Kronos: A Foundation Model for the Language of Financial Markets

Yu Shi[1]*, Zongliang Fu[2]*, Shuo Chen[1], Bohan Zhao[1], Wei Xu[1], Changshui Zhang[2], Jian Li[1]

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University
[2]Department of Automation, Tsinghua University
{shi-y23, fzl22, s-chen25, zhaobh23}@mails.tsinghua.edu.cn,
weixu@tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn, lapordge@gmail.com

Abstract

The success of large-scale pre-training paradigm, exemplified by Large Language Models (LLMs), has inspired the development of Time Series Foundation Models (TSFMs). However, their application to financial candlestick (K-line) data remains limited, often underperforming non-pre-trained architectures. Moreover, existing TSFMs often overlook crucial downstream tasks such as volatility prediction and synthetic data generation. To address these limitations, we propose **Kronos, a unified, scalable pre-training framework tailored to financial K-line modeling**. Kronos introduces a specialized tokenizer that discretizes continuous market information into token sequences, preserving both price dynamics and trade activity patterns. We pre-train Kronos using an autoregressive objective on a massive, multi-market corpus of over 12 billion K-line records from 45 global exchanges, enabling it to learn nuanced temporal and cross-asset representations. Kronos excels in a zero-shot setting across a diverse set of financial tasks. On benchmark datasets, Kronos boosts price series forecasting RankIC by 93% over the leading TSFM and 87% over the best non-pre-trained baseline. It also achieves a 9% lower MAE in volatility forecasting and a 22% improvement in generative fidelity for synthetic K-line sequences. These results establish Kronos as a robust, versatile foundation model for end-to-end financial time series analysis.

Code — https://github.com/shiyu-coder/Kronos

Extended version — https://arxiv.org/abs/2508.02739

Introduction

The emergence of Foundation Models (FMs) has initiated a paradigm shift across artificial intelligence, reshaping the methodologies of representation learning and downstream task adaptation. This shift is exemplified by the success of Large Language Models (LLMs) for natural language processing (Brown et al. 2020; Achiam et al. 2023), with parallel breakthroughs in computer vision (Radford et al. 2021; Kirillov et al. 2023; Yu et al. 2023).

Inspired by these advances, the FM paradigm has recently been extended to temporal data, giving rise to Time Series Foundation Models (TSFMs) (Garza, Challu, and Mergenthaler-Canseco 2023; Woo et al. 2024; Xiaoming et al. 2025). The central aim is to build pre-trained, task-agnostic architectures that serve as universal backbones for diverse time series analytical tasks—from forecasting and anomaly detection to causal inference—thereby substantially reducing the need for bespoke model design in each application domain.

Within this expanding research landscape, financial markets stand out as a critical and challenging application area for TSFMs, given their inherent data richness, high-frequency observations, and complex, non-stationary temporal dynamics. At the core of this domain are K-line sequences, multivariate time series derived from candlestick charts that record **O**pen, **H**igh, **L**ow, and **C**lose prices, along with trading **V**olume and **A**mount (Turnover) over fixed intervals (**OHLCVA**). These sequences constitute a highly compact, information-dense "language"

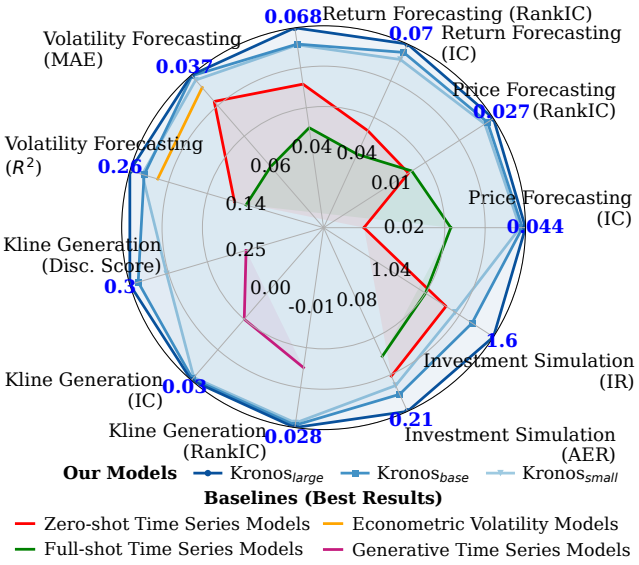

Figure 1: Comprehensive performance of Kronos across several quantitative finance tasks. The chart benchmarks our Kronos models (blue family) against several categories of specialized baselines. A greater distance from the center signifies superior performance.

*Equal contribution

through which market participants interpret price movements, volatility regimes, liquidity shifts, and collective sentiment (Nison 2001). Consequently, K-line data forms the bedrock of numerous algorithmic trading strategies, portfolio optimization schemes, and risk management systems.

However, applying general-purpose TSFMs to financial K-line data presents significant challenges, due to two principal factors. First, K-line sequences exhibit unique statistical properties—such as low signal-to-noise ratios, strong non-stationarities, and intricate, high-order dependencies among OHLCVA attributes (Zhang and Hua 2025; Baidya and Lee 2024)—that are often misaligned with the inductive biases of generic TSFMs. Second, the financial domain has largely been underserved by mainstream TSFM research; financial sequences constitute a minor fraction of pre-training corpora for most existing TSFMs (Das et al. 2024; Gao et al. 2024; Xiaoming et al. 2025; Goswami et al. 2024) , and the spectrum of downstream tasks critical to quantitative finance—spanning volatility estimation, synthetic sequence generation, and risk management—remains largely unaddressed. These factors lead to an important observation, which we empirically validate in this work: general-purpose TSFMs often underperform specialized, non-pre-trained models (e.g., iTransformer (Liu et al. 2023)) on financial tasks and fail to generalize across the broader landscape of quantitative finance.

