Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

QuantBench: benchmarking AI methods for quantitative investment from a full pipeline perspective

Saizhuo WANG[†1], Hao KONG[†5], Jiadong GUO[1], Fengrui HUA[3], Yiyan QI[2], Wanyun ZHOU[3], Jiahao ZHENG[2], Xinyu WANG[4], Lionel M. NI[3], Jian GUO[†‡2]

[1]*Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong 999077, China*

[2]*International Digital Economy Academy, Shenzhen 518045, China*

[3]*Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 518055, China*

[4]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*

[5]*Department of Information Systems, Business Statistics and Operations Management, The Hong Kong University of Science and Technology, Hong Kong 999077, China*

[†]E-mail: swangeh@connect.ust.hk; hkongab@connect.ust.hk; guojian@idea.edu.cm

Received Apr. 29, 2025; Revision accepted Oct. 9, 2025; Crosschecked Oct. 24, 2025; Published online Nov. 6, 2025

Abstract: The field of artificial intelligence (AI) in quantitative investment has seen significant advancements, yet it lacks a standardized benchmark aligned with industry practices. This gap hinders research progress and limits the practical application of academic innovations. We present QuantBench, an industrial-grade benchmark platform designed to address this critical need. QuantBench offers three key strengths: (1) standardization that aligns with quantitative investment industry practices; (2) flexibility to integrate various AI algorithms; (3) full-pipeline coverage of the entire quantitative investment process. Our empirical studies using QuantBench reveal some critical research directions, including the need for continual learning to address distribution shifts, improved methods for modeling relational financial data, and more robust approaches to mitigate overfitting in low signal-to-noise environments. By providing a common ground for evaluation and fostering collaboration between researchers and practitioners, QuantBench aims to accelerate progress in AI for quantitative investment, similar to the impact of benchmark platforms in computer vision and natural language processing. The code is open-sourced on GitHub at https://github.com/SaizhuoWang/quantbench.

Key words: Benchmark; Quantitative investment; Deep learning; Foundation models
https://doi.org/10.1631/FITEE.2500280 **CLC number:** TP181

1 Introduction

Artificial intelligence (AI) techniques—including traditional search-based methods (such as genetic programming, machine-learning, and deep-learning approaches) and more recently, generative AI (GenAI) methods based on foundation models—have been widely applied to quantitative investment (hereafter referred to as "quant"). Although research communities have extensively studied AI applications in quant and developed many new trading algorithms in recent years, there remains a lack of standardized benchmarks that align with the evaluation practices used in the quant industry and by trading firms. The absence of a unified benchmark dataset and testing pipeline, combined with the diverse and often inconsistent evaluation standards across existing research papers, may impede research progress and limit the real-world

‡ Corresponding author
ORCID: Saizhuo WANG, https://orcid.org/0000-0001-8175-8451; Jian GUO, https://orcid.org/0009-0003-5046-2588

applicability of new methods. In contrast, the establishment of standardized benchmarks, as seen in computer vision (Deng J et al., 2009) and natural language processing (Wang A et al., 2019), has been instrumental in accelerating research and enabling practical advancements in those fields.

To address this need, we propose QuantBench, an industrial-grade benchmark platform that offers universality and comprehensive pipeline coverage: (1) standardization (QuantBench adheres to research standards that are consistent with industrial practices in quant); (2) flexibility (QuantBench is designed to support the scalable integration of various AI algorithms into the system); (3) full-pipeline coverage (QuantBench encompasses the entire pipeline of general quant strategies, offering broad support for standardized datasets, model implementations, and evaluations).

Specifically, QuantBench covers a general quant research pipeline, integrating diverse tasks and learning paradigms into a single workflow (Fig. 1). By incorporating a broad array of AI methods, QuantBench bridges the gaps created by the segmented nature of the field, enhances reproducibility through open-source implementations, and facilitates the integration of advancements across disciplines. In addition, QuantBench emphasizes the importance of a unified dataset, constructing datasets with both breadth and depth, while maintaining consistency in data format across varied data types. Moreover, QuantBench offers a detailed market simulation framework that can vary in realism depending on the specific trading scenario. This robust evaluation framework accommodates quant-specific metrics that are both task-specific (e.g., signal and portfolio performance) and task-agnostic (e.g., alpha correlation and decay). By aligning evaluation metrics with tasks, QuantBench ensures that the chosen metrics capture the intricacies of these tasks, thereby enhancing the relevance of the findings. Additionally, it pays careful attention to task-agnostic metrics to address the unique challenges presented by financial data, including low signal-to-noise ratios and nonstationarity.

The contributions of QuantBench can be summarized as follows:

1. QuantBench enhances research efficiency by alleviating researchers from time-consuming preprocessing tasks, enabling them to focus on critical algorithmic innovations.

2. QuantBench serves as a unified platform. Industry practitioners can leverage this platform to rapidly implement state-of-the-art algorithms in their investment strategies. Conversely, QuantBench amplifies the impact of academic research by facilitating its practical application.

3. Through the standardization provided by QuantBench, communication between academia and industry is improved, thereby accelerating the progress in the field of AI for quant.

4. Our empirical studies using QuantBench identify several compelling research directions with significant potential value.

2 Quant pipeline

QuantBench provides a unified framework to evaluate the entire quant research pipeline, as illustrated in Fig. 1. It covers key phases of quant research, from raw data preparation to final trading

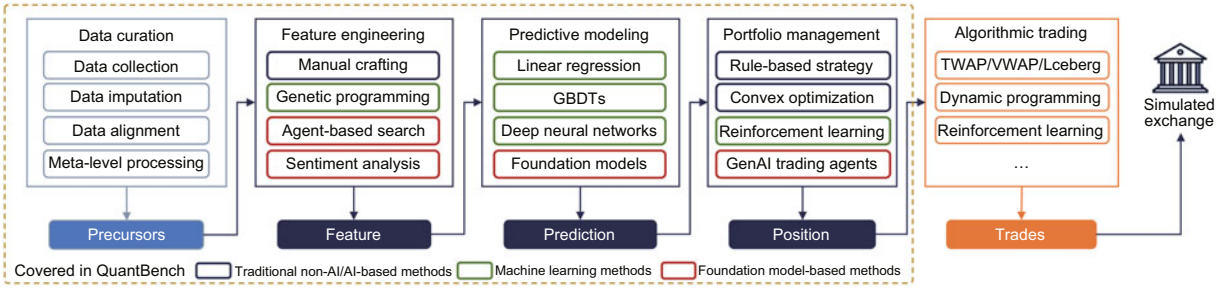

Fig. 1 Quant research pipeline covered in QuantBench. Data curation is implemented as built-in features, and methods for feature engineering, predictive modeling, and portfolio management are evaluated in QuantBench. Methods supported in QuantBench are categorized as traditional non-AI and AI-based methods, machine learning methods, and foundation model-based methods. References to color refer to the online version of this figure

execution, as follows:

1. Data curation. This phase involves preparing the raw data used throughout the pipeline. It includes data collection, imputation, alignment, and meta-level processing to ensure consistency and usability across different stages of research. QuantBench provides built-in implementations to automate these essential preprocessing steps.

2. Feature engineering. This phase focuses on extracting predictive features from curated financial data. In QuantBench, feature engineering supports both traditional manual methods and AI-driven approaches. Manual crafting allows researchers to design domain-specific factors, while genetic programming (Zhang TP et al., 2020; Cui et al., 2021) automates the discovery of symbolic expressions from raw data. In addition, agent-based search methods (Yu et al., 2023) leverage autonomous exploration strategies to discover interpretable features, and sentiment analysis integrates textual and news data as auxiliary signals.

3. Predictive modeling. In this phase, models are trained to predict market behaviors, including asset returns (regression), market movements (classification) (Koa et al., 2024), or asset rankings (ranking) (Feng et al., 2019). QuantBench supports a wide range of modeling techniques, spanning traditional linear models, machine learning algorithms like gradient boosted decision trees (GBDTs) and deep neural networks (DNNs), and newer foundation model-based approaches.

4. Portfolio management. Based on model predictions, this stage determines optimal portfolio allocations aligned with investor objectives. QuantBench supports rule-based strategies (e.g., characteristic-sorted portfolios (Cattaneo et al., 2020)), convex optimization formulations (e.g., mean-variance optimization (Markowitz, 1952, 1959)), and learning-based methods including reinforcement learning. Furthermore, GenAI trading agents are incorporated to autonomously design and manage trading strategies with dynamic adaptation.

