

QuantEval: A Benchmark for Financial Quantitative Tasks in Large Language Models

Anonymous Author(s)

	59
 Quantitative Analysis Task	60
Q: The risk-free interest rate is 6%, the dividend yield is 4%, and the current spot price of the stock is 400. A 4-month futures contract is priced at 405. Is there an arbitrage opportunity?	61
Theoretical futures price: $F_0 = S_0 e^{(r-q)T} = 400 \times e^{(0.06-0.04) \times \frac{4}{12}}$ $\approx 400 \times e^{0.00666667} \approx 402.68. F_{\text{mkt}} (= 405) > F_0 (= 402.68)$ Buy the spot and sell the futures to arbitrage	62
Answer: BUY ✓	63
 Theoretical Understanding Task	64
Q: A pension fund has a present value of 1.74 billion. When the discount rate decreases by 0.5% the liability increases by 0.08 billion. This liability position is equivalent to which of the following?	65
A. A short position in equities ✗ B. A short position in cash ✗ C. A short position in a 9-year bond ✗ D. A short position in a bond with duration of about 9 years ✓	66
 Computational Implementation Task	67
Task: Develop a reinforcement learning trading strategy on the VeighNa platform using the CTA template as an experienced quant researcher.	68
Description : • Apply RL to learn <i>trading decisions</i> in simulation. • States include <i>price, indicators, and position</i> ; rewards balance <i>profit and risk</i> . • Support long/short futures trading; train on large data and avoid overfitting.	69
Code Requirements: ◆ Provide <i>detailed explanations</i> of the program logic in Chinese. ◆ Use <i>type hinting</i> in the code. ◆ Ensure <i>full PEP 8</i> compliance.	70
	71

Figure 1: Examples from the three core tasks in QuantEval: knowledge-based question answering, quantitative mathematical reasoning, and quantitative strategy coding.

Abstract

Large Language Models (LLMs) have shown strong abilities across many fields, but their evaluation in financial quantitative tasks remains limited and scattered. Current financial benchmarks mainly target knowledge-based question answering (QA), often overlooking the core tasks of quantitative reasoning and strategy development that are fundamental to quantitative finance. To bridge this gap, we propose **QuantEval**, a comprehensive benchmark designed to evaluate the financial quantitative competency of LLM in three key dimensions: knowledge-based QA, quantitative mathematical reasoning, and quantitative strategy coding. QuantEval’s samples are rigorously curated from multiple sources, including real-world financial market data, publicly available data sources, and synthetically constructed complex quantitative scenarios, ensuring strong alignment with practical challenges in quantitative finance. We conduct extensive evaluations of more than a dozen state-of-the-art LLMs on QuantEval, revealing significant performance challenges,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW, Dubai, UAE

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-XXXX-X/2026/04
<https://doi.org/XXXXXXX.XXXXXXX>

indicating substantial room for improvement. Motivated by these findings, we further conduct large-scale supervised fine-tuning (SFT) and reinforcement learning (RL) experiments using proprietary non-benchmark and open-source data to explore effective approaches for enhancing model performance. We expect QuantEval to serve as a valuable resource for advancing research on LLMs’ quantitative capabilities, facilitating their practical application in financial markets.

CCS Concepts

- Applied computing → Secure online transactions;
- Computing methodologies → Natural language processing.

Keywords

Quantitative Finance, Large Language Models, Benchmark

ACM Reference Format:

Anonymous Author(s). 2026. QuantEval: A Benchmark for Financial Quantitative Tasks in Large Language Models. In *Proceedings of The Web Conference 2026 (WWW)*. ACM, New York, NY, USA, 12 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Large Language Models (LLMs) have recently revolutionized the field of natural language processing, demonstrating remarkable capabilities in understanding, reasoning, and generating human-like

text across diverse domains [12, 15, 20]. Their ability to perform complex tasks such as multi-step reasoning, code generation, and domain-specific question answering has sparked widespread interest in leveraging these models for specialized applications, including finance. The financial industry, characterized by its reliance on quantitative analysis, mathematical modeling, and algorithmic strategy development, presents a particularly promising yet challenging domain for LLM deployment.

Despite the growing enthusiasm, the evaluation of LLMs in financial quantitative tasks remains fragmented and insufficient. Existing financial benchmarks predominantly focus on knowledge-based question answering (QA), emphasizing factual recall, terminology understanding, or simple numerical reasoning [4, 5, 16, 17, 23]. While these tasks are important for assessing domain familiarity, they fall short of capturing the core competencies required in quantitative finance. Specifically, quantitative finance demands rigorous mathematical reasoning over complex financial data, as well as the ability to design, implement, and validate executable trading strategies that can operate effectively in dynamic market environments.

The gap between current benchmarks and real-world financial quantitative challenges limits the practical applicability of LLMs in this domain. Quantitative finance is inherently multifaceted, involving not only conceptual knowledge but also precise numerical computation, logical inference, and algorithmic coding skills. For example, tasks such as option pricing, risk assessment, portfolio optimization, and algorithmic trading strategy development require models to integrate domain expertise with quantitative rigor and programming proficiency. Existing datasets rarely encompass this breadth, often neglecting the critical dimension of quantitative strategy coding and the evaluation of generated code in realistic market scenarios.

To address these limitations, we propose **QuantEval**, a comprehensive benchmark explicitly designed to evaluate the financial quantitative capabilities of LLMs across three fundamental dimensions: (1) *Knowledge-based QA*, assessing understanding of financial concepts and terminology; (2) *Quantitative Mathematical Reasoning*, challenging models to solve mathematically rigorous problems grounded in real financial data and theory; and (3) *Quantitative Strategy Coding*, evaluating the ability to generate executable trading strategies whose performance can be validated through backtesting on historical market data. The benchmark consists of approximately 1.6k carefully curated samples, ensuring a thorough evaluation across these dimensions.