To address these shortcomings, we introduce **Kronos, a unified, scalable pre-training framework designed specifically for financial K-line data**. Kronos employs a specialized tokenizer to discretize continuous, multivariate K-line inputs into a sequence of compact tokens, preserving critical price–volume interactions. It then undergoes autoregressive pre-training on an expansive, heterogeneous corpus of over 12 billion K-line records drawn from over 45 global markets and 7 temporal granularities.

We validate the efficacy of Kronos through comprehensive experiments across a range of quantitative finance tasks, with a high-level summary presented in Figure 1. On the core task of price series forecasting, Kronos establishes a new state-of-the-art, boosting the RankIC by 93% over the leading TSFM and by 87% over the best-performing non-pre-trained baseline. Furthermore, it demonstrates strong versatility by achieving a 9% lower MAE in volatility forecasting and a 22% improvement in generative fidelity for synthetic K-line generation. These findings highlight the broad effectiveness of our approach and underscore Kronos's potential as a robust foundation model for interpreting the complex "language" of financial markets.

Our main contributions can be summarized as follows:

- We propose a novel modeling framework for financial K-line data that learns hierarchical representations. It features a specialized tokenizer that quantizes each multivariate K-line record into structured, dual-component (coarse and fine) tokens, coupled with a tailored autoregressive objective that predicts these subtokens sequentially. This coarse-to-fine prediction scheme allows Kronos to explicitly model multi-scale market dynamics.
- We conduct large-scale pre-training for a family of Kronos models with varying capacities. This is performed on a massive, diverse financial corpus of over 12 billion K-line records from over 45 global exchanges, which is fundamental to learning the robust and generalizable market representations that underpin the models' effectiveness.
- We conduct comprehensive empirical evaluations across a set of quantitative finance tasks. Our results show that Kronos establishes a new state-of-the-art in price series forecasting, significantly outperforming both TSFMs and specialized baselines. The model's versatility is further demonstrated by its strong performance across a broader spectrum of quantitative tasks, including volatility forecasting and synthetic K-line generation.

Preliminary

Let D-dimensional vector $\mathbf{x}_t \in \mathbb{R}^D$ denote the K-line observation at discrete time t, comprising D key financial indicators. In this work, we fix the dimension $D = 6$ to represent OHLCVA attributes (Open, High, Low, Close prices, trading Volume, and Amount). Given a historical sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, our objective is to predict the following H observations $\hat{\mathbf{x}}_{T+1:T+H} = (\hat{\mathbf{x}}_{T+1}, \hat{\mathbf{x}}_{T+2}, \ldots, \hat{\mathbf{x}}_{T+H})$.

Rather than operating on raw continuous inputs, Kronos first quantizes each multivariate observation $\mathbf{x}_t$ into a discrete token b_t via a learnable codebook $\mathcal{C}$. Consequently, the original sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ is mapped to $\mathbf{b}_{1:T} = (b_1, \ldots, b_T)$. The forecasting task then reduces to an autoregressive token-sequence modeling problem:

$$p(\mathbf{b}_{T+1:T+H} \mid \mathbf{b}_{1:T}) = \prod_{h=1}^{H} p\big(b_{T+h} \mid \mathbf{b}_{1:T+h-1}\big). \quad (1)$$

Such a discrete formulation is inherently scalable and naturally extends to other tasks that can be framed generatively, such as synthetic data generation and volatility forecasting.

Methodology

Kronos abstracts financial K-line sequences as a discrete language and implements this via a two-phase framework illustrated in Figure 2: **(1) K-line Tokenization** and **(2) Autoregressive Pre-training**. In the first phase, we design a specialized Transformer-based tokenizer to quantize a continuous, multivariate K-line sequence into a corresponding sequence of discrete tokens, via a learnable codebook. Each K-line item (OHLCVA) is treated as an individual instance and quantized into a discrete token. Each token is composed of a coarse-grained subtoken and a fine-grained subtoken. This property is enforced via a hierarchical reconstruction loss, which explicitly compels the subtokens to model distinct levels of information, thereby creating a coarse-to-fine informational hierarchy. In the second phase, an autoregressive decoder-only Transformer is pre-trained on these tokenized sequences, using the standard next-token prediction objective to sequentially forecast both subtoken levels at each future time step conditioned on the given historical context. This unified *discretize-and-generate* paradigm enables Kronos to construct a high-fidelity, hierarchical representation of market dynamics, providing a robust foundation for downstream quantitative analysis.