5. Algorithmic trading. The final stage involves executing trades efficiently in a simulated exchange environment. QuantBench currently supports a variety of classical execution strategies such as time-weighted average price (TWAP), volume-weighted average price (VWAP), and iceberg orders, as well as dynamic programming and reinforcement learning approaches for optimal execution.

As summarized in Table 1, QuantBench offers end-to-end coverage across data modalities (market data: M; graph data: G), model classes (time-series model: T; spatiotemporal model: S), training paradigms (supervised learning: SP; reinforcement learning: RL), and tasks (stock prediction: Pred; portfolio optimization: PO), providing broader pipeline coverage than Qlib, TradeMaster, and FinRL-Meta among the compared benchmarks.

System architecture: QuantBench is structured in a layered and modular manner, as shown in Fig. 2, closely following the workflow outlined above. At the foundation, the bottom layer aggregates a diverse set of datasets and models, providing comprehensive support for different types of financial data and modeling needs. The middle layer processes model outputs, integrates feedback mechanisms, and applies standardized evaluation metrics, enabling fair comparisons and systematic model improvements. At the top, QuantBench specifies learning objectives and evaluation criteria aligned with practical financial goals such as utility maximization and PO. Training and evaluation processes are organized into clear pathways.

1. Training: model + data $\rightarrow$ output $\rightarrow$ learning objective $\rightarrow$ feedback $\rightarrow$ updated model.

2. Evaluation: model + data $\rightarrow$ output $\rightarrow$ evaluation module $\rightarrow$ performance metrics.

Through this architectural design, QuantBench advances beyond prior benchmarks by unifying the full quant pipeline—from factor mining and modeling to PO and execution—within a consistent architecture. As shown in Table 2 and Fig. 2, diverse tasks can be expressed in a uniform form of input, output, objective, feedback, and evaluation. This integration not only broadens coverage, but also offers a novel perspective for comparing AI methods across paradigms within a single benchmark.

Table 1 Comparison with other quant benchmarks

Platform	Data		Model		Training		Task	
	M	G	T	S	SP	RL	Pred	PO
Qlib	✓	✓	✓		✓		✓	
TradeMaster	✓		✓			✓		✓
FinRL-Meta	✓		✓			✓		✓
QuantBench	✓	✓	✓	✓	✓		✓	✓

Table 2 Comparison of different tasks covered in QuantBench

Task	Input	Output	Objective	Feedback	Evaluation
Factor mining	Data	Features	Regression	Reward	Signal
Modeling	Features	Prediction	Regression/Classification/Ranking	Gradient	Signal
End-to-end modeling	Features	Position	Utility maximization	Gradient	Portfolio
Portfolio optimization	Prediction	Position	Utility maximization	Reward	Portfolio
Order execution	Position	Trade	Utility maximization	Reward	Execution

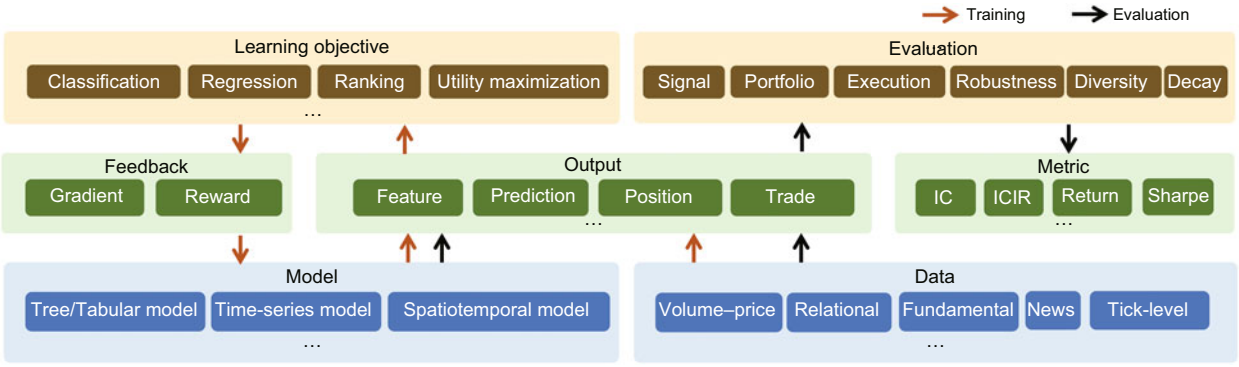

Fig. 2 Layered architecture of QuantBench. ICIR: information coefficient information ratio

3 Benchmark design

3.1 Dataset

Dataset serves as the fundamental source of information for quantitative predictions and decision-making in financial markets. The development of robust datasets in quantitative finance follows two primary directions: increasing breadth and enhancing depth. QuantBench addresses both aspects comprehensively with the data processing pipeline shown in Fig. 3. Increasing breadth involves incorporating diverse information sources, while enhancing depth focuses on improving data granularity. For instance, in low-frequency scenarios, such as monthly portfolio rebalancing, the integration of alternative data sources (e.g., satellite imagery or credit card transaction data) can provide a more comprehensive context for each data point. Conversely, enhancing depth, such as transitioning from daily to minute-level stock price data, allows for the detection of subtle patterns and trends, thereby potentially improving trading outcomes through the preservation of more detailed information.

3.1.1 Increasing data breadth

The expansion of information sources is crucial for developing a comprehensive view of the market. By integrating diverse data types, it becomes possible to uncover latent relationships and enhance predictive accuracy. QuantBench incorporates the following information sources:

1. Market data. These include price and volume data for stocks, options, and other financial instruments. These data form the basis for many technical analysis strategies, and can reveal trends in market sentiment. In QuantBench, we cover both aggregated bars (e.g., open, high, low, close, volume (OHLCV)) at regular time intervals and tick-level trade/quote data at irregular timesteps. Our starting point of market data ranges from 2003 to 2006 and the current cutoff is May 2024.

2. Fundamental data. These include financial statements, earnings reports, and other company-specific information. These data provide insight into a company's underlying value and growth prospects. In QuantBench, we collect fundamental data from the income statements, balance sheets, and statements of cash flows from publicly listed companies, and construct 21 built-in features out of these data.

3. Relational data. These capture the interactions between different entities, such as supply chain relationships or corporate ownership structures. We collect relational data from Wikidata by performing entity alignments based on company names and ticker symbols. We also construct a fully

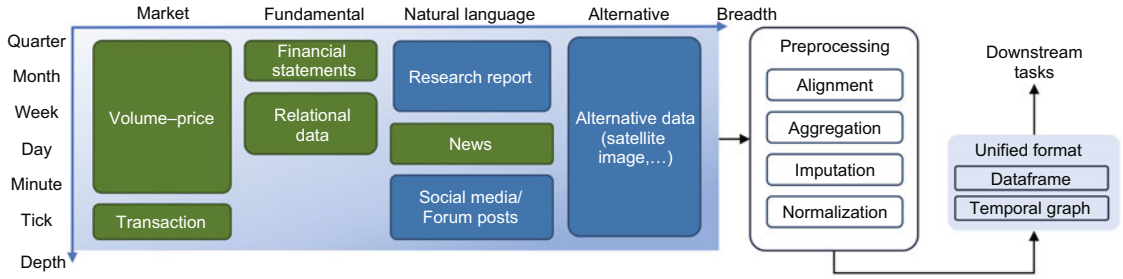

Fig. 3 Data processing pipeline of QuantBench. Blocks with green background are already supported in QuantBench, and blocks with blue background are planned to be supported in the future. References to color refer to the online version of this figure

connected graph/hypergraph from industrial categorizations following the global industry classification standard (GICS) categorization. Notably, to avoid future information leakage, we take the temporal information of relations from Wikidata into account, and support graph snapshots at any given timestamp. The statistics on relational data are shown in Table 3.

4. News. These data, including articles, social media posts, and press releases, can provide real-time information on market-moving events and shifts in sentiment. In QuantBench, we collect both the original news contents and the processed numerical features such as news count and normalized sentiment scores.