QuantEval distinguishes itself through a carefully curated and diverse dataset that integrates authentic market data, expert-validated financial knowledge, and synthetically generated complex quantitative scenarios. A key innovation is the incorporation of a Commodity Trading Advisor (CTA) backtesting framework, which enables rigorous evaluation of generated trading strategies not only for syntactic correctness but also for financial performance metrics such as returns, risk, and Sharpe ratios. This grounding in practical financial workflows ensures that QuantEval reflects the true challenges faced by quantitative analysts and algorithmic traders.

We conduct extensive experiments evaluating over a dozen state-of-the-art LLMs on QuantEval, including both open-source and proprietary models. Our results reveal significant performance gaps, particularly in quantitative reasoning and strategy coding tasks,

underscoring the difficulty of these challenges and the need for further research. We also investigate the effect of Chain-of-Thought (CoT) prompting by evaluating model responses both with and without CoT. Motivated by these findings, we explore large-scale supervised fine-tuning and reinforcement learning techniques to enhance model capabilities, showing promising directions for future development.

In summary, our contributions are as follows:

- We introduce **QuantEval**, the first comprehensive benchmark that systematically evaluates LLMs on financial quantitative tasks spanning knowledge QA, mathematical reasoning, and strategy coding, thereby filling a critical gap in existing evaluation frameworks.
- We design a rigorous data collection and generation pipeline that combines real-world financial market data, expert-curated content, and synthetic quantitative strategy code based on an industry-recognized CTA backtesting framework, ensuring high-quality, diverse, and practically relevant samples.
- We perform thorough evaluations of more than a dozen leading LLMs, revealing substantial challenges and providing a clear roadmap for advancing LLMs' quantitative finance capabilities.
- We conduct preliminary large-scale supervised fine-tuning and reinforcement learning experiments, demonstrating effective approaches to improve LLM performance on complex financial quantitative tasks.

2 QuantEval

2.1 Overview

QuantEval is a rigorously designed benchmark aimed at comprehensively evaluating Large Language Models (LLMs) on financial quantitative tasks. Unlike existing financial benchmarks that predominantly emphasize knowledge-based question answering, QuantEval explicitly targets three core dimensions that collectively define the competencies required in quantitative finance: knowledge-based question answering, quantitative mathematical reasoning, and quantitative strategy coding. Each dimension presents unique challenges and requires specialized data construction methodologies to ensure the benchmark's practical relevance, rigor, and diversity.

QuantEval is constructed through a carefully architected data pipeline that integrates authentic market data, expert domain knowledge, and synthetically generated complex quantitative scenarios. This pipeline is organized into three principal stages, Data Preparation, Data Construction, and Validation, each adapted to the distinct characteristics of the three task categories while maintaining a unified framework for quality assurance and scalability. The three stages are summarized in Figure 2.

2.2 Data Preparation

The initial stage focuses on collecting and organizing foundational materials that serve as the bedrock for subsequent question and code generation. Given the heterogeneity of the three task categories, tailored data sourcing strategies are employed to capture the breadth and depth of quantitative finance.

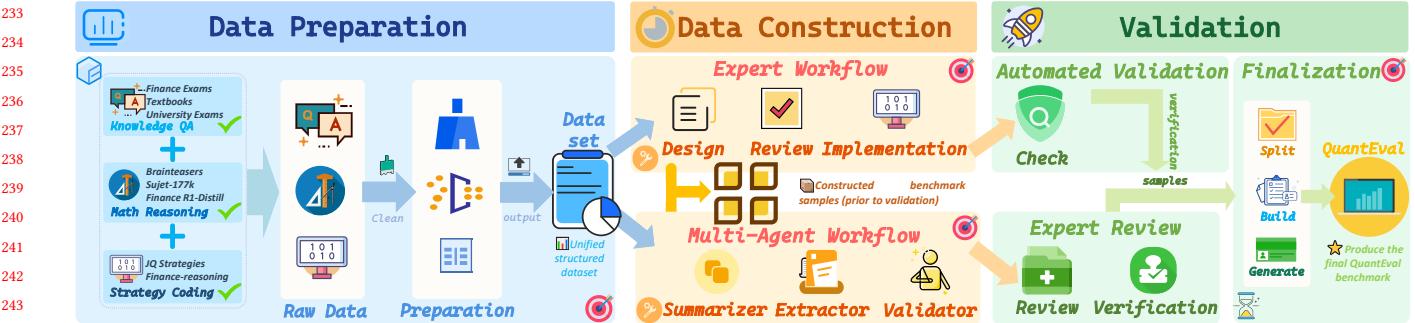


Figure 2: The QuantEval data construction pipeline consists of three main stages: Data Preparation, Data Construction, and Validation. Data Preparation involves sourcing and organizing financial texts, market data, and algorithmic trading code. Data Construction combines expert annotation and a multi-agent automated framework to generate high-quality knowledge questions, quantitative reasoning problems, and executable strategy coding tasks. Validation employs both automated checks and expert review. The integration of a Commodity Trading Advisor (CTA) backtesting framework enables realistic evaluation of generated trading strategies.

For the knowledge-based question answering dimension, authoritative textual materials are sourced from financial textbooks, peer-reviewed academic papers, regulatory filings, and curated glossaries of financial terminology. Domain experts specializing in quantitative finance meticulously review these materials, selecting passages that cover essential concepts, theoretical frameworks, and domain-specific vocabulary. To ensure comprehensive coverage and mitigate potential biases, the materials span multiple subdomains such as asset pricing, risk management, derivatives, and portfolio theory. The collected texts are standardized and segmented into coherent knowledge units, facilitating precise and contextually grounded question formulation.