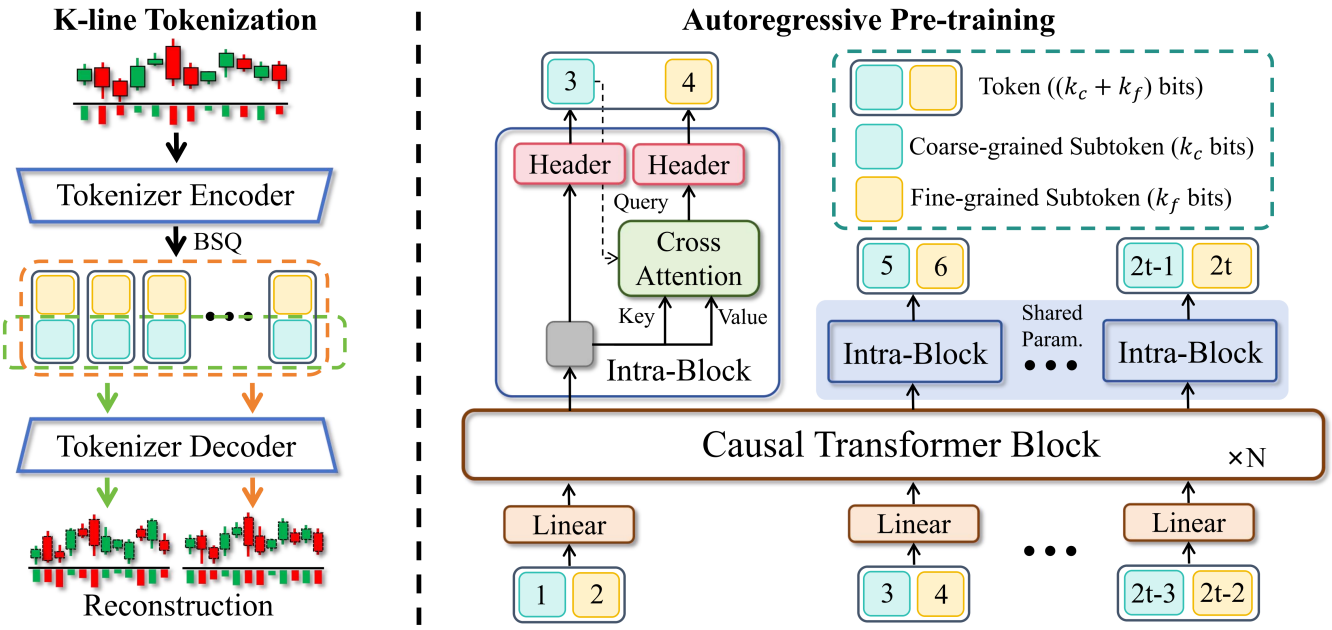

Figure 2: The two-stage framework of Kronos. **(1) Instance-based K-line Tokenization:** A Transformer-based autoencoder with a dual reconstruction objective quantizes continuous K-line data into a vocabulary of hierarchical discrete tokens, each comprising a coarse and a fine subtoken. **(2) Autoregressive Pre-training:** A decoder-only Transformer is pre-trained to model the temporal dynamics by sequentially predicting the hierarchical subtokens for the next time step, conditioned on the past.

K-line Tokenization

The first stage of Kronos transforms a continuous, D-dimensional K-line sequence $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^D$ encodes OHLCVA indicators, into a corresponding series of discrete tokens. This is achieved using a Transformer-based autoencoder (Figure 3) composed of an encoder E_{enc}, a quantizer Q, and a decoder E_{dec}. Drawing inspiration from video quantization methods in generative modeling (Van Den Oord, Vinyals et al. 2017; Yu et al. 2023), we adapt Binary Spherical Quantization (BSQ) (Zhao, Xiong, and Krähenbühl 2024), a variant of Look-up Free Quantization (LFQ) (Yu et al. 2023), for this task. BSQ quantizes a continuous latent vector $\boldsymbol{\xi}_t$ into a k-bit binary code $b_t \in \{-1, 1\}^k$ by projecting it onto a set of learnable hyperplanes. While a large number of bits k (e.g., $k = 20$) is desirable for capturing rich financial patterns, it results in an exponentially large vocabulary of size 2^k, which introduces significant challenges for the subsequent autoregressive model in terms of computational cost and parameter size. To mitigate this, we follow recent work in video quantization and generation (Yu et al. 2023; Wang et al. 2025) and factorize the k-bit code into n subspaces. Motivated by the trade-off between parameter savings and latency costs, we set $n = 2$. We partition the code into a coarse subtoken b_t^c and a fine subtoken b_t^f of equal bit length, $k_c = k_f = k/2$, where $k = k_c + k_f$. The resulting code b_t is a concatenation of these two subtokens: $b_t = [b_t^c, b_t^f]$, with $b_t^c, b_t^f \in \{-1, 1\}^{k/2}$. This decomposition transforms a single prediction over a large vocabulary of size 2^k into two sequential predictions over $2^{k/2}$ entries, substantially reducing both computational and parameter complexity.

To enforce a coarse-to-fine structure within each token, we train the tokenizer with a composite objective that combines a hierarchical reconstruction loss and a commitment loss for BSQ:

$$\mathcal{L}_{\text{tokenizer}} = \mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{fine}} + \lambda \mathcal{L}_{\text{quant}}, \tag{2}$$

where λ is a balancing hyperparameter. The components are defined as:

- $\mathcal{L}_{\text{coarse}} = \mathbb{E}\left[\|\mathbf{x} - E_{\text{dec}}(\mathbf{b}^c)\|^2\right]$, which trains the coarse subtoken $\mathbf{b}^c$ to form a low-fidelity reconstruction.
- $\mathcal{L}_{\text{fine}} = \mathbb{E}\left[\|\mathbf{x} - E_{\text{dec}}(\mathbf{b})\|^2\right]$, which evaluates the high-fidelity reconstruction using the complete token $\mathbf{b}$.
- $\mathcal{L}_{\text{quant}}$ is the quantization loss from BSQ (Zhao, Xiong, and Krähenbühl 2024) that regularizes the learning process. It penalizes the L2 distance between continuous latent vectors $\boldsymbol{\xi}$ and their binary codes $\mathbf{b}$, aligning the encoder's outputs with the learned codebook to ensure stable training.