Furthermore, QuantBench incorporates a diverse range of markets, universes, and feature sets to support more granular analysis:

1. Markets and universes. As shown in Table 4, the dataset spans a range of markets, from those heavily influenced by automated trading to emerging markets, each exhibiting distinct trading patterns and dynamics. In markets dominated by automated trading, price adjustments tend to be swift and the trading patterns intricate, reflecting the rapid decision-making processes. Conversely, emerging markets often display slower price dynamics influenced significantly by individual investors. This diversity allows QuantBench to evaluate model performance across various market behaviors. Additionally, stocks are categorized into large market capitalization (large-cap), medium market capitalization (mid-cap), and small market capitalization (small-cap) segments, each presenting unique characteristics. For example, large-cap stocks generally exhibit stability, contrasting with the higher volatility and

Table 3 Statistics of relational data on stocks

Country/Region	Number of stocks	Wikidata				Industry	
		N_{Node}	Ratio (%)	$N_{\text{1-hop}}$	$N_{\text{2-hop}}$	Industry number	Average degree
Chinese mainland	5128	728	14.20	23	4809	30	159.87
US	6375	1775	27.84	219	41 401	10	585.63
UK	2556	215	8.41	10	9805	10	207.08
Hong Kong	2688	1367	50.86	98	6995	10	217.5

The statistics are taken with respect to the full stock universe on the regional market. N_{Node}: number of nodes; $N_{\text{1-hop}}$: number of 1-hop edges; $N_{\text{2-hop}}$: number of 2-hop edges

Table 4 Major regions and stock universes involved in QuantBench on the country/region market

Market	Chinese mainland		US		Hong Kong		UK	
	Name	N_{stock}	Name	N_{stock}	Name	N_{stock}	Name	N_{stock}
Large-cap	CSI 300	300	S&P 500	500	HSLI	30	FTSE 100	100
Mid-cap	CSI 500	500	S&P 400	400	HSMI	50	FTSE 250	250
Small-cap	CSI 1000	1000	S&P 600	600	HSSI	150	FTSE SmallCap	300

N_{stock}: number of stocks

potential for growth in mid-cap and small-cap stocks. Such categorization facilitates a detailed analysis of how market capitalization affects stock features and the predictive accuracy of models, thereby highlighting the differences in risk and returns across segments.

2. Feature sets. Leveraging QuantBench's capabilities in factor mining, we have integrated several widely-used feature sets at the daily level. These include the following: Alpha158 (Yang et al., 2020), which offers a range of technical indicators; Alpha101 (Kakushadze, 2016), featuring short-term volume–price (VP) characteristics; Alpha191 (Li C and Liu, 2017), tailored specifically for the Chinese stock market and focusing on short-term VP patterns.

3.1.2 Enhancing depth

Enhancing data resolution provides a detailed view of market dynamics. On a broader scale, quarterly or monthly data can uncover long-term trends and cyclical patterns, which are invaluable for strategic asset allocation and portfolio planning. On a finer scale, tick-level data capture the intricacies of intraday price movements and the effects of market microstructure, which are crucial for developing high-frequency trading strategies. The selection of modeling techniques is also influenced by data frequency, with high-frequency data often necessitating more computationally efficient and scalable methods. In QuantBench, data frequencies range from quarterly (fundamental data derived from financial statements) to tick level (detailing trades and quotes), enabling a wide array of quant tasks. Lower frequency data suit applications such as factor investing and risk modeling, while higher-frequency data are essential for analyses such as order book scrutiny and trade execution optimization. The capability to integrate data at varying frequencies also facilitates the development of innovative multi-scale strategies.

3.2 Methods

QuantBench categorizes various modeling methods from two orthogonal analytical perspectives: architectural design and training objective. The initial model suite comprises a diverse array of representative AI quant modeling methods. We will continuously expand our repository with state-of-the-art models to ensure that QuantBench remains a cutting-edge benchmarking platform. Based on whether the model treats each asset as individual or correlated, models can be categorized as temporal and spatiotemporal. Moreover, with the new pretraining paradigm becoming increasingly prevalent, we include time-series foundation models, as Fig. 4 illustrates the evolution of these types of models:

1. Temporal models. These leverage the historical data of the individual assets for prediction. Representative examples include the following: gradient boosted tree models, e.g., XGBoost (Chen TQ and Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), chosen for their robust performance on tabular data (Grinsztajn et al., 2022); recurrent neural networks (RNNs), e.g., long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), and adaptations such as state-frequency model (SFM) (Zhang LH et al., 2017), dual-stage attention-based recurrent neural network (DA-RNN) (Qin et al., 2017), and Hawkes-GRU (Sawhney et al., 2021b), which excel at capturing time-series dependencies; non-recurrent neural networks, e.g., temporal convolutional network (TCN) (Bai et al., 2018) and MLP-Mixer (Tolstikhin et al., 2021), and Transformer-based models such as Informer (Zhou HY et al., 2021), Autoformer (Wu et al., 2022), FEDformer (Zhou T et al., 2022), and PatchTST (Nie et al., 2022), offering alternative mechanisms for sequential data modeling.

2. Spatiotemporal models. QuantBench incorporates several models for spatial modeling, addressing the interconnected landscape of stocks. Simple graph models such as graph attention networks (GAT) (Veličković et al., 2018) and graph convolutional networks (GCNs) (Kipf and Welling, 2017) are used to process information across market graphs effectively. Additionally, for more complex financial networks that involve various types of relationships, heterogeneous graph models such as relational graph convolutional networks (RGCN) (Schlichtkrull et al., 2018) and relational stock ranking (RSR) network (Feng et al., 2019) are implemented. To capture higher-order relationships beyond simple pairwise interactions between stocks, QuantBench also includes hypergraph models such as ESTIMATE (Huynh et al., 2022), spatiotemporal hypergraph convolutional network (STHCN) (Sawhney et al., 2020), and spatiotemporal hypergraph

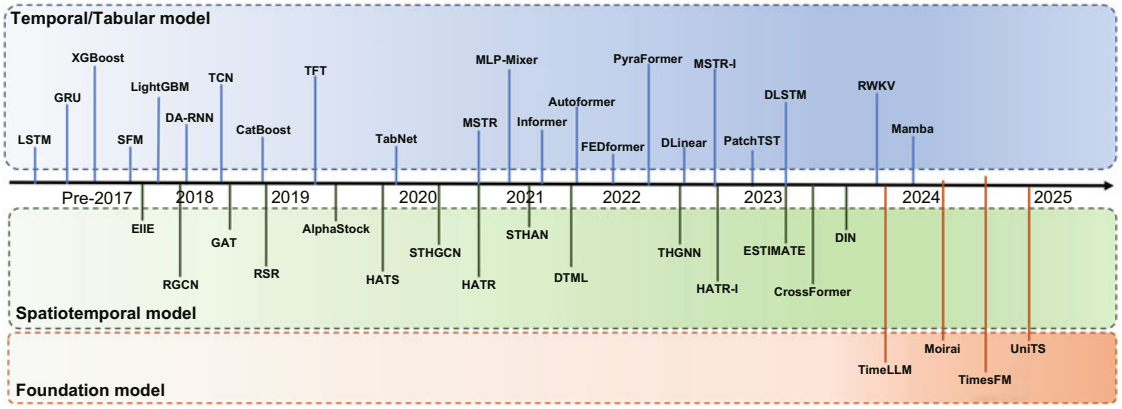

Fig. 4 A non-exhaustive illustration of models covered in QuantBench and their evolution

attention network (STHAN) (Sawhney et al., 2021a), which offer a more comprehensive analysis of the financial market's structure.

3. Model training objective. In terms of training objective, the selection of a training objective is informed by task-specific requirements and the intended outcome. For example, classification tasks predict binary outcomes, such as the direction of stock price movements, which are useful for market timing strategies. Regression tasks forecast continuous outcomes such as stock returns that are usually used for downstream portfolio optimization in stock cross-sectional strategies. Ranking focuses on the relative order of assets rather than their precise values, enhancing the profitability of characteristic-sorted portfolio. In contrast, utility maximization directly seeks to enhance financial metrics that account for both risk and return, which is usually used for end-to-end modelling that directly generates positions. Notably, some objectives, especially those involving complex simulations or non-standard feedback mechanisms, present unique challenges for optimization. For instance, objectives involving direct financial gain, such as price advantage, are not inherently differentiable due to the inclusion of trading simulation steps. These require alternative approaches such as reinforcement learning, where a model is refined based on a reward system derived from its trading performance.

