better understand the trade-offs introduced by domain-specific fine-tuning.

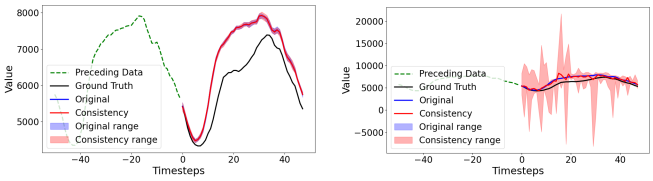
III. Extension 2: Consistency Calibration for Chronos(C3)

A. Overview

Time series forecasts often inform critical decisions. These decisions span a variety of real-world applications. For example, farmers may rely on temperature predictions to optimize irrigation. Hedge funds may use financial forecasts to guide asset trading. In such contexts, it is essential to estimate how likely a prediction is to be correct. This ability enables risk mitigation and supports more informed strategies. Calibration helps achieve this by aligning a model's confidence levels with its actual accuracy. For instance, a well-calibrated model that predicts a value with 80% confidence should be correct in approximately 80% of similar cases. To improve the calibration of Chronos, we apply the consistency calibration method introduced by Tao et al. [3], which perturbs model outputs to better align confidence estimates with accuracy.

Consistency Calibration (CC) is a post-hoc calibration method that improves model reliability by linking prediction confidence to robustness under perturbations. The underlying principle is that a model's confidence in a prediction should reflect its stability: confident predictions should remain unchanged under small perturbations, while uncertain ones should vary. To capture this behavior, CC applies controlled Gaussian noise to the model's internal representations (logits) and aggregates multiple perturbed outputs to estimate more reliable probabilities. This method is especially effective for classification-based models, as it captures uncertainty through sensitivity analysis, improving calibration without retraining or requiring additional labeled data.

Our approach applies perturbations by adding Gaussian noise directly to the model's logits. This technique, inspired by [3], simulates the effect of input perturbations by disturbing the model's internal representations. Figures 2a and 2b illustrate the impact on predictions for a sample time series. As observed, a small amount of noise has minimal effect on the predicted value range but significantly alters the output probabilities, thereby influencing calibration. In contrast, larger perturbations severely distort the forecast and increase the prediction error.



(a) $\sigma = 2$ (b) $\sigma = 8$

Fig. 2: Consistency calibration results on the first time series from the `monash_australian_electricity` dataset (same sample in both plots). The figure shows ground truth, original forecast, and calibrated predictions under Gaussian noise with $\sigma = 2$ (a) and $\sigma = 8$ (b), using 128 perturbations.

B. Methodology

Chronos treats time series forecasting as a sequence-to-sequence classification problem. Given a time series context window $\mathbf{x} \in \mathbb{R}^T$, the model predicts future values by generating discrete tokens from a fixed vocabulary. It transforms continuous time series values into discrete tokens by mean-scaling the input $\mathbf{x}$ and quantizing it into a fixed vocabulary $\mathcal{V}$, where $|\mathcal{V}| = 4096$. For a prediction horizon of length h, the model produces logits $\mathbf{z} \in \mathbb{R}^{h \times |\mathcal{V}|}$, which are then converted into token probabilities using the softmax function applied across the vocabulary dimension. To improve the calibration of Chronos predictions, we apply consistency calibration by adding Gaussian noise to the model's logit vectors and aggregating the resulting predictions.

1) Perturbation Process: For each timestep t, we generate R perturbed versions of the logit vector $\mathbf{z}_t \in \mathbb{R}^{|\mathcal{V}|}$:

$$\tilde{\mathbf{z}}_t^i = \mathbf{z}_t + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \quad i = 1, \ldots, R \tag{2}$$

where σ is the noise strength and $\mathbf{I}$ is the identity matrix.

2) Consistency Aggregation: For each perturbed logit vector $\tilde{\mathbf{z}}_t^i$, we select the most probable token index:

$$k_{i,t}^* = \arg\max(\tilde{\mathbf{z}}_t^i) \tag{3}$$

where $k_{i,t}^* \in \{1, \ldots, |\mathcal{V}|\}$ is the predicted token index. The consistency-calibrated probability of token k at timestep t is:

$$p_{\text{consistency}}(k \mid t) = \frac{1}{R} \sum_{i=1}^{R} \mathbb{1}[k_{i,t}^* = k] \tag{4}$$

where $\mathbb{1}[\cdot]$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

3) Calibration Evaluation: We evaluate calibration quality using the Expected Calibration Error (ECE), which quantifies the discrepancy between predicted confidence and empirical accuracy within confidence bins.

1) **Binning**: Divide the confidence interval $[0, 1]$ into M equal-width bins $B_1, B_2, \ldots, B_M$.
2) **Bin Statistics**: For each bin B_m, compute:

$$C_m = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

$$A_m = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}[\hat{y}_i = y_i]$$

where:

- C_m is the average predicted confidence for samples in bin B_m,
- A_m is the empirical accuracy in the same bin,
- $\hat{p}_i$ is the model's predicted confidence for sample i,
- $\hat{y}_i$ and y_i are the predicted and true labels, respectively.

3) **ECE**: The Expected Calibration Error is defined as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |A_m - C_m| \tag{5}$$

where N is the total number of predictions.

ECE reflects how well a model's confidence aligns with its actual accuracy. A calibrated model predicting with 80% confidence should be correct about 80% of the time. Lower ECE values indicate more reliable confidence estimates, which is especially important in settings where decisions depend on model uncertainty.

C. Experimental Setup

We conducted our experiments on the zero-shot datasets from the original Chronos benchmark, using the same forecast horizons per dataset (ranging from 8 to 30 steps).