This hierarchical reconstruction objective is central to our design. By optimizing $\mathcal{L}_{\text{coarse}}$, the coarse subtoken $\mathbf{b}^c$ learns to capture the principal structure of the input. Consequently, during the optimization of $\mathcal{L}_{\text{fine}}$, the fine-grained subtoken $\mathbf{b}^f$ is guided to encode the residual information required to refine the coarse approximation. Prior work has shown that a coarse-to-fine decoding order improves generation quality (Wang et al. 2025). Instead of identifying and prioritizing the decoding of tokens that inherently contain coarse information, our approach is designed to explicitly impose

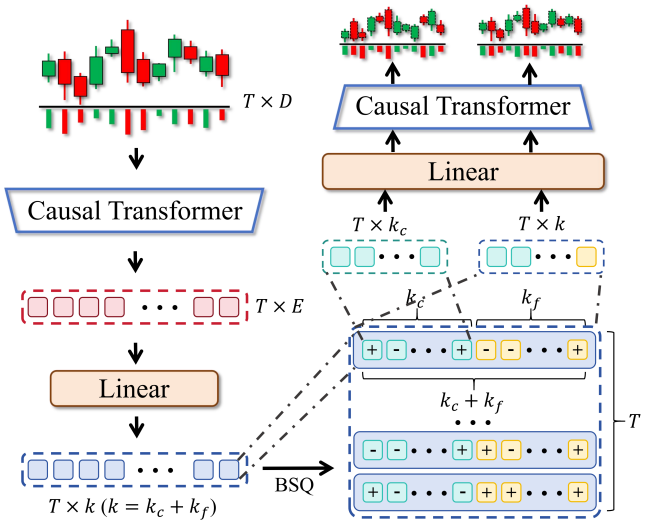

Figure 3: Architecture of the K-line Tokenizer. It employs a Transformer-based autoencoder with a Binary Spherical Quantization (BSQ) layer.

this hierarchy into the tokens during quantization. This ensures that the first subtoken consistently represents coarse-grained information, establishing the desired conditional dependency for the subsequent autoregressive modeling stage.

Hierarchical Autoregressive Modeling

Following the tokenization stage, the resulting discrete sequences are modeled using a decoder-only Transformer, denoted as E_{ar}, which employs causal-attention to ensure that predictions at each time step depend exclusively on historical context. The primary objective is to estimate the joint distribution over the token sequence $\mathbf{b} = \{b_1, \ldots, b_T\}$. A simplified form of Equation 1 can be derived as:

$$p(\mathbf{b}) = \prod_{t=1}^{T} p(b_t|\mathbf{b}_{<t}), \tag{3}$$

where $\mathbf{b}_{<t}$ denotes all preceding tokens up to time $t-1$.

Given the hierarchical token design, in which each token is structured as $b_t = [b_t^c, b_t^f]$, we further decompose the conditional probability using the chain rule to explicitly capture the inherent coarse-to-fine dependency:

$$p(b_t|\mathbf{b}_{<t}) = p(b_t^c|\mathbf{b}_{<t}) \cdot p(b_t^f|\mathbf{b}_{<t}, b_t^c). \tag{4}$$

This formulation allows the model to first predict the coarse-grained subtoken, which serves as a scaffold for subsequently generating the fine-grained residual subtoken. Consequently, the pre-training objective reduces to maximizing the log-likelihood of the observed sequence under this hierarchical factorization.

As depicted in Figure 2 (Right), the autoregressive process begins by constructing a unified input vector for each time step. Specifically, at time i, the subtokens b_i^c and b_i^f are independently projected into vector representations using two distinct embedding layers, resulting in representations $e_c(b_i^c)$ and $e_f(b_i^f)$, respectively. These embeddings are

	Layers	$\mathbf{d}_{\text{model}}$	$\mathbf{d}_{\text{ff}}$	Heads	Vocab. (2^k)	Params
Kronos$_{small}$	8	512	1024	8	20	24.7M
Kronos$_{base}$	12	832	2048	16	20	102.3M
Kronos$_{large}$	18	1664	3072	32	20	499.2M

Table 1: Model configurations for the Kronos family. We detail the number of Transformer layers, model dimension ($\mathbf{d}_{\text{model}}$), feed-forward dimension ($\mathbf{d}_{\text{ff}}$), number of attention heads, vocabulary size, and the total number of parameters.

then concatenated and linearly projected to produce a fused input vector:

$$\mathbf{v}_i = W_{\text{fuse}}([e_c(b_i^c); e_f(b_i^f)]), \tag{5}$$

where $[\cdot;\cdot]$ denotes concatenation, and W_{fuse} is a learnable weight matrix responsible for projecting the combined representation into the model's latent space.

The sequence of fused inputs $\{\mathbf{v}_1, \ldots, \mathbf{v}_{t-1}\}$ is then processed by the Transformer E_{ar}, which outputs contextualized hidden states. The final hidden state from processing $\mathbf{b}_{<t}$, denoted as $\mathbf{h}_t$, is then used to predict the token b_t. This hidden state subsequently informs the autoregressive predictions of both coarse and fine subtokens at the next step, thereby enabling the model to effectively capture multi-scale temporal dependencies inherent in the data.