Remark While we strive to faithfully reimplement models and reproduce results, discrepancies may arise between our versions and the original works. To foster transparency and community engagement, we open-source QuantBench's implementation. We encourage contributions, including corrections from original authors and new methods, to enrich and refine QuantBench.

4 Empirical results

In this section, we present our empirical findings with QuantBench, and derive some important findings and therefore potential future research directions.

4.1 Evaluation setup

To ensure comparability and reproducibility, all experiments in QuantBench are evaluated under standardized portfolio construction and cost modeling protocols.

1. Portfolio construction. Unless otherwise noted, we report results based on characteristic-sorted portfolios, a common practice in both academic and industry research. QuantBench also supports more sophisticated portfolio formation methods, including mean–variance optimization, enabling users to evaluate both heuristic and optimization-based allocation schemes within a unified framework.

2. Transaction costs. Trading costs are decomposed into two components. The first is a transaction fee, which is market-specific and directly determined by the rules of the respective exchange. The second is market impact, modeled as price slippage. Following stylized empirical findings in market microstructure, slippage is approximated using the square root law, capturing the increasing cost of executing larger trades. This two-part design ensures that backtest results reflect both structural frictions

and endogenous trading impact.

3. Execution price. Unless otherwise specified, trades are assumed to be executed at the closing price of each period, which provides a consistent and transparent reference point across datasets and markets.

4.2 Comparison between tree models and DNNs

Given the inherent noise in financial data, feature engineering plays a crucial role in improving the signal-to-noise ratio. This experiment examines the effect of feature engineering on model performance across tree-based and DNN models. Using Chinese stock market data, we employ two feature sets (Alpha101 and Alpha158) while comparing XGBoost (tree-based) and LSTM (DNN) models. The backtest uses a ranking-based stock selection strategy, selecting the top 300 stocks at each cross-section without applying feature selection. The results in Table 5 indicate that DNNs generally outperform tree-based models in terms of the information coefficient (IC). However, tree models produced better returns and Sharpe ratios (SRs) with Alpha101. When using Alpha158, DNN performance improves, though the gains are more pronounced in terms of the IC than returns. These findings suggest that tree models tend to perform better with feature sets that have strong predictive power, potentially due to reduced overfitting or differences in problem structure (Grinsztajn et al., 2022). On the other hand, DNNs excel at capturing intricate, complex patterns, making them more adept at modeling sophisticated relationships. In future research, integrating factor mining with model design could further enhance the performance of both types of models.

4.3 Comparison among different models

This experiment explores the impact of model architecture on predictive performance. Using US stock data with VP and fundamental features, we train models using IC loss and conduct backtests under the same strategy used in other experiments. Additionally, Wikidata, which provides intra-stock relational information, is incorporated. Results in Table 6 show that vanilla RNN models perform well, with slight improvements seen in adapted versions like such as ALSTM. In contrast, certain models, such as hypergraph neural networks (NNs), fail to perform adequately, probably due to mismatches between the data type and the intended use case for these models. Transformer models designed for time-series data also underperform in stock prediction tasks, a finding consistent with that of other studies. Wikidata's inclusion yields minimal improvements, possibly because the information is already widely known and exploited by other market participants. Relational graph neural networks (GNNs), such as RGCN, which account for differences in edge types, are more effective than homogeneous models such as GCN. Adaptive graph models, which learn graph structures from data, outperform others, suggesting that latent relationships between stocks hold valuable predictive power. The results point to future research opportunities in designing models that can better integrate temporal and cross-sectional information through unified architectural approaches, avoiding bottlenecks between different stages of information processing.

4.4 Different training objectives

The choice of training objective is critical in quant modeling as it directly influences the resulting model's final performance. This experiment investigates the effect of various training objectives, including classification (CLF) loss, IC loss, mean-squared error (MSE) loss, and pairwise ranking loss (Ranking), alongside a combination of MSE and ranking loss with weighted coefficients. The experimental setup mirrors that of previous sections, with the

Table 5 Comparison between XGBoost and LSTM in terms of the performance on different features

Feature	IC (%)		Diff. IC	Return (%)		Diff. return	SR		Diff. SR
	XGBoost	LSTM	(%)	XGBoost	LSTM	(%)	XGBoost	LSTM	(%)
Alpha101	2.31±0.00	4.76±0.13	−51.47	24.58±0.08	24.25±3.17	1.36	0.8093±0.0029	0.7741±0.0984	4.55
Alpha158	2.53±0.00	5.95±0.50	−57.48	20.31±0.00	23.76±5.76	−14.52	0.6407±0.0000	0.7561±0.1750	−15.26

Diff. IC: difference in IC; Diff. return: difference in return; Diff. SR: difference in SR. Red indicates negative values, meaning that the XGBoost method performs worse than the LSTM; green indicates positive values, meaning that the XGBoost method performs better than the LSTM. References to color refer to the online version of this table

results summarized in Table 7. Different training objectives excel in different performance metrics. For instance, LSTM models trained with classification loss achieve the highest Sharpe ratio, while IC loss yields the best IC and return metrics. This underscores the importance of tailoring the loss function to the model's final objective—whether it is return-focused or prediction-focused. Moreover, the effectiveness of training objectives varies between models. For cross-sectional models such as RGCN, IC loss proves superior, while temporal models such as LSTM do not benefit as much from IC loss. This suggests that future work should focus on aligning training objectives with the specific characteristics of the model architecture. Additionally, this experiment highlights the potential for research in automated

Table 6 Comparisons of performance of different models

Model type	Model	IC (%)	ICIR (%)	Return (%)	MDD (%)	SR	CR
Linear model	ARIMA	1.73	62.45	35.84	−10.95	1.1231	2.0872
Vanilla RNN	LSTM	3.89±0.29	79.56±7.72	54.90±5.53	−8.64±1.23	3.0175±0.2395	6.3922±0.5040
	GRU	4.12±0.12	81.85±13.56	61.36±4.64	−8.44±2.40	3.4433±0.3542	7.6200±1.7613
Adapted RNN	ALSTM	4.22±0.03	93.47±3.87	54.13±2.91	−8.65±1.28	3.0361±0.2256	6.3909±1.1978
	SFM	3.58±0.18	86.57±3.55	53.43±6.08	−9.06±0.60	2.9749±0.3702	5.9478±1.0961
	Multi-scale RNN	3.79±0.25	81.55±13.31	52.75±3.37	−9.02±1.71	2.9540±0.0769	5.9677±0.9095
	D-LSTM	2.86±0.25	71.36±7.11	37.23±1.57	−9.95±1.07	2.1891±0.0957	3.7554±0.2451
	Hawkes-GRU	3.46±0.62	89.47±33.15	0.35±10.81	−15.28±0.87	−0.0195±0.7246	0.0502±0.7303
Other sequence model	MSTR-I	2.52±0.30	52.30±10.97	33.53±4.27	−10.48±1.71	1.9079±0.1575	3.2627±0.6748
	TCN	3.86±0.16	92.13±10.40	54.72±4.33	−9.49±1.72	2.9729±0.1510	5.8813±0.9236
	Mixer	3.67±0.07	79.43±8.71	46.78±3.48	−8.92±2.31	2.7001±0.3210	5.5299±1.4799
	DLinear	2.99±0.22	63.39±9.54	40.09±4.53	−7.80±0.50	2.5229±0.3029	5.1675±0.7925
Transformer	Autoformer	0.08±0.13	2.54±4.47	−8.82±3.82	−15.68±5.05	−0.7467±0.3440	−0.5514±0.0728
	FEDformer	0.13±0.08	2.47±1.41	−3.16±1.37	−11.12±0.58	−0.2449±0.1082	−0.2812±0.1117
	PyraFormer	1.14±0.00	12.08±0.25	7.20±7.44	−6.55±1.75	0.6162±0.6166	0.9819±0.8742
	PatchTST	1.07±0.18	17.91±3.07	8.96±6.10	−12.40±2.07	0.6079±0.4118	0.7806±0.5557
Tabular	TFT	3.99±0.05	80.06±3.05	54.31±5.02	−7.10±0.68	3.2324±0.1920	7.6561±0.0300
Vanilla GNN	GCN	−0.10±0.04	−6.30±2.60	−13.07±1.79	−19.50±1.91	−1.1009±0.1568	−0.6685±0.0275
	GAT	3.90±0.15	85.48±3.86	58.07±4.69	−8.20±0.61	3.0730±0.1991	7.1307±0.9839
Relational GNN	RGCN	3.78±0.32	75.62±9.61	49.58±6.17	−8.83±2.17	2.7706±0.3492	5.9268±1.8845
	HATS	0.23±0.10	12.97±6.03	−11.98±3.76	−19.31±2.60	−1.0024±0.3119	−0.6110±0.1372
	RSR	0.12±0.08	5.79±3.70	−12.21±4.07	−18.62±4.30	−1.0150±0.3333	−0.6459±0.0643
	HATR-I	3.07±0.02	59.75±4.35	37.66±7.83	−11.00±5.27	2.1913±0.5738	4.0609±2.6585
Hypergraph NN	STHAN	0.05±0.13	3.11±6.73	−11.44±2.40	−17.73±2.35	−0.9541±0.2098	−0.6411±0.0525
	STHGCN	−0.01±0.04	−0.44±2.98	−9.90±0.79	−16.41±0.84	−0.8278±0.0662	−0.6031±0.0243
Adaptive GNN	THGNN	4.93±0.22	100.27±2.95	65.04±2.01	−9.32±0.37	3.3184±0.1365	6.9788±0.1852
	DTML	3.87±0.18	80.53±4.49	54.71±4.71	−7.99±0.65	3.1993±0.1832	6.9029±1.0647
	CrossFormer	4.50±0.42	77.54±13.49	55.03±6.50	−9.06±1.23	3.1269±0.2902	6.1366±0.8872