The quantitative mathematical reasoning dimension requires datasets rooted in realistic financial computations and problem-solving scenarios. To this end, extensive historical market data, including stock prices, interest rates, volatility indices, and macroeconomic indicators, are gathered from reputable public sources such as Bloomberg, Yahoo Finance, and Quandl. Complementing these data, problem templates are extracted from quantitative finance coursework, professional certification materials (e.g., CFA, FRM), and industry whitepapers. These templates are enriched with real market data to create contextually rich problems that require calculations involving option pricing models, risk metrics, portfolio optimization, and other quantitative finance techniques.

For the quantitative strategy coding dimension, algorithmic trading strategies and backtesting scripts are collected from a variety of sources, including open-source repositories, academic publications, and proprietary Commodity Trading Advisor (CTA) frameworks. These codebases encompass a wide spectrum of quantitative approaches, such as momentum strategies, mean reversion, statistical arbitrage, and factor-based models. Financial experts curate and annotate these code snippets to ensure clarity, correctness, and alignment with realistic trading scenarios. Importantly, the CTA strategies incorporated have been designed and validated by domain experts with extensive professional experience and have been

actively employed in offline industry settings. The CTA methodology is well-recognized and widely adopted within the quantitative finance community, providing a robust and industry-validated foundation for the benchmark's coding tasks. Alongside the code, metadata including strategy parameters, historical performance metrics, and market conditions are collected to support comprehensive evaluation and benchmarking.

To further enhance the dataset's complexity and practical relevance, synthetic quantitative scenarios are designed by combining real market data with hypothetical events or constraints. These scenarios simulate market shocks, regulatory changes, portfolio rebalancing under risk limits, and other realistic financial decision-making contexts. The synthetic data are generated through controlled scripts informed by expert input, enabling the creation of challenging reasoning and coding tasks that mirror the intricacies of real-world quantitative finance.

2.3 Data Construction

Building upon the prepared materials, the second stage focuses on generating the actual benchmark samples—comprising knowledge questions, mathematical problems, and coding tasks—through a hybrid approach that combines expert annotation with automated multi-agent generation. This dual approach balances the need for high-quality, domain-accurate content with scalability and diversity.

Financial domain experts play a pivotal role in crafting high-quality samples, particularly for knowledge QA and complex reasoning tasks. Their responsibilities include revising and refining existing questions to enhance clarity, relevance, and difficulty, ensuring that distractors (incorrect options) are plausible yet unambiguously incorrect, and that questions demand genuine understanding rather than superficial recall. Experts also design novel questions and problems based on curated texts, market data, and synthetic scenarios. For mathematical reasoning, this involves formulating multi-step problems that require quantitative calculations and logical inference grounded in financial theory. For coding tasks,

349 experts specify challenges such as strategy implementation, partial
 350 code completion, debugging, and parameter tuning, ensuring that
 351 the tasks reflect realistic quantitative trading workflows. Additionally,
 352 experts review and annotate quantitative strategy code samples
 353 to verify correctness, readability, and alignment with benchmark
 354 objectives, providing detailed explanations and expected outputs
 355 to facilitate evaluation.

356 Inspired by expert workflows, a multi-agent automated construction
 357 framework is developed to efficiently scale data generation, particularly
 358 for mathematical reasoning and coding tasks. This framework
 359 comprises specialized roles that collaborate to produce high-quality
 360 samples. A summarizer agent analyzes financial texts and market data to
 361 generate concise summaries highlighting key knowledge points and quantitative
 362 relationships. A problem extractor agent identifies and extracts relevant data segments and problem
 363 templates suitable for question or code generation. The question and code generator agent leverages advanced LLMs, employing
 364 Chain-of-Thought prompting to produce mathematically rigorous
 365 questions and executable code snippets. This agent operates under
 366 detailed prompt templates and is guided by examples derived
 367 from expert annotations to ensure adherence to domain standards.
 368 Finally, a validation agent performs preliminary checks on generated
 369 samples, including syntactic correctness, logical consistency, and alignment with task requirements. This multi-agent system
 370 enables efficient, large-scale generation of diverse and high-quality
 371 benchmark samples while maintaining domain relevance and rigor.
 372
 373
 374

2.4 Validation

375 Ensuring the reliability, rigor, and practical relevance of QuantEval
 376 samples is of paramount importance. To this end, a multi-tiered
 377 validation process involving both automated agents and domain
 378 experts is implemented.

379 The automated validation agent conducts initial screening to
 380 detect duplicate or near-duplicate questions, inconsistencies, and
 381 trivial or overly simplistic items. For knowledge QA and reasoning
 382 tasks, it verifies that questions cannot be answered correctly
 383 through superficial keyword matching or simple heuristics, thereby
 384 ensuring that genuine understanding and reasoning are required.
 385 For coding tasks, the agent checks code executability, correctness of
 386 outputs, and adherence to specified constraints, including successful
 387 execution within the integrated CTA backtesting framework.
 388 Samples failing these criteria are either revised or discarded to
 389 maintain dataset quality.

390 Subsequently, domain experts perform thorough reviews of all
 391 benchmark samples. Their evaluation focuses on confirming factual
 392 accuracy, mathematical soundness, and code validity. Experts
 393 ensure that questions and tasks are unambiguous, well-phrased, and aligned with the intended difficulty levels. They also conduct
 394 ethical and sensitivity checks to verify that samples do not contain
 395 proprietary, sensitive, or ethically problematic content. When
 396 necessary, experts request revisions or reject samples that do not
 397 meet stringent quality standards. This iterative feedback loop
 398 between validation and construction stages continues until the dataset
 399 achieves the desired quality threshold, ensuring that QuantEval is
 400 both challenging and trustworthy.

2.5 Implementation Details

401 The data construction pipeline extensively leverages state-of-the-art
 402 LLMs, primarily GPT-4o and Claude-4-sonnet, for generation
 403 and validation tasks. Chain-of-Thought prompting is employed
 404 throughout to enhance reasoning transparency and improve output
 405 quality. The multi-agent system is orchestrated via modular
 406 APIs, allowing seamless integration of expert inputs and automated
 407 components.