Coarse Subtoken Prediction. The history vector $\mathbf{h}_t$ is projected by a linear head W_c to produce logits for the first subtoken's distribution:

$$p(b_t^c|\mathbf{b}_{<t}) = \text{softmax}(W_c\mathbf{h}_t) \tag{6}$$

Fine Subtoken Prediction. To model the conditional dependency in Equation (4), the context needs to be updated with the predicted coarse subtoken, $\hat{b}_t^c$. During training, we use the model's own prediction from the previous step, $\hat{b}_t^c$, which is sampled from the predicted distribution $p(b_t^c|\mathbf{b}< t)$, rather than using the ground-truth subtoken (i.e., teacher-forcing). We find that this sampling strategy enhances model robustness by mitigating exposure bias, better aligning the training distribution with the auto-regressive nature of multi-step inference where ground-truth tokens are unavailable. We use a cross-attention mechanism where the embedding of $\hat{b}_t^c$ acts as the query, and the history $\mathbf{h}_t$ provides the key and value. The result is projected by the second head W_f:

$$\begin{aligned} \mathbf{h}_t^{\text{update}} &= \text{CrossAttn}(q = e_c(\hat{b}_t^c), k = v = \mathbf{h}_t) \\ p(b_t^f|\mathbf{b}_{<t}, \hat{b}_t^c) &= \text{softmax}(W_f\mathbf{h}_t^{\text{update}}) \end{aligned} \tag{7}$$

The overall training objective $\mathcal{L}_{\text{ar}}$ is the negative log-likelihood of the data, summed over both prediction steps:

$$\mathcal{L}_{\text{ar}} = -\mathbb{E}_{\mathbf{b}\sim\mathcal{D}} \sum_{t=1}^{T} \left[\log p(b_t^c|\mathbf{b}_{<t}) + \log p(b_t^f|\mathbf{b}_{<t}, \hat{b}_t^c)\right] \tag{8}$$

where $\mathcal{D}$ represents the data distribution.

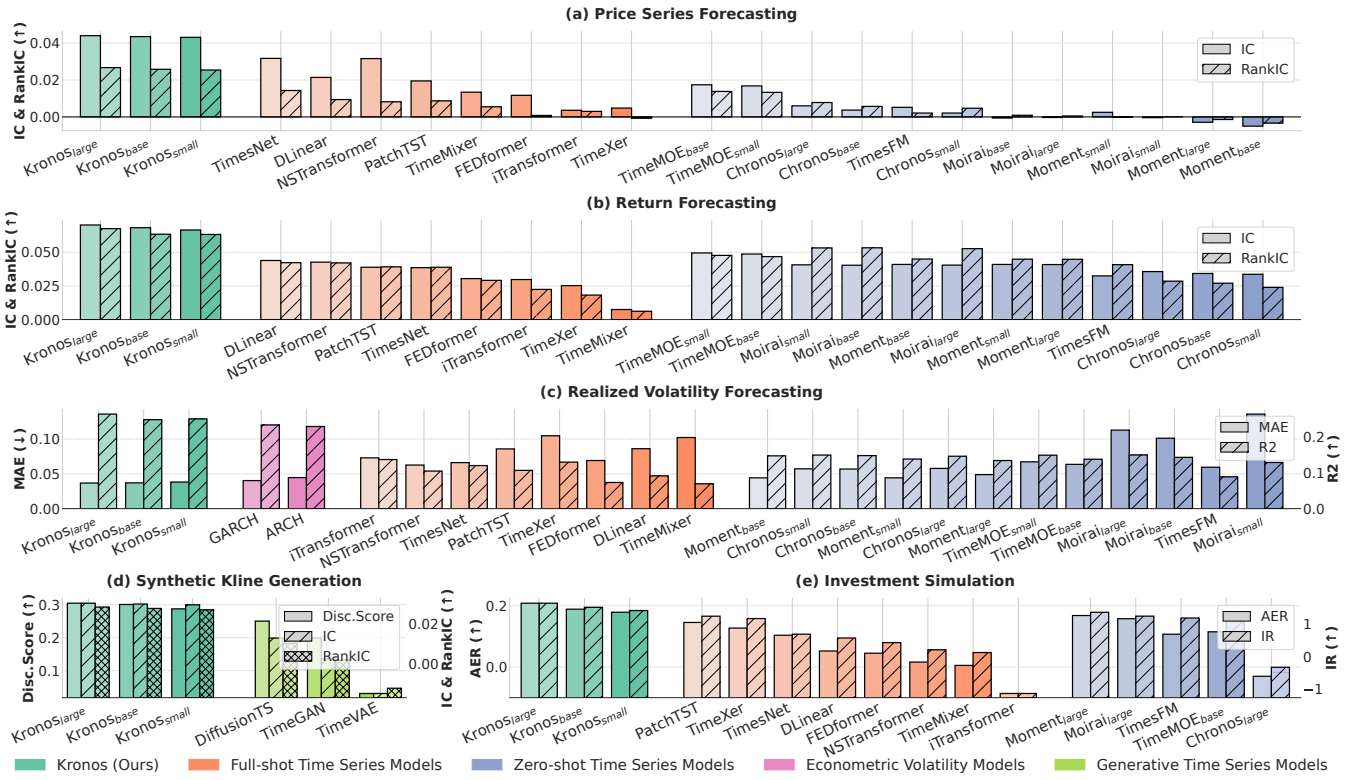

Figure 4: Main experimental results across five representative financial tasks. Subfigures (a-c) show forecasting performance on price series, returns, and realized volatility. Subfigure (d) displays generative model performance in terms of fidelity and usefulness. Subfigure (e) presents the investment simulation backtesting results.