Table 7 Comparison across different training objectives

Model	Objective	Return (%)	SR	IC (%)
LSTM	CLF	19.31±0.80	1.9757±0.1888	2.77±0.32
	IC	36.03±3.99	1.8642±0.1930	3.62±0.09
	MSE	8.97±10.41	0.4484±0.5537	1.78±1.20
	Ranking	−1.11±2.74	−0.0978±0.2347	−0.07±0.09
RGCN	CLF	19.53±1.48	2.1073±0.1532	2.66 ± 0.11
	IC	44.97±3.23	2.1990±0.1855	3.97±0.30
	MSE	9.13±13.99	0.5141±0.9210	1.47±1.18
	Ranking	−3.75±6.26	−0.2929±0.5210	0.05±0.03
DTML	CLF	14.10±1.96	1.5255±0.0703	2.55±0.10
	IC	38.56±11.29	1.8796±0.6417	3.24±0.55
	MSE	−1.98±1.57	−0.1616±0.1272	0.22±0.37
	Ranking	−2.21±3.59	−0.1793±0.2833	−0.07±0.07

machine learning (AutoML) (to optimize the selection of training objectives automatically) and end-to-end learning (Liu T et al., 2023; Wei et al., 2023) to better incorporate the final goal as the training objective.

4.5 Alpha decay in quant models

In continuously evolving financial markets, alpha decay is a common phenomenon. This experiment aims to measure the speed of market evolution by evaluating different model updating schemes using walk-forward optimization techniques with varying rolling steps (3 months, 6 months, and 12 months, as well as no rolling). The study was conducted on US stock data from 2021 to 2023, with the S&P 500 used as a benchmark. Fig. 5 shows that more frequent model updates, such as the 3-month rolling scheme, consistently produce the best performance, while the no-rolling approach performs the worst. This suggests that more frequent updates help address the distribution shifts caused by market evolution, thus slowing alpha decay. However, these frequent updates come with significant computational costs, as the model must be retrained more often. This emphasizes the need for future research into more efficient online learning and continual learning techniques that can reduce the computational burden while still allowing for timely model updates.

4.6 Hyperparameter tuning for quant models

A key challenge in quantitative finance is the selection of a validation set for hyperparameter tuning. Traditionally, the validation set is selected as the tail segment of the training data just before the test set, assuming that it reflects the best out-of-sample performance. However, validation set selection is not unique (de Prado, 2018), and different methods can yield varied results. In this experiment, we compare different validation set construction methods: tail, random, and fragmented (scattered fragments across the historical period). We test two training strategies: normal, where the validation data are not used in training, and retrain, where the model is retrained using the entire dataset (including the validation set) after the hyperparameters are tuned. The results in Table 8 show that retraining on the full dataset had little or negative impact on the tail and random settings but produced positive results in the fragmented setting, suggesting retraining as a better method for hyperparameter tuning under this setting. The random validation set method outperforms the tail method, likely because it introduces more diverse data patterns into the validation process. While the fragmented setting shows potential, further research is needed to refine this approach and optimize hyperparameter stability.

4.7 Quant model ensemble

Financial data typically exhibit a low signal-to-noise ratio, making models prone to overfitting. This experiment assesses how overfitting affects model performance and explores ensembling as a mitigation strategy. We use VP data from US stocks and train an MLP-Mixer model more than 40 repeated runs, each with a different random seed. An ensemble of the 40 models' predictions is computed and backtested. Fig. 6 shows significant variance in performance across different runs, confirming the susceptibility of models to overfitting on noisy patterns. However, ensembling the predictions helps reduce this variance, improving robustness and protecting against overfitting. Even a simple model averaging approach provides a notable performance boost. Future research should focus on the methods that encourage diversity during model training, building more robust ensembles and further mitigating the effects of overfitting.

4.8 Model correlation

Fig. 7b illustrates the correlation matrix of model predictions on the CSI 300. A low correlation among models can be observed and it indicates potential for ensembling different model outputs. We illustrate the variances among different repeated runs of a single MLP-Mixer model in Fig. 6. The superior performance of the ensemble indicates that even for the same model, it may capture different views of the same dataset, which contributes to better predictions.

4.9 The effect of adding more information

Incorporating diverse information sources is increasingly prevalent in quantitative finance. This experiment evaluates whether adding more information enhances predictive accuracy. We use five information sources: VP, fundamental (F), news (N),

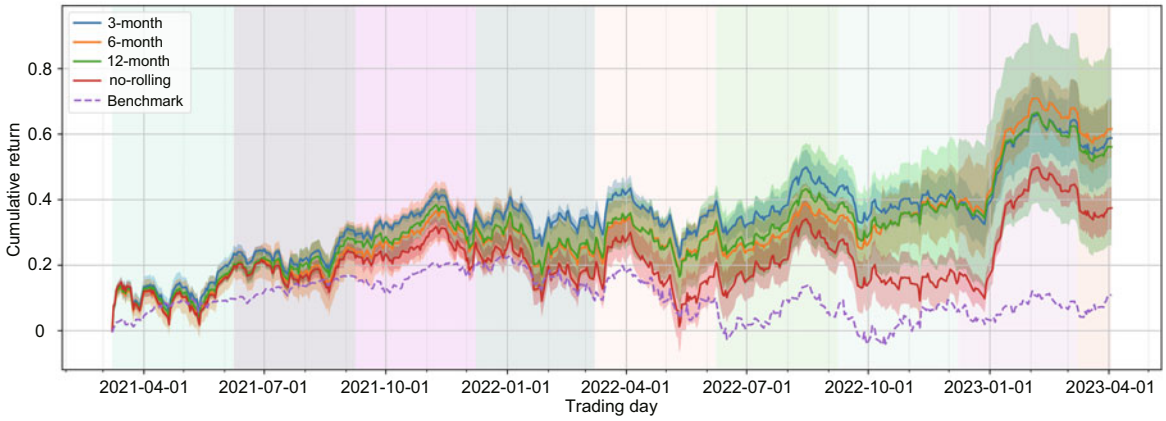

Fig. 5 Comparison of different rolling schemes

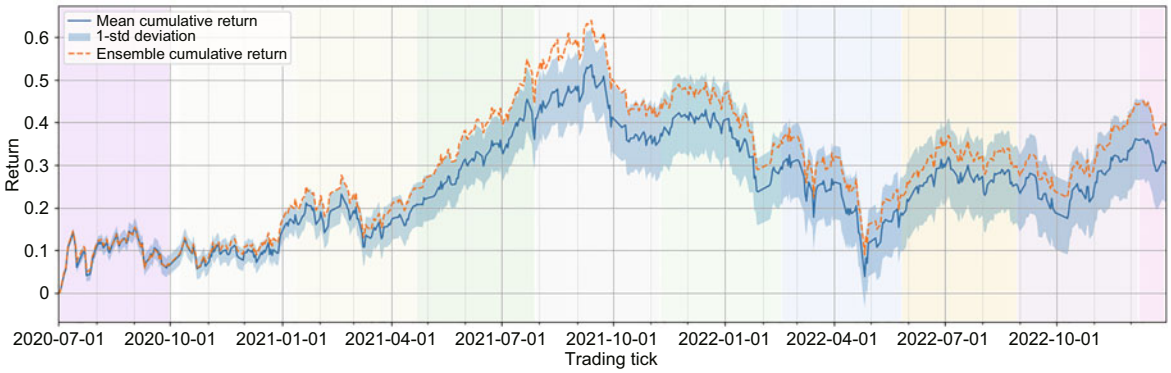

Fig. 6 Ensemble curve with variance illustrated. The shaded area indicates different rolling periods