408 A key feature of the quantitative strategy coding dimension is
 409 the integration of a CTA backtesting framework. This framework
 410 enables the generation of realistic trading scenarios and the validation
 411 of code correctness through simulated executions on historical
 412 market data. The backtesting environment supports evaluation of
 413 strategy performance metrics, risk characteristics, and robustness
 414 under varying market conditions. By grounding the benchmark
 415 in this industry-recognized methodology, QuantEval ensures that
 416 the coding tasks reflect authentic quantitative finance workflows
 417 and provide meaningful challenges for LLMs aspiring to practical
 418 financial applications.

2.6 Data Statistics

419 Our dataset comprises a total of 1,575 samples, systematically organized
 420 into three primary categories: *Knowledge-based QA* (660 samples),
 421 *Quantitative Mathematical Reasoning* (855 samples), and
 422 *Quantitative Strategy Coding* (60 samples).

423 For the *Knowledge-based QA* and *Quantitative Reasoning* tasks,
 424 the dataset includes both multiple-choice questions and fill-in-the-blank
 425 formats, designed to evaluate a range of competencies from conceptual
 426 understanding to complex problem-solving. Each question is carefully
 427 crafted to challenge models on different aspects of financial theory,
 428 quantitative methods, and data interpretation.

429 In the *Quantitative Strategy Coding* category, each sample provides
 430 a detailed problem description alongside a widely recognized,
 431 expert-validated trading strategy implemented in code. This code
 432 represents the most reliable and credible approach to achieving
 433 the stated financial returns. Additionally, the dataset includes the
 434 corresponding performance metrics of these strategies, enabling
 435 rigorous evaluation of generated code against trusted benchmarks.

436 All samples are presented exclusively in English to maintain
 437 consistency and facilitate broad accessibility. This structured and
 438 diverse dataset supports comprehensive assessment of large language
 439 models across multiple facets of financial quantitative tasks.

3 Evaluation on QuantEval

3.1 Evaluation Setup

440 We conduct a comprehensive evaluation of over a dozen state-of-the-art Large Language Models (LLMs) on the proposed **QuantEval**
 441 benchmark, covering its three core task dimensions: *Knowledge-based QA*, *Quantitative Mathematical Reasoning*, and *Quantitative Strategy Coding*. The evaluated models span both **open-source**
 442 and **proprietary** families, including Qwen3 variants (4B, 8B, 14B, 30B) [18], DeepSeek distilled models [6], DianJin-R1-7B [22], Claude-4-sonnet [3], Gemini, and GPT-5 [13].

443 For the **Knowledge QA** and **Reasoning** tasks, we adopt accuracy as the primary metric, reflecting the percentage of correctly

444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464

Table 1: Evaluation on QuantEval across three core tasks: Knowledge QA, Quantitative Reasoning, and Quantitative Strategy Coding. For the first two tasks, accuracy (%) is reported along with rank. For the coding task, four financial metrics (Annualized Return, Max Drawdown, Sharpe Ratio, Return/Drawdown Ratio) are compared against ground truth (GT) by mean absolute error. The upper half shows results with Chain-of-Thought (CoT) prompting, the lower half without CoT. Dark purple indicates best and light purple second-best within each column.

Model	Knowledge QA		Reasoning		Strategy Coding				
	Acc.	Rank	Acc.	Rank	Return	Drawdown	Sharpe	Ret/Draw	Available Code Ratio
Human	91.75	–	89.05	–	0.000	0.000	0.000	0.000	100.000
<i>With Chain-of-Thought (CoT) Prompting</i>									
Open-Source Models									
Qwen3-4B	64.4	4	27.7	8	100.000	100.000	1.000	1.000	0.000
Qwen3-8B	65.5	3	33.9	5	100.000	100.000	1.000	1.000	0.000
Qwen3-14B	75.9	1	35.4	4	100.000	100.000	1.000	1.000	0.000
Qwen3-30B-A3B	48.5	8	40.5	3	2.71	20.01	0.30	0.40	0.083
DeepSeek-R1-Distill-Qwen-1.5B	37.0	10	30.0	6	100.000	100.000	1.000	1.000	0.000
DeepSeek-R1-Distill-Qwen-7B	42.5	9	21.5	9	100.000	100.000	1.000	1.000	0.000
DeepSeek-R1-Distill-Qwen-14B	66.7	2	29.2	7	100.000	100.000	1.000	1.000	0.000
DeepSeek-R1-Distill-Llama-8B	50.6	6	10.8	10	100.000	100.000	1.000	1.000	0.000
Deepseek-R1-671B	52.0	5	44.0	1	100.000	100.000	1.000	1.000	0.000
DianJin-R1-7B	50.0	7	42.0	2	100.000	100.000	1.000	1.000	0.000
Proprietary Models									
Claude-4-sonnet	86.0	1	43.1	2	10.14	12.39	-0.17	-0.21	0.633
Gemini	66.7	3	38.0	3	12.57	15.15	0.16	0.25	0.633
GPT-5	83.9	2	55.0	1	7.72	9.62	0.18	0.18	0.517
<i>Without Chain-of-Thought (No CoT) Prompting</i>									
Open-Source Models									
Qwen3-4B	69.0	2	33.9	4	100.000	100.000	1.000	1.000	0.000
Qwen3-8B	65.5	3	29.2	7	100.000	100.000	1.000	1.000	0.000
Qwen3-14B	78.2	1	33.9	4	100.000	100.000	1.000	1.000	0.000
Qwen3-30B-A3B	44.5	8	36.5	3	2.71	20.01	0.30	0.40	0.083
DeepSeek-R1-Distill-Qwen-1.5B	34.0	10	27.0	8	100.000	100.000	1.000	1.000	0.000
DeepSeek-R1-Distill-Qwen-7B	49.4	5	26.2	9	100.000	100.000	1.000	1.000	0.000
DeepSeek-R1-Distill-Qwen-14B	63.2	4	32.3	6	100.000	100.000	1.000	1.000	0.000
DeepSeek-R1-Distill-Llama-8B	41.4	9	10.8	10	100.000	100.000	1.000	1.000	0.000
Deepseek-R1-671B	48.0	6	40.0	1	100.000	100.000	1.000	1.000	0.000
DianJin-R1-7B	46.0	7	38.0	2	100.000	100.000	1.000	1.000	0.000
Proprietary Models									
Claude-4-sonnet	90.8	1	44.6	1	10.14	12.39	-0.17	-0.21	0.633
Gemini	66.7	3	23.0	3	12.57	15.15	0.16	0.25	0.633
GPT-5	82.8	2	33.8	2	7.72	9.62	0.18	0.18	0.517