Model Pre-training

Dataset To ensure the quality of pre-training, we curate a large-scale, high-quality financial K-line dataset from the ground up. In contrast to foundation-model research on generic time series—where well-curated public datasets are readily pooled—comprehensive, high-quality financial data remain limited. Our dataset spans over 12 billion observations across 7 sampling frequencies, encompassing a broad spectrum of asset classes drawn from 45 global exchanges. To guarantee data quality, we develop a streamlined data-cleaning pipeline tailored to the unique characteristics of financial K-line data, which identifies and filters out low-quality segments such as those with abnormal price spikes or prolonged periods of inactivity.

Model Training Informed by the scaling laws observed in LLMs (Kaplan et al. 2020), we trained three variants of Kronos with increasing parameter counts, up to nearly 0.5 billion, to provide a trade-off between performance and inference budget. The detailed model configurations are presented in Table 1. Considering resource constraints and practical deployment scenarios, we limit the maximum context length to 512 tokens. Nevertheless, this design remains fully compatible with arbitrary forecasting horizons by leveraging K-line data at varying frequencies; for instance, using 1-minute data for short-term forecasting and daily data for weekly or monthly predictions.

Inference At inference time, we generate future token sequences autoregressively, analogous to text generation. The stochasticity of this process is controlled via standard techniques like temperature scaling and top-p (nucleus) sampling (Holtzman et al. 2019). The probability of sampling token i from logits $\mathbf{z}$ is given by $p_i \propto \exp(z_i/T)$, where T is the temperature. For tasks requiring high precision, prediction accuracy can be enhanced by generating multiple future trajectories (i.e., Monte Carlo rollouts) and averaging the decoded continuous values to produce a more stable forecast. As demonstrated in our experiments, this approach consistently improves forecast quality.

Experiments

To comprehensively evaluate the capabilities of Kronos as a foundation model for financial K-line data, we design a suite of experiments spanning 5 representative tasks. These tasks are selected to evaluate Kronos's performance in both predictive and generative applications, thereby demonstrating its versatility in practical quantitative finance scenarios.

Experimental Setup

The experimental tasks span predictive applications (price series, return and realized volatility forecasting), generative capabilities (synthetic K-line generation), and an investment simulation to gauge real-world applicability.

Model	Prediction Space	Training Objective	Price Series Forecasting		Return Forecasting		Volatility Forecasting	
			IC (↑)	RankIC (↑)	IC (↑)	RankIC (↑)	MAE (↓)	R^2 (↑)
Direct-AR	Continuous	Mean Squared Error (MSE)	0.0212	0.0149	0.0416	0.0399	0.0565	0.1608
Prob-AR	Continuous	Negative Log-Likelihood (NLL)	0.0179	0.0102	0.0356	0.0329	0.0464	0.1383
Kronos-Parallel	Discrete	Cross-Entropy	0.0345	0.0226	0.0529	0.0505	0.0461	0.1784
Kronos$_{small}$	**Discrete**	**Cross-Entropy**	**0.0431**	**0.0254**	**0.0665**	**0.0622**	**0.0384**	**0.2490**

Table 2: Ablation study dissecting the architectural choices of Kronos. We compare our model against variants targeting different **Prediction Spaces** (continuous vs. discrete) with corresponding **Training Objectives**. *Direct-AR* serves as a standard regression baseline. *Prob-AR* evaluates the benefit of probabilistic modeling in the continuous space. *Kronos-Parallel* ablates our sequential subtoken design by predicting subtokens concurrently. Best results are in **bold**.

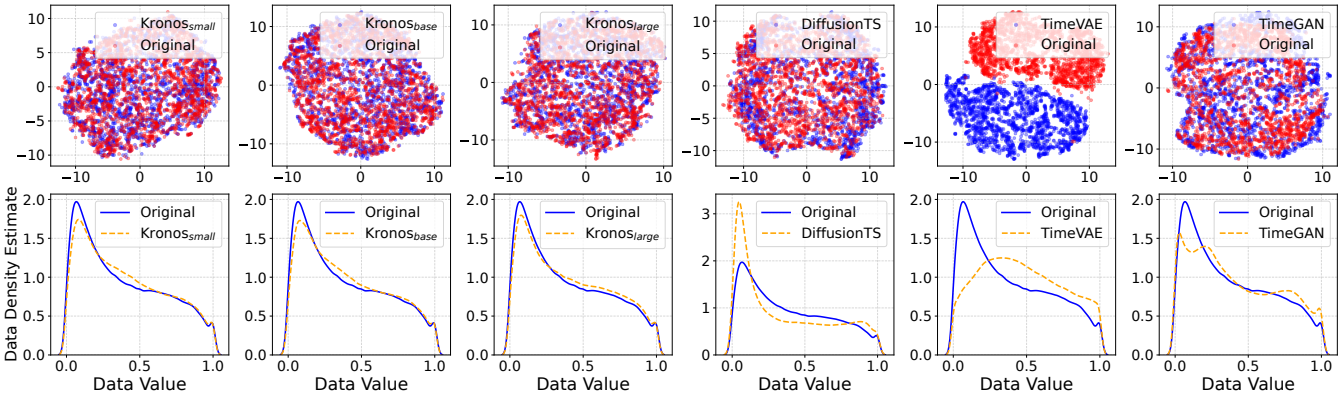

Figure 5: Visual comparison of generative models on the dataset of Shanghai Stock Exchange, 15-minute frequency. **Top row:** t-SNE embeddings of original (red) versus synthetic (blue) data. **Bottom row:** Kernel Density Estimates (KDE) of original versus synthetic data.