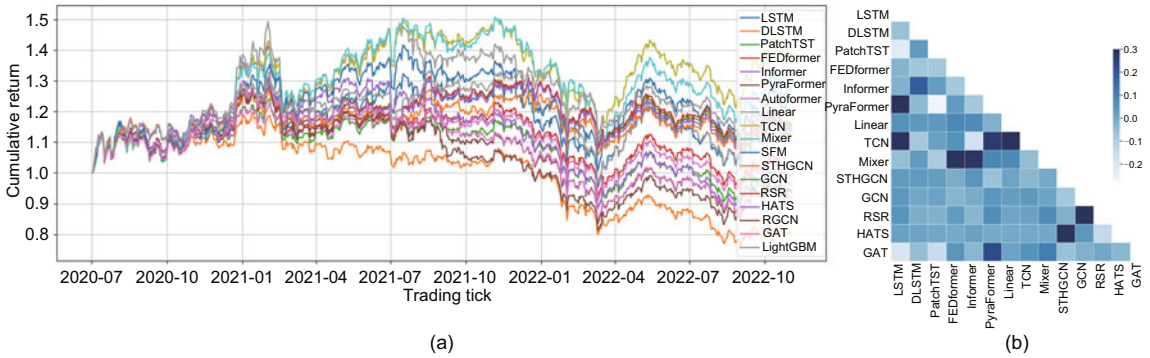

Fig. 7 Comparison of different models on the CSI 300 dataset: (a) net-value curves; (b) correlation matrix

industry (I), and Wikidata (W), applying two modeling techniques with XGBoost (tree-based) and a DNN. In XGBoost, all sources are used as features, while in the DNN, industry and Wikidata are represented as graphs. The data used are from the US stock market. As seen in Tables 9 and 10, raw VP data have weak predictive power. Adding fundamental, news, and industry data improves the performance for both models, though DNN performance suffered when industry data are structured as graphs due to the limitations of RGCN. Wikidata improve the IC performance but reduce the returns, suggesting that GNNs may be more suited to PO than direct return prediction. In summary, while more information generally improves the model performance, the method of integration (features vs. graphs) is crucial. Future work should focus on optimizing models to better leverage diverse data sources.

Table 8 Comparison of different settings of validation set selection and training schema

Training	Segmentation	IC (%)	ICIR (%)	Return (%)	SR
Normal	Tail	3.29±0.77	75.39±31.09	29.87±13.00	1.5454±0.7088
	Random	3.86±0.26	93.96±9.25	36.84±9.27	1.8969±0.4680
	Fragmented	1.71±0.56	41.58±17.52	26.55±8.12	1.2500±0.4283
Retrain	Tail	3.15±0.22	77.18±3.76	26.62±6.06	1.4431±0.2227
	Random	3.80±0.39	88.34±13.73	41.36±11.08	1.9334±0.4886
	Fragmented	2.59±0.98	57.05±20.24	30.26±9.47	1.5372±0.4348

Table 9 Performance of a DNN model on different combinations of information sources

Information source	DNN			
	IC (%)	ICIR (%)	Return (%)	SR
VP	3.43±0.09	77.05±5.73	−0.29±3.61	−0.0193±0.1961
VP-F	3.56±0.23	85.77±7.44	30.89±4.04	1.5592±0.2115
VP-F-N	4.07±0.21	90.86±7.07	31.97±10.74	1.6825±0.4451
VP-F-N-I	1.97±1.32	46.85±23.49	13.51±5.97	0.9173±0.4189
VP-F-N-W	3.80±0.18	76.58±5.27	18.06±17.81	0.9607±0.9431

Table 10 Performance of the XGBoost model on different combinations of information sources

Information source	Tree model			
	IC (%)	ICIR (%)	Return (%)	SR
VP	0.78±0.00	18.69±0.00	−5.87±0.87	−0.2939±0.0438
VP-F	1.16±0.00	28.21±0.00	−4.65±0.97	−0.2213±0.0468
VP-F-N	1.10±0.00	26.35±0.00	−6.19±2.51	−0.2901±0.1188
VP-F-N-I	1.07±0.00	28.29±0.00	1.74±0.11	0.0833±0.0048
VP-F-N-W	–	–	–	–

"–" indicates that it is impossible to obtain the data owing to the incompatibility of W type data in the tree-based model

5 Related works

5.1 AI methods for quantitative investment

Quantitative modeling traditionally starts with multi-factor models. While being explainable, these conventional models relying on linear regression and predefined factors often miss capturing complex nonlinear interactions among factors. In contrast, recent advances have integrated AI into quantitative finance (Cheng et al., 2020; Hu et al., 2021; Sonkiya et al., 2022; Kelly and Xiu, 2023), using techniques from gradient-boosted trees (Chen TQ and Guestrin, 2016; Ke et al., 2017) to deep learning, which excels in identifying intricate patterns that improve predictive precision. AI models (Qin et al., 2017; Zhang LH et al., 2017; Chen YM et al., 2018; Deng SM et al., 2019; Feng et al., 2019; Ding et al., 2021; Du et al., 2021; Sawhney et al., 2021b; Wang HY et al., 2021; Wang JN et al., 2021; Xu et al., 2021) are trained to forecast market trends and guide PO through mechanisms such as reinforcement learning (Jiang ZY et al., 2017; Wang JY et al., 2019; Wang ZC et al., 2021; Zhang YF et al., 2022) and imitation learning (Liu Y et al., 2020; Niu et al., 2022), reflecting a holistic approach to modern quant modeling. For a comprehensive overview of deep learning in quant, Jiang WW (2021) provided an extensive survey. Moreover, the specific properties of financial data call for relevant studies in causal inference (Zhu et al., 2021), transfer learning (Li WD et al., 2022), ensemble learning (Sun et al., 2023a), and continual learning (Zhao et al., 2023).

5.2 AI benchmarks for quants

Existing works have also explored evaluating these AI-driven methods. Qlib (Yang et al., 2020) is a benchmark for stock prediction, but it mostly focuses on temporal models and has an earlier cutoff (early 2022). FinRL-Meta (Liu XY et al., 2022) and TradeMaster (Sun et al., 2023c) are two comprehensive platforms for quant with a special focus on reinforcement learning. PRUDEX-Compass

(Sun et al., 2023b), on the other hand, provides a systematic evaluation framework for financial reinforcement learning (FinRL) methods across six axes: profitability, risk-control, universality, diversity, reliability, and explainability. Compared with these works, QuantBench is designed to provide a holistic view of the full quant research pipeline, rather than focusing on specific techniques such as reinforcement learning.

5.3 Emerging directions in AI for quant

1. Diffusion models. Diffusion models (He et al., 2025), are generative modeling techniques and have been widely applied in many GenAI scenarios. Recent generative approaches have also been proposed to address key challenges in quantitative modeling. DiffsFormer (Gao et al., 2024) leverages diffusion models with Transformer architectures to generate artificial stock factors, mitigating data scarcity by editing and augmenting samples in a controlled manner. Cao et al. (2025) introduced a retrieval-augmented, diffusion-based market simulator capable of generating controllable and adaptable market dynamics across multiple frequencies. Supporting stress testing and "what–if" causal prompts provides a versatile environment for evaluating quantitative strategies under diverse and volatile scenarios.