Chronos typically generates predictions autoregressively, sampling one token at a time where each step conditions on the previously predicted one. This process is repeated 20 times per input series to produce multiple coherent forecast trajectories. However, due to hardware and memory constraints, we deviated from this approach. Instead, we performed a single forward pass per series and stored the logits for all forecast steps. After applying softmax, we generated 20 output sequences by independently sampling each time step without conditioning on previous predictions. This non-autoregressive approximation removed step-to-step dependencies but enabled us to scale up to 24,000 time series.

Afterwards, we applied consistency calibration (C3) by injecting Gaussian noise on the same logits and aggregated the perturbed outputs. From the resulting calibrated distributions, we sampled another set of 20 sequences per series using the same independent decoding method.

For evaluation, we measured forecasting accuracy using Mean Absolute Scaled Error (MASE) and Weighted Quantile

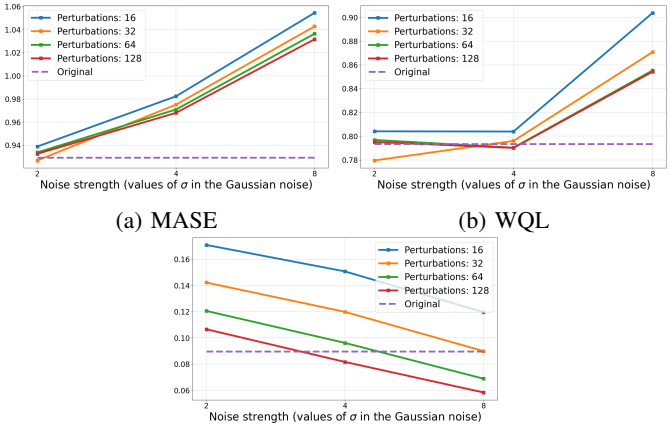
Loss (WQL), both computed relative to the seasonal naive baseline and aggregated across datasets using the geometric mean. However, because our generation procedure does not use autoregressive sampling, the resulting MASE and WQL scores are not directly comparable to those reported in the original Chronos paper.

Calibration quality was evaluated using Expected Calibration Error (ECE), also aggregated with the geometric mean. To study the effect of calibration strength, we tested various noise scales ($\sigma \in \{2, 4, 8\}$) and numbers of perturbations $R \in \{16, 32, 64, 128\}$.

D. Results

We evaluated Expected Calibration Error (ECE) on the zero-shot datasets of the Chronos paper, which exhibit greater predictive uncertainty and higher error rates. These datasets also better reflect real-world deployment conditions, as the time series were not seen during training.

Figures 3a, 3b, and 3c report results for MASE, WQL, and ECE across different values of noise strength (σ) and number of perturbations (R).



(a) MASE (b) WQL

(c) ECE

Fig. 3: Trends under perturbation for MASE, WQL, and ECE

As expected, increasing σ consistently reduced ECE but also increased forecast errors, revealing a trade-off between calibration and predictive accuracy. This trade-off can be further understood through the behavior of the noise distribution. As σ increases, the Gaussian perturbations approximate a locally uniform distribution, flattening the output probabilities. In theory, if all forecast bins were equally likely across the time series,though this is rarely true in practice, the ECE would approach zero. However, this would also imply that the model is no longer retaining meaningful information from the input and is effectively making random predictions. Thus, as the perturbations grow larger, the model becomes better calibrated but increasingly imprecise, asymptotically achieving perfect calibration at the cost of complete loss of forecasting accuracy.

Also, it can be seen that increasing R (the number of perturbations) helped stabilize results, reducing both ECE and forecasting error. However, due to computational limits, we could not reach convergence, as seen in the gap between $R = 64$ and $R = 128$.

Another important observation is shown in Figure 4, which plots ECE distributions across datasets for the naive and calibrated models with $\sigma = 2$ and $R = 128$. As seen in Figure 3, this level of noise does not significantly impact forecast errors. However, it noticeably reduces the variance in ECE values across datasets: the calibrated results cluster more tightly around 0.1. Specifically, consistency calibration increased ECE on smaller datasets with initially low values(below 0.5), while reducing it on those with higher baseline miscalibration. This indicates that calibration becomes more balanced and consistent across dataset scales, even with minimal perturbation.

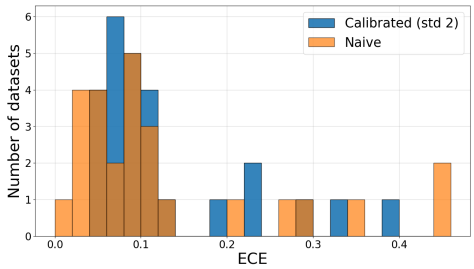


Fig. 4: ECE values across datasets for the naive and calibrated models, using $\sigma = 2$ and $R = 128$ perturbations.

E. Limitations and Future Work

Future work could explore generating distinct logits for each forecast step, rather than reusing a single forward pass, and consider dataset-specific or dynamic tuning of the noise scale σ. Additionally, we used top-k sampling with $k = 50$, following common LLM practice and the Chronos setup, but this choice may influence ECE and deserves further investigation. Due to computational constraints, we performed only one autoregressive pass per series, which likely contributed to the discrepancies in MASE and WQL compared to the original paper (Figures 3a and 3b). Investigating alternative sampling strategies such as temperature-controlled decoding may help improve both reliability and calibration.

References

[1] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, "Chronos: Learning the language of time series," 2024. [Online]. Available: https://arxiv.org/abs/2403.07815

[2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: https://arxiv.org/abs/1910.10683

[3] L. Tao, H. Guo, M. Dong, and C. Xu, "Consistency calibration: Improving uncertainty calibration via consistency among perturbed neighbors," 2024. [Online]. Available: https://arxiv.org/abs/2410.12295