For a rigorous comparison, we benchmark Kronos against a comprehensive suite of 25 baseline models. These baselines are carefully selected to represent the state-of-the-art across four distinct paradigms: non-pre-trained full-shot models (e.g., iTransformer (Liu et al. 2023), DLinear (Zeng et al. 2023)), zero-shot time series foundation models (e.g., TimeMOE (Xiaoming et al. 2025), Chronos (Ansari et al. 2024)), econometric volatility models (e.g., GARCH (Bollerslev 1986), classical approaches for volatility prediction from econometrics), and generative time series models (e.g., DiffusionTS (Yuan and Qiao 2024)). An overview of our main experimental results is presented in Figure 4.

Main Results

Prediction Tasks Figure 4(a-c) presents the results for the three forecasting tasks. Kronos achieves consistent state-of-the-art performance across all of them. In particular, for price series forecasting, Kronos achieves a remarkable 93% improvement in RankIC compared to the strongest TSFM baseline, and an 87% gain over the best non-pre-trained model. Furthermore, as the model size scales up, performance on these tasks consistently improves, empirically validating the scaling laws for time series foundation models (Yao et al. 2024).

Generative Tasks Following established practices (Yoon, Jarrett, and Van der Schaar 2019), we evaluate the quality of synthetic data from three perspectives: *diversity*, *fidelity*, and *usefulness*. To assess *diversity*—how well generated samples cover the real data's distribution—we use two visual methods: projecting original and synthetic data into a 2D space using t-SNE, and comparing their distributions via kernel density estimation (KDE). As shown in Figure 5, the t-SNE plots show that Kronos's synthetic data better overlaps the original data space, and the KDE plots confirm a higher similarity in distributions.

For quantitative evaluation, we assess *fidelity* (i.e., data realism) using the discriminative score, which measures how difficult it is for a classifier to distinguish between original and synthetic samples. We also evaluate *usefulness* (the synthetic data's effectiveness for training downstream models) via the Train-on-Synthetic, Test-on-Real (TSTR) protocol, where a forecasting model is trained on synthetic data and its resulting IC and RankIC are evaluated on a test set composed of real data. As shown in Figure 4(d), Kronos achieves the best performance in both *fidelity* and *usefulness*. This superiority is also enhanced as the model size scales.

Investment Simulation To validate Kronos's performance in a realistic investment scenario, we simulate a long-only investment strategy on the Chinese A-shares market by constructing portfolios with the top-k stocks ranked by each model's predictive signals. As shown in Figure 4(e), Kronos outperforms all other baselines, achieving the highest Annualized Excess Return (AER) and Information Ratio (IR).

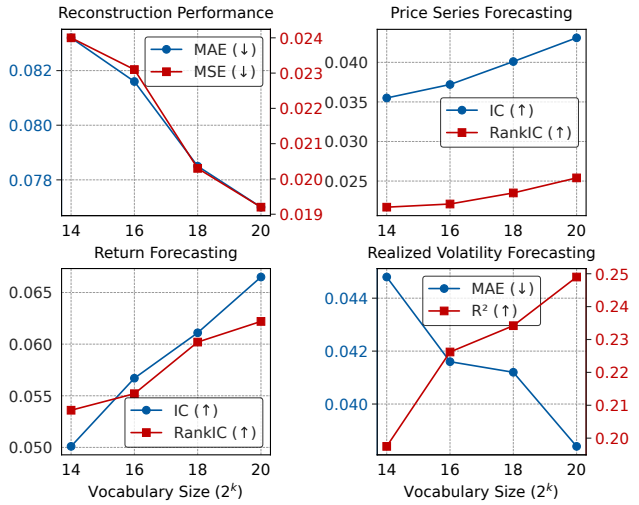

Figure 6: Impact of vocabulary size on model performance. We plot reconstruction quality and downstream forecasting performance as vocabulary size increases.

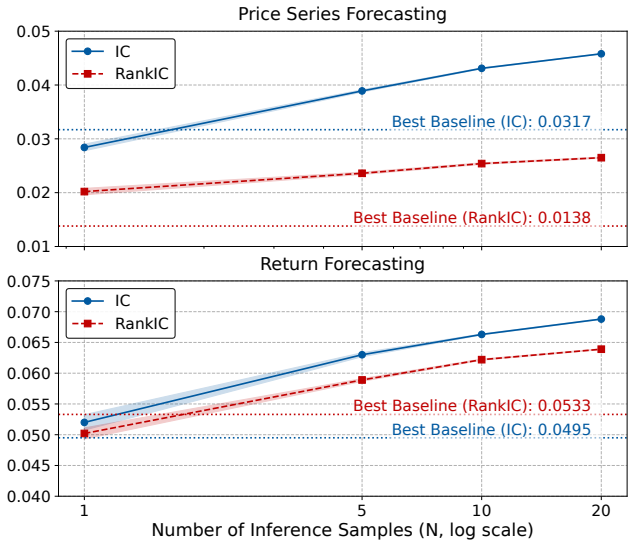

Figure 7: Impact of the number of inference samples (N) on forecasting performance.The lines represent the mean performance over 5 runs with different random seeds, while the shaded areas indicate the standard deviation.