2. Foundation models. With the emergence of ChatGPT and other foundation models, large language models (LLMs) have been increasingly applied in quantitative research. Their applications in quant can be broadly divided into two categories: as agents for quant research and as analysts for financial decision-making. For the former, Alpha-GPT (Wang SZ et al., 2023) and QuantAgent (Wang SZ et al., 2024) leverage LLMs to assist researchers in formulating and refining alphas, either from scratch or based on given trading ideas. For the latter, FinAgent (Zhang WT et al., 2024) represents the first multimodal foundation agent for financial trading. It integrates numerical, textual, and visual data through a market intelligence module, enhanced by dual-level reflection and memory retrieval for adaptive learning.

6 Production readiness

Beyond research reproducibility, a practical benchmark for quant must account for considerations faced in real-world deployment. QuantBench incorporates several design choices to bridge the gap between academic evaluation and production trading environments.

1. Latency. Execution latency is a critical factor in real-time trading. QuantBench allows users to specify latency budgets, which are reflected in the simulated exchange environment. This includes order placement delays, message transmission time, and the effect of execution speed on the attained prices. By incorporating latency constraints into the simulation, QuantBench enables researchers to assess whether strategies remain effective under realistic operational conditions.

2. Risk controls. Professional trading systems enforce stringent safeguards to prevent catastrophic losses. QuantBench includes configurable modules for basic risk management, such as position limits, leverage constraints, and stop-loss rules. It also supports monitoring of drawdowns and turnover, allowing users to evaluate whether a model complies with industry-standard risk thresholds.

3. Operational costs. In addition to explicit transaction fees and market impact, QuantBench highlights broader operational costs that influence strategy viability. These include computational overhead associated with training and updating models, as well as the costs of data acquisition and storage. While these are not modeled as direct deductions in backtests, they are documented and can be parameterized by users to approximate production expenses.

4. Alignment with practice. By incorporating latency, risk controls, and operational costs, QuantBench moves closer to a production-ready evaluation environment. Although simplified compared to live systems, these extensions ensure that strategies validated within QuantBench are assessed not only on predictive accuracy but also on their robustness and sustainability under realistic deployment constraints.

7 Discussion

Despite the significant strides made with QuantBench, opportunities for enhancement remain. In terms of data, there is a need to broaden the scope by incorporating alternative data sources and to deepen the granularity by including metrics such as order flow, which would enrich the dataset with

more detailed information. On the model front, integrating newer architectures and innovative formulations will further enhance the platform's capability. For evaluation, implementing more realistic and efficient backtesting methods, along with metrics tailored to specific tasks, will improve the robustness and applicability of the benchmarks.

Contributors

Saizhuo WANG and Jian GUO conceived and designed the research. Saizhuo WANG, Hao KONG, Jiadong GUO, Fengrui HUA, Wanyun ZHOU, Jiahao ZHENG, and Xinyu WANG conducted the experiments and drafted the paper. Yiyan QI and Jian GUO helped organize the paper. Lionel M. NI and Jian GUO provided advisory support. All the authors participated in the revision of the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data and code that support the findings of this study are available via https://github.com/SaizhuoWang/quantbench.

References

Bai SJ, Kolter JZ, Koltun V, 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. https://arxiv.org/abs/1803.01271

Cao BK, Lin XY, Qi YY, et al., 2025. Financial wind tunnel: a retrieval-augmented market simulator. https://doi.org/10.48550/arXiv.2503.17909

Cattaneo MD, Crump RK, Farrell MH, et al., 2020. Characteristic-sorted portfolios: estimation and inference. *Rev Econ Stat*, 102(3):531-551. https://doi.org/10.1162/rest_a_00883

Chen TQ, Guestrin C, 2016. XGBoost: a scalable tree boosting system. Proc 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.785-794. https://doi.org/10.1145/2939672.2939785

Chen YM, Wei ZY, Huang XJ, 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. Proc 27th ACM Int Conf on Information and Knowledge Management, p.1655-1658. https://doi.org/10.1145/3269206.3269269

Cheng DW, Yang FZ, Wang XY, et al., 2020. Knowledge graph-based event embedding framework for financial quantitative investments. Proc 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.2221-2230. https://doi.org/10.1145/3397271.3401427

Cui C, Wang W, Zhang MH, et al., 2021. AlphaEvolve: a learning framework to discover novel alphas in quantitative investment. Proc Int Conf on Management of Data, p.2208-2216. https://doi.org/10.1145/3448016.3457324

Deng J, Dong W, Socher R, et al., 2009. ImageNet: a large-scale hierarchical image database. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.248-255. https://doi.org/10.1109/CVPR.2009.5206848

Deng SM, Zhang NY, Zhang W, et al., 2019. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. World Wide Web Conf, p.678-685. https://doi.org/10.1145/3308560.3317701

de Prado ML, 2018. Advances in Financial Machine Learning. John Wiley & Sons, Hoboken, USA.

Ding Q, Wu S, Sun H, et al., 2021. Hierarchical multi-scale Gaussian Transformer for stock movement prediction. Proc 29th Int Joint Conf on Artificial Intelligence, p.4640-4646. https://doi.org/10.24963/ijcai.2020/640

Du YT, Wang JD, Feng WJ, et al., 2021. AdaRNN: adaptive learning and forecasting of time series. Proc 30th ACM Int Conf on Information & Knowledge Management, p.402-411. https://doi.org/10.1145/3459637.3482315

Feng FL, He XN, Wang X, et al., 2019. Temporal relational ranking for stock prediction. *ACM Trans Inform Syst*, 37(2):27. https://doi.org/10.1145/3309547

Gao Y, Chen HK, Wang X, et al., 2024. DiffsFormer: a diffusion Transformer on stock factor augmentation. https://doi.org/10.48550/arXiv.2402.06656

Grinsztajn L, Oyallon E, Varoquaux G, 2022. Why do tree-based models still outperform deep learning on typical tabular data? Proc 36th Int Conf on Neural Information Processing Systems, Article 37. https://dl.acm.org/doi/10.5555/3600270.3600307

He CM, Shen YQ, Fang CY, et al., 2025. Diffusion models in low-level vision: a survey. *IEEE Trans Pattern Anal Mach Intell*, 47(6):4630-4651. https://doi.org/10.1109/TPAMI.2025.3545047

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neur Comput*, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu ZX, Zhao YQ, Khushi M, 2021. A survey of forex and stock price prediction using deep learning. *Appl Syst Innov*, 4(1):9. https://doi.org/10.3390/asi4010009

Huynh TT, Nguyen MH, Nguyen TT, et al., 2022. Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction. Proc 16th ACM Int Conf on Web Search and Data Mining, p.850-858. https://doi.org/10.1145/3539597.3570427

Jiang WW, 2021. Applications of deep learning in stock market prediction: recent progress. *Expert Syst Appl*, 184:115537. https://doi.org/10.1016/j.eswa.2021.115537

Jiang ZY, Xu DX, Liang JJ, 2017. A deep reinforcement learning framework for the financial portfolio management problem. https://doi.org/10.48550/arXiv.1706.10059

Kakushadze Z, 2016. 101 formulaic alphas. http://arxiv.org/abs/1601.00991

Ke GL, Meng Q, Finley T, et al., 2017. LightGBM: a highly efficient gradient boosting decision tree. Proc 31st Int Conf on Neural Information Processing Systems, p.3149-3157.

Kelly B, Xiu D, 2023. Financial machine learning. *Found Trends Finance*, 13(3-4):205-363. https://doi.org/10.1561/0500000064

Kipf TN, Welling M, 2017. Semi-supervised classification with graph convolutional networks. 5th Int Conf on Learning Representations.

Koa KJL, Ma YS, Ng R, et al., 2024. Learning to generate explainable stock predictions using self-reflective large language models. Proc ACM Web Conf, p.4304-4315. https://doi.org/10.1145/3589334.3645611

Li C, Liu FB, 2017. Multi-Factor Stock Selection via Short-Term Volume–Price. Guotai Junan Securities, Hong Kong, China.