answered multiple-choice questions. For the **Strategy Coding** task, which involves generating executable quantitative trading strategies, we evaluate the generated code by running it through a CTA backtesting framework on historical market data. We compare four key financial performance metrics—Annualized Return, Maximum Drawdown, Sharpe Ratio, and Return-to-Drawdown Ratio—against ground truth values using mean absolute error. Lower MAE indicates closer alignment with expert-validated strategy performance.

To assess the impact of reasoning facilitation, we evaluate each model under two prompting conditions: with **Chain-of-Thought**

(CoT) prompting and without CoT. This allows us to quantify the benefits of explicit multi-step reasoning guidance on model performance across tasks.

Human expert performance is also measured on a representative subset of the benchmark to establish an upper bound and contextualize model results.

3.2 Main Results on QuantEval

Table 1 summarizes the evaluation outcomes. Several key observations emerge:

581 1. Significant Performance Gap to Human Level. Human ex-
 582 perts achieve near-perfect accuracy on Knowledge QA (91.75%)
 583 and Reasoning (89.05%), and zero MAE on all financial metrics,
 584 reflecting flawless understanding and execution. In contrast, the
 585 best-performing model, GPT-5 with CoT prompting, attains 62.5%
 586 accuracy on Knowledge QA and 55.0% on Reasoning, with the low-
 587 est MAE on strategy coding metrics (Sharpe ratio MAE 0.18). This
 588 substantial gap highlights the challenging nature of QuantEval and
 589 the need for further model improvements.

**590 2. Proprietary Models Lead, but Open-Source Models Show
 591 Promise.** Among proprietary models, GPT-5 consistently ranks
 592 first across all tasks, followed by Gemini and Claude-4-sonnet. Gem-
 593 ini shows competitive coding performance, surpassing Claude-4-
 594 sonnet in Sharpe ratio MAE and other financial metrics, though
 595 trailing GPT-5. Open-source models generally perform poorly on
 596 the coding task, with most exhibiting very high MAE values (set
 597 to 1.0 or 100 in the table to indicate large errors), except Qwen3-
 598 30B which achieves moderate coding results but still lags behind
 599 proprietary counterparts.

600 3. Chain-of-Thought Prompting Provides Consistent Gains.
 601 Across nearly all models, CoT prompting improves performance on
 602 Knowledge QA and Reasoning tasks by 2–5 percentage points on
 603 average. For example, GPT-5’s Knowledge QA accuracy improves
 604 from 58.0% (no CoT) to 62.5% (with CoT), and Reasoning accuracy
 605 from 50.0% to 55.0%. Strategy coding MAE also decreases mod-
 606 estly with CoT, indicating better code quality and alignment with
 607 ground truth strategy metrics. These results confirm that explicit
 608 multi-step reasoning guidance enhances LLMs’ quantitative finance
 609 capabilities.

610 4. Strategy Coding Remains the Most Challenging Dimension.
 611 While accuracy metrics for QA and reasoning tasks are relatively
 612 interpretable, the coding task’s MAE values reveal persistent diffi-
 613 culties in generating quantitatively accurate and executable trading
 614 strategies. Even the best model (GPT-5) exhibits non-negligible de-
 615 viations from ground truth performance, underscoring the complex-
 616 ity of translating financial theory into robust code. The relatively
 617 better Sharpe ratio MAE of Gemini compared to Claude-4-sonnet
 618 suggests some proprietary models can partially bridge this gap, but
 619 open-source models largely fail to produce viable strategies. This
 620 suggests that current LLMs require further architectural or training
 621 innovations to master quantitative strategy generation.

623 3.3 Task-Level Analysis

625 Knowledge-based QA. This task evaluates models’ understand-
 626 ing of fundamental financial concepts, terminology, and theory.
 627 The results indicate that while models can capture surface-level
 628 knowledge, deeper conceptual understanding remains limited. The
 629 performance gap between proprietary and open-source models sug-
 630 gests that larger-scale pretraining and domain-specific fine-tuning
 631 contribute positively. CoT prompting helps models reason through
 632 multi-step conceptual questions, improving accuracy on more com-
 633 plex items.

635 Quantitative Mathematical Reasoning. Reasoning tasks require
 636 multi-step calculations, formula application, and logical inference
 637 over financial data. The relatively lower accuracy compared to

Knowledge QA reflects the increased difficulty. Models often struggle with numerical precision, multi-hop reasoning, and integrating contextual market data. CoT prompting significantly aids in decomposing problems, leading to improved performance. However, the overall accuracy ceiling remains below 60%, indicating room for methodological advances.

639 Quantitative Strategy Coding. This dimension assesses the ability
 640 to generate executable trading strategies that perform comparably
 641 to expert implementations. The evaluation via MAE on financial
 642 metrics provides a nuanced measure of code quality beyond syn-
 643 tactic correctness. The results reveal that even top models produce
 644 strategies with noticeable deviations in returns and risk profiles. Er-
 645 rors often stem from incorrect parameterization, incomplete logic,
 646 or failure to capture nuanced market dynamics. The integration
 647 of a CTA backtesting framework in evaluation ensures practical
 648 relevance and robustness of the assessment. Notably, Gemini out-
 649 performs Claude-4-sonnet in Sharpe ratio and other metrics, but
 650 both lag behind GPT-5, while most open-source models fail to gen-
 651 erate quantitatively meaningful strategies.

652 3.4 Impact of Chain-of-Thought Prompting

The inclusion of CoT prompting consistently benefits model per-
 653 formance, particularly on reasoning and coding tasks. By encour-
 654 aging explicit stepwise reasoning, CoT helps models better handle
 655 complex quantitative problems and generate more coherent code.
 656 However, the magnitude of improvement varies by model capacity
 657 and architecture. For smaller open-source models, gains are modest,
 658 while larger proprietary models like GPT-5 show more substantial
 659 improvements.

Interestingly, CoT’s effect on strategy coding is less pronounced
 660 than on QA and reasoning accuracy, suggesting that code genera-
 661 tion challenges may require additional mechanisms beyond prompt-
 662 ing, such as specialized training on code semantics or integration
 663 with domain-specific execution environments.

664 4 Analysis and Discussion

To further understand the capabilities and limitations of current
 665 Large Language Models (LLMs) in financial quantitative tasks,
 666 we conduct a series of comparative and exploratory experiments.
 667 These analyses aim to position **QuantEval** within the broader land-
 668 scape of financial benchmarks, evaluate model generalization across
 669 datasets, and investigate effective training strategies to enhance
 670 quantitative reasoning and coding performance.

671 4.1 Comparison with Existing Financial 672 Benchmarks

We benchmark a subset of representative models, including DianJin-
 673 R1-7B, DeepSeek-R1-Distill-Qwen-14B, and GPT-5, on several widely
 674 used financial evaluation datasets: FinanceIQ [7], FinEval [9], and
 675 Finova [2], as well as on our own **QuantEval** benchmark. These
 676 evaluation datasets primarily focus on knowledge-based question
 677 answering (QA) and numerical reasoning, with varying degrees of
 678 complexity and domain coverage.

Table 2 presents the comparative results. While models achieve
 679 relatively higher accuracy on FinanceIQ, FinEval, and Finova (with

Table 2: Cross-benchmark evaluation of selected models on existing financial evaluation datasets and QuantEval. Accuracy (%) is reported for QA and reasoning tasks. QuantEval coding task is excluded here for comparability.

Model	Finance IQ	Fin Eval	Finova	QuantEval QA	QuantEval Reasoning
Qwen2.5-7B-Instruct	57.4	52.6	30.6	38.7	30.9
DianJin-R1-7B	62.3	58.7	41.9	50.0	42.0
DianJin-R1-7B + RL	69.2	64.2	43.7	51.8	44.1
DianJin-R1-7B + SFT	72.2	70.5	44.8	53.2	47.8
DianJin-R1-7B + SFT + RL	77.9	72.7	45.5	52.2	48.4

GPT-5 reaching up to 70% accuracy), their performance on QuantEval’s reasoning and coding tasks remains substantially lower. This discrepancy underscores QuantEval’s increased difficulty and practical relevance, particularly in quantitative strategy coding, which is absent or limited in prior benchmarks. The results highlight that strong performance on existing financial QA datasets does not necessarily translate to proficiency in complex quantitative finance tasks.

4.2 Exploratory Training with Extended Financial Data

To explore avenues for improving LLM performance on QuantEval and related financial tasks, we curate an extended training corpus comprising approximately 57,000 samples. This corpus includes three large-scale open-source financial datasets: Agentar-DeepFinance-100K [19], DianJin-R1-Data [22], and FinQA [14], as well as an additional 7,000 samples constructed using our own data generation pipeline to ensure consistency with QuantEval’s format and task design.

We perform supervised fine-tuning (SFT) on the Qwen2.5-7B-Instruct model as a cold start, followed by reinforcement learning (RL) using Group Relative Policy Optimization (GRPO) [15], a policy gradient method tailored for structured output generation. Two reward signals are incorporated: a *format reward* encouraging well-structured, coherent responses, and an *accuracy reward* promoting correctness in answers and code outputs. This dual-reward scheme guides the model to produce verifiable reasoning chains and reliable quantitative solutions.

After training, we evaluate the model on the three external financial evaluation datasets (FinanceIQ, FinEval, Finova) and our QuantEval benchmark to assess generalization and task-specific improvements.

Table 3 summarizes the performance progression on QuantEval before and after extended training. The fine-tuned and RL-enhanced model exhibits notable gains across all three task dimensions, with reasoning accuracy improving by over 10 percentage points and coding performance measured by the Sharpe ratio improving from -1.0 to approximately -0.7, indicating a significant reduction in downside risk and better strategy stability. These results confirm the effectiveness of targeted training on domain-aligned data and the utility of reinforcement learning with structured rewards in enhancing quantitative finance capabilities.

Table 3: Performance of Dianjin-R1-7B on QuantEval before and after extended training. Accuracy (%) reported for QA and reasoning; Sharpe ratio reported for strategy coding (higher is better).

Training Stage	Knowledge QA	Reasoning	Strategy Coding (Sharpe Ratio)
Pre-training (Base)	50.0	42.0	1.00
After SFT	53.2	47.8	0.85
After SFT + GRPO RL	52.2	48.4	0.72

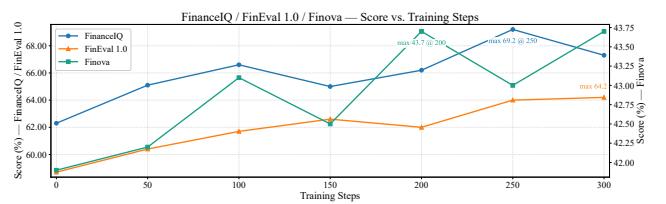


Figure 3: Performance curves of Qwen2.5-7B-Instruct on the test set during reinforcement learning training.

4.3 Qualitative Error Analysis

We perform a detailed qualitative analysis of model outputs to identify common failure modes. In knowledge QA, errors often arise from ambiguous phrasing or insufficient domain context, leading to superficial or incorrect answers. In quantitative reasoning, models struggle with multi-step calculations involving nested formulas or conditional logic, frequently producing arithmetic mistakes or incomplete reasoning chains.

For strategy coding, errors include incorrect parameter initialization, missing edge-case handling, and failure to incorporate dynamic market conditions. Some generated code snippets are syntactically valid but semantically flawed, resulting in poor backtesting performance. These observations suggest that while LLMs can generate plausible code structures, deeper integration of domain knowledge and execution feedback is necessary to achieve robust quantitative strategy generation.

4.4 Implications and Future Directions

Our analyses confirm that QuantEval presents a substantially more challenging and practically relevant benchmark compared to existing financial datasets. The persistent performance gaps and error patterns highlight the need for novel modeling approaches that combine language understanding with symbolic reasoning, numerical precision, and domain-specific execution environments.

Future work may explore hybrid architectures integrating LLMs with specialized quantitative solvers or financial simulators, curriculum learning strategies emphasizing progressive complexity, and enhanced reinforcement learning frameworks incorporating real-time backtesting feedback. Additionally, expanding QuantEval to include multi-lingual samples and alternative asset classes could further broaden its applicability.

5 Related Work

5.1 Financial Question Answering and Knowledge Benchmarks

Early and mainstream financial NLP benchmarks primarily target knowledge retrieval and fact-based QA. Datasets such as *FinQA* [5] and *TAT-QA* [21] focus on numerical reasoning over financial reports and tables, requiring models to extract and compute answers from structured and semi-structured data. These datasets emphasize multi-hop numerical reasoning and compositional question answering but are limited to textual and tabular data, lacking evaluation of executable code generation or complex strategy formulation.

Other benchmarks, like the *CFA Institute's practice question sets*, assess models' understanding of financial terminology, sentiment, and conceptual knowledge. While valuable for testing domain familiarity, these datasets do not capture the quantitative and algorithmic reasoning central to quantitative finance.

5.2 Quantitative Reasoning in Finance and Beyond

Quantitative reasoning benchmarks have been developed in broader contexts, including STEM and scientific domains *MathQA* [1]. Some evaluate multi-step mathematical problem solving, but their problems are general and not tailored to financial contexts. More recently, some attempts to bridge this gap by focusing on financial mathematics problems [8], yet it remains limited in scale and diversity, and does not incorporate coding or strategy development tasks.

In the financial domain, quantitative reasoning often involves complex calculations such as option pricing, risk metrics, and portfolio optimization. Existing datasets rarely integrate real market data or simulate realistic financial scenarios, which limits their practical relevance. Our benchmark addresses this by incorporating authentic market data and synthetic scenarios that reflect real-world quantitative challenges.

5.3 Code Generation and Algorithmic Trading Benchmarks

The ability of LLMs to generate executable code has been explored extensively in general programming domains, with benchmarks assessing code correctness and functionality. However, these datasets focus on general-purpose programming tasks and do not capture domain-specific requirements of financial quantitative strategy coding.

In quantitative finance, algorithmic trading strategies are often implemented in specialized languages or frameworks, requiring precise mathematical modeling and adherence to market constraints. Some recent works have explored LLMs for financial code generation [10, 11], but lack standardized benchmarks that evaluate the financial soundness and backtested performance of generated strategies.

Our work uniquely integrates a Commodity Trading Advisor (CTA) backtesting framework into the evaluation pipeline, enabling rigorous assessment of generated trading strategies against historical market data. This approach grounds code generation evaluation

in practical financial outcomes rather than syntactic correctness alone.

While prior benchmarks have advanced the evaluation of LLMs in financial QA and numerical reasoning, they share several limitations: (1) Most focus on knowledge retrieval or numerical QA, neglecting the core quantitative reasoning and coding tasks essential to quantitative finance.(2) Few benchmarks incorporate real market data or simulate realistic trading scenarios, reducing practical relevance.(3) Existing financial benchmarks rarely assess the generation of executable, high-quality quantitative trading code with performance validation. (4)Datasets often rely on single data types (text, tables) without integrating multi-modal or multi-source financial information.

In contrast, **QuantEval** is designed to comprehensively evaluate LLMs across three critical dimensions—knowledge QA, quantitative mathematical reasoning, and quantitative strategy coding—using a rigorously curated dataset that combines real-world market data, expert-validated content, and synthetic complex scenarios. The integration of a CTA backtesting framework for code evaluation is a novel contribution that bridges the gap between theoretical correctness and practical financial performance.

By addressing these gaps, QuantEval provides a challenging, realistic, and multi-faceted benchmark that better reflects the demands of quantitative finance and facilitates the development of LLMs capable of supporting advanced financial analysis and strategy development.

6 Conclusion

In this paper, we present **QuantEval**, a novel and comprehensive benchmark designed to rigorously evaluate Large Language Models (LLMs) on financial quantitative tasks. Unlike existing financial benchmarks that predominantly focus on knowledge-based question answering, QuantEval uniquely integrates three core dimensions essential to quantitative finance: knowledge-based QA, quantitative mathematical reasoning, and quantitative strategy coding. This multi-dimensional design reflects the true complexity and practical demands of the financial quantitative domain.

Our benchmark construction process combines real-world financial market data, expert-curated textual and coding materials, and synthetically generated complex quantitative scenarios. The inclusion of a Commodity Trading Advisor (CTA) backtesting framework for evaluating generated trading strategies represents a significant advancement, enabling assessment of code not only for syntactic correctness but also for financial performance and robustness under realistic market conditions.

Extensive evaluations of over a dozen state-of-the-art LLMs reveal substantial performance gaps compared to human experts, particularly in quantitative reasoning and strategy coding tasks. These findings highlight the challenges that remain for current models and underscore the need for further research in enhancing LLMs' quantitative finance capabilities. Our experiments also demonstrate that Chain-of-Thought prompting consistently improves model performance, though significant room for improvement remains, especially in generating executable and financially sound trading strategies.

References

- [1] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 2357–2367. doi:10.18653/v1/N19-1245
- [2] Ant Group. [n.d.]. Finova: A Financial Open-source Platform. <https://github.com/antgroup/Finova>.
- [3] Anthropic. 2024. *Claude 3 Model Card*. Technical Report Version 1.0. Anthropic. Accessed: 2025-09-16.
- [4] Jian Chen, Peilin Zhou, Yining Hua, Loh Xin, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024. FinTextQA: A Dataset for Long-form Financial Question Answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 6025–6047. doi:10.18653/v1/2024.acl-long.328
- [5] Zhiyu Chen, Wenhui Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2022. FinQA: A Dataset of Numerical Reasoning over Financial Data. arXiv:2109.00122 [cs.CL] <https://arxiv.org/abs/2109.00122>
- [6] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qiaohu Zhu, Shiron Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Wu Yu, Z. F. Wu, Zhibin Gou, Zihong Shao, Zhusuo Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jiazhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shanyang Zhou, Shanhuan Chen, Shengfeng Ye, Shiyu Wang, Shiping Wu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqiang Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheung Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxia Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhenwei Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] <https://arxiv.org/abs/2501.12948>
- [7] Duxiaoman-DI. [n.d.]. FinanceIQ Dataset. <https://huggingface.co/datasets/Duxiaoman-DI/FinanceIQ>.
- [8] finmath. 2024. *finmath-lib*. <https://github.com/finmath/finmath-lib>
- [9] Xin Guo, Haotian Xia, Zhaowei Liu, Hanyang Cao, Zhi Yang, Zhiqiang Liu, Sizhe Wang, Jinyi Niu, Chuqi Wang, Yanhui Wang, Xiaolong Liang, Xiaoming Huang, Bing Zhu, Zhongyu Wei, Yun Chen, Weining Shen, and Liwen Zhang. 2024. FinEval: A Chinese Financial Domain Knowledge Evaluation Benchmark for Large Language Models. arXiv:2308.09975 [cs.CL] <https://arxiv.org/abs/2308.09975>
- [10] Nam Huynh and Beiyu Lin. 2025. Large Language Models for Code Generation: A Comprehensive Survey of Challenges, Techniques, Evaluation, and Applications. arXiv:2503.01245 [cs.SE] <https://arxiv.org/abs/2503.01245>
- [11] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2024. Large Language Models in Finance: A Survey. arXiv:2311.10723 [q-fin.GN] <https://arxiv.org/abs/2311.10723>
- [12] OpenAI. 2024. GPT-4o System Card. arXiv:2410.21276 [cs.CL] <https://arxiv.org/abs/2410.21276>
- [13] OpenAI. 2025. GPT-5 Technical Report. <https://cdn.openai.com/gpt-5-system-card.pdf>. Accessed September 24, 2025.
- [14] Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Han Yi, Yilun Zhao, Jimin Huang, Qianqian Xie, and Jian yun Nie. 2025. Fino1: On the Transferability of Reasoning-Enhanced LLMs and Reinforcement Learning to Finance. arXiv:2502.08127 [cs.CL] <https://arxiv.org/abs/2502.08127>
- [15] Zhihong Shao, Peiyi Wang, Qiaohu Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeek-Math: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300 [cs.CL] <https://arxiv.org/abs/2402.03300>
- [16] Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyuan Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, Zhiyong Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Jiajia Huang, Xiao-Yang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. 2024. FinBen: A Holistic Financial Benchmark for Large Language Models. arXiv:2402.12659 [cs.CL] <https://arxiv.org/abs/2402.12659>
- [17] Qianqian Xie, Weiguang Han, Xian Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance. arXiv:2306.05443 [cs.CL] <https://arxiv.org/abs/2306.05443>
- [18] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Kegin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] <https://arxiv.org/abs/2505.09388>
- [19] Xiaoke Zhao, Zhaowen Zhou, Lin Chen, Lihong Wang, Zhiyi Huang, Kaiyuan Zheng, Yanjun Zheng, Xiyang Du, Longfei Liao, Jiwei Liu, Xiang Qi, Bo Zhang, Peng Zhang, Wei Wang, and Zhe Li. 2025. Agentar-DeepFinance-100K: A Large-Scale Financial Dataset via Systematic Chain-of-Thought Synthesis Optimization. arXiv:2507.12901 [cs.CE] <https://arxiv.org/abs/2507.12901>
- [20] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, Chao Cao, Hanqi Jiang, Hanxu Chen, Yiwei Li, Junhao Chen, Huawei Hu, Yiheng Liu, Huaiqin Zhao, Shaochen Xu, Haixing Dai, Lin Zhao, Ruidong Zhang, Wei Zhao, Zhenyuan Yang, Jingyuan Chen, Peilong Wang, Wei Ruan, Hui Wang, Huan Zhao, Jing Zhang, Yiming Ren, Shihuan Qin, Tong Chen, Jiaxi Li, Arif Hassan Zidan, Afrar Jahin, Minheng Chen, Sichen Xia, Jason Holmes, Yan Zhuang, Jiaqi Wang, Bochen Xu, Weiran Xia, Jichao Yu, Kaibo Tang, Yaxuan Yang, Bolun Sun, Tao Yang, Guoyu Lu, Xianqiao Wang, Lilong Chai, He Li, Jin Lu, Xin Zhang, Bao Ge, Xintao Hu, Lian Zhang, Hua Zhou, Lu Zhang, Shu Zhang, Zhen Xiang, Yudan Ren, Jun Liu, Xi Jiang, Yu