This demonstrates that the model can effectively translate its superior predictive accuracy into tangible investment gains.

Ablation Study

We conduct ablation studies to validate our core design choices, focusing on two questions: (Q1) the effectiveness of our modeling paradigm compared to other alternatives, and (Q2) the impact of vocabulary size.

Analysis of Modeling Paradigms. To address Q1, we compare Kronos against variants that differ in their prediction spaces and objectives, while maintaining comparable parameter counts. (Table 2). We test two continuous-space models: *Direct-AR* (a regression baseline with MSE) and *Prob-AR*. Following established work (Yao et al. 2024), *Prob-AR* uses a Student-t mixture distribution to better model heavy-tailed data distributions. The results show that our discrete-space models markedly outperform these continuous alternatives.We also find that *Kronos-Parallel*, a variant that predicts subtokens concurrently, performs worse than our sequential approach, demonstrating the importance of modeling subtoken dependencies. These findings validate our discrete, sequential modeling framework as a more effective approach for this domain.

Impact of Vocabulary Size. To answer Q2, we investigate how vocabulary size affects model performance. As shown in Figure 6, increasing the vocabulary size improves both reconstruction quality and forecasting accuracy. A larger vocabulary provides a finer-grained representation, reducing quantization error. Crucially, this enhanced representational precision translates to better predictive outcomes. This finding aligns with observations in video generation, where for quantization techniques like LFQ and BSQ, larger vocabularies have been shown to lead to improved generation quality (Zhao, Xiong, and Krähenbühl 2024; Yu et al. 2023).

Test-Time Scaling

A notable advantage of our probabilistic, generative framework is the ability to enhance predictive accuracy at inference time without retraining the model. By leveraging stochastic sampling, Kronos can generate multiple distinct future trajectories from the same context. We investigate the effect of ensembling these predictions by averaging the outcomes from an increasing number of sampled paths. Figure 7 presents the performance on forecasting tasks as a function of the number of samples. The results demonstrate a consistent improvement in both IC and RankIC as more samples are included in the ensemble. Averaging across multiple paths mitigates the stochasticity inherent in the generation process and reduces prediction variance, yielding a more robust and stable estimate. This capability offers a trade-off, allowing practitioners to balance computational cost at inference with desired levels of predictive accuracy.

Conclusion

In this work, we introduce Kronos, a foundation model specifically designed for financial K-line sequences. Kronos employs a novel two-stage framework, where an instance-based tokenizer first discretizes continuous market data into hierarchical coarse-to-fine tokens, which are then modeled by a large autoregressive Transformer. Comprehensive empirical evaluations demonstrate that Kronos establishes new state-of-the-art benchmarks in price series forecasting, as well as in other relevant applications such as synthetic K-line generation and volatility forecasting, significantly outperforming existing TSFMs and other baselines. These results position Kronos as a robust and versatile foundation for a range of applications in quantitative finance.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Baidya, R.; and Lee, S.-W. 2024. Addressing the Non-Stationarity and Complexity of Time Series Data for Long-Term Forecasts. *Applied Sciences*, 14(11): 4436.

Bollerslev, T. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3): 307–327.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

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

Gao, S.; Koker, T.; Queen, O.; Hartvigsen, T.; Tsiligkaridis, T.; and Zitnik, M. 2024. Units: Building a unified time series model. *arXiv e-prints*, arXiv–2403.

Garza, A.; Challu, C.; and Mergenthaler-Canseco, M. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589*.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Nison, S. 2001. *Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East*. Penguin.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Wang, Y.; Lin, Z.; Teng, Y.; Zhu, Y.; Ren, S.; Feng, J.; and Liu, X. 2025. Bridging continuous and discrete tokens for autoregressive visual generation. *arXiv preprint arXiv:2503.16430*.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *International Conference on Machine Learning*, 53140–53164. PMLR.

Xiaoming, S.; Shiyu, W.; Yuqi, N.; Dianqi, L.; Zhou, Y.; Qingsong, W.; and Jin, M. 2025. Time-MoE: Billion-Scale Time Series Foundation Models with Mixture of Experts. In *ICLR 2025: The Thirteenth International Conference on Learning Representations*. International Conference on Learning Representations.

Yao, Q.; Yang, C.-H. H.; Jiang, R.; Liang, Y.; Jin, M.; and Pan, S. 2024. Towards Neural Scaling Laws for Time Series Foundation Models. *arXiv preprint arXiv:2410.12360*.

Yoon, J.; Jarrett, D.; and Van der Schaar, M. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.

Yu, L.; Lezama, J.; Gundavarapu, N. B.; Versari, L.; Sohn, K.; Minnen, D.; Cheng, Y.; Birodkar, V.; Gupta, A.; Gu, X.; et al. 2023. Language Model Beats Diffusion–Tokenizer is Key to Visual Generation. *arXiv preprint arXiv:2310.05737*.

Yuan, X.; and Qiao, Y. 2024. Diffusion-ts: Interpretable diffusion for general time series generation. *arXiv preprint arXiv:2403.01742*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, 9, 11121–11128.

Zhang, L.; and Hua, L. 2025. Major Issues in High-Frequency Financial Data Analysis: A Survey of Solutions. *Mathematics*, 13(3): 347.

Zhao, Y.; Xiong, Y.; and Krähenbühl, P. 2024. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*.