Li WD, Yang X, Liu WQ, et al., 2022. DDG-DA: data distribution generation for predictable concept drift adaptation. Proc 36th AAAI Conf on Artificial Intelligence, p.4092-4100. https://doi.org/10.1609/aaai.v36i4.20327

Liu T, Roberts S, Zohren S, 2023. Deep inception networks: a general end-to-end framework for multi-asset quantitative strategies. https://doi.org/10.48550/arXiv.2307.05522

Liu XY, Xia ZY, Rui JY, et al., 2022. FinRL-Meta: market environments and benchmarks for data-driven financial reinforcement learning. Proc 36th Int Conf on Neural Information Processing Systems, Article 134. https://dl.acm.org/doi/abs/10.5555/3600270.3600404

Liu Y, Liu Q, Zhao HK, et al., 2020. Adaptive quantitative trading: an imitative deep reinforcement learning approach. Proc 34th AAAI Conf on Artificial Intelligence, p.2128-2135. https://doi.org/10.1609/aaai.v34i02.5587

Markowitz H, 1952. Portfolio selection. *J Finance*, 7(1):77-91. https://doi.org/10.2307/2975974

Markowitz H, 1959. Portfolio Selection: Efficient Diversification of Investments. Yale University Press, New Haven, USA.

Nie YQ, Nguyen NH, Sinthong P, et al., 2022. A time series is worth 64 words: long-term forecasting with transformers. 11th Int Conf on Learning Representations.

Niu H, Li SY, Li J, 2022. MetaTrader: an reinforcement learning approach integrating diverse policies for portfolio optimization. Proc 31st ACM Int Conf on Information & Knowledge Management, p.1573-1583. https://doi.org/10.1145/3511808.3557363

Prokhorenkova L, Gusev G, Vorobev A, et al., 2018. CatBoost: unbiased boosting with categorical features. Proc 32nd Int Conf on Neural Information Processing Systems, p.6639-6649.

Qin Y, Song DJ, Chen HF, et al., 2017. A dual-stage attention-based recurrent neural network for time series prediction. Proc 26th Int Joint Conf on Artificial Intelligence, p.2627-2633. https://doi.org/10.24963/ijcai.2017/366

Sawhney R, Agarwal S, Wadhwa A, et al., 2020. Spatiotemporal hypergraph convolution network for stock movement forecasting. Proc IEEE Int Conf on Data Mining, p.482-491. https://doi.org/10.1109/ICDM50108.2020.00057

Sawhney R, Agarwal S, Wadhwa A, et al., 2021a. Exploring the scale-free nature of stock markets: hyperbolic graph learning for algorithmic trading. Proc Web Conf, p.11-22. https://doi.org/10.1145/3442381.3450095

Sawhney R, Agarwal S, Wadhwa A, et al., 2021b. Stock selection via spatiotemporal hypergraph attention network: a learning to rank approach. Proc 35th AAAI Conf on Artificial Intelligence, p.497-504. https://doi.org/10.1609/aaai.v35i1.16127

Schlichtkrull M, Kipf TN, Bloem P, et al., 2018. Modeling relational data with graph convolutional networks. 15th Int Conf on the Semantic Web, p.593-607. https://doi.org/10.1007/978-3-319-93417-4_38

Sonkiya P, Bajpai V, Bansal A, 2022. Stock price prediction using artificial intelligence: a survey. IEEE 7th Int Conf for Convergence in Technology, p.1-9. https://doi.org/10.1109/I2CT54291.2022.9825415

Sun S, Wang XR, Xue WQ, et al., 2023a. Mastering stock markets with efficient mixture of diversified trading experts. Proc 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.2109-2119. https://doi.org/10.1145/3580305.3599424

Sun S, Qin ML, Wang XR, et al., 2023b. PRUDEX-Compass: towards systematic evaluation of reinforcement learning in financial markets. https://arxiv.org/abs/2302.00586

Sun S, Qin ML, Zhang WT, et al., 2023c. TradeMaster: a holistic quantitative trading platform empowered by reinforcement learning. Proc 37th Int Conf on Neural Information Processing Systems, Article 2576.

Tolstikhin I, Houlsby N, Kolesnikov A, et al., 2021. MLP-Mixer: an all-MLP architecture for vision. Proc 35th Int Conf on Neural Information Processing Systems, Article 1857.

Veličković P, Cucurull G, Casanova A, et al., 2018. Graph attention networks. http://arxiv.org/abs/1710.10903

Wang A, Singh A, Michael J, et al., 2019. GLUE: a multi-task benchmark and analysis platform for natural language understanding. Proc 7th Int Conf on Learning Representations.

Wang HY, Li S, Wang TJ, et al., 2021. Hierarchical adaptive temporal-relational modeling for stock trend prediction. Proc 30th Int Joint Conf on Artificial Intelligence, p.3691-3698. https://doi.org/10.24963/ijcai.2021/508

Wang JN, Zhang S, Xiao YH, et al., 2021. A review on graph neural network methods in financial applications. https://doi.org/10.48550/arXiv.2111.15367

Wang JY, Zhang Y, Tang K, et al., 2019. AlphaStock: a buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. Proc 25th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.1900-1908. https://doi.org/10.1145/3292500.3330647

Wang SZ, Yuan H, Zhou L, et al., 2023. Alpha-GPT: human-AI interactive alpha mining for quantitative investment. https://doi.org/10.48550/arXiv.2308.00016

Wang SZ, Yuan H, Ni LM, et al., 2024. QuantAgent: seeking holy grail in trading by self-improving large language model. https://doi.org/10.48550/arXiv.2402.03755

Wang ZC, Huang BW, Tu SK, et al., 2021. DeepTrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. Proc 35th AAAI Conf on Artificial Intelligence, p.643-650. https://doi.org/10.1609/aaai.v35i1.16144

Wei ZK, Dai B, Lin DH, 2023. E2EAI: end-to-end deep learning framework for active investing. Proc 4th ACM Int Conf on AI in Finance, p.55-63. https://doi.org/10.1145/3604237.3626848

Wu HX, Xu JH, Wang JM, et al., 2022. Autoformer: decomposition Transformers with auto-correlation for long-term series forecasting. Proc 35th Int Conf on Neural Information Processing Systems, Article 1717.

Xu WT, Liu WQ, Wang LW, et al., 2021. HIST: a graph-based framework for stock trend forecasting via mining concept-oriented shared information. https://doi.org/10.48550/arXiv.2110.13716

Yang X, Liu W, Zhou D, et al., 2020. Qlib: an AI-oriented quantitative investment platform. http://arxiv.org/abs/2009.11189

Yu S, Xue HY, Ao X, et al., 2023. Generating synergistic formulaic alpha collections via reinforcement learning. Proc 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.5476-5486. https://doi.org/10.1145/3580305.3599831

Zhang LH, Aggarwal C, Qi GJ, 2017. Stock price prediction via discovering multi-frequency trading patterns. Proc 23rd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.2141-2149. https://doi.org/10.1145/3097983.3098117

Zhang TP, Li YQ, Jin YF, et al., 2020. AutoAlpha: an efficient hierarchical evolutionary algorithm for mining alpha factors in quantitative investment. https://doi.org/10.48550/arXiv.2002.08245

Zhang WT, Zhao LX, Xia HC, et al., 2024. A multimodal foundation agent for financial trading: tool-augmented, diversified, and generalist. Proc 30th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.4314-4325. https://doi.org/10.1145/3637528.3671801

Zhang YF, Zhao PL, Wu Q, et al., 2022. Cost-sensitive portfolio selection via deep reinforcement learning. *IEEE Trans Knowl Data Eng*, 34(1):236-248. https://doi.org/10.1109/TKDE.2020.2979700

Zhao LF, Kong SM, Shen YY, 2023. DoubleAdapt: a meta-learning approach to incremental learning for stock trend forecasting. Proc 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.3492-3503. https://doi.org/10.1145/3580305.3599315

Zhou HY, Zhang SH, Peng JQ, et al., 2021. Informer: beyond efficient Transformer for long sequence time-series forecasting. Proc 35th AAAI Conf on Artificial Intelligence, p.11106-11115. https://doi.org/10.1609/aaai.v35i12.17325

Zhou T, Ma ZQ, Wen QS, et al., 2022. FEDformer: frequency enhanced decomposed Transformer for long-term series forecasting. https://doi.org/10.48550/arXiv.2201.12740

Zhu YD, Chen WY, Zhang Y, et al., 2021. Probabilistic framework for modeling event shocks to financial time series. Proc 2nd ACM Int Conf on AI in Finance, Article 42. https://doi.org/10.1145/3490354.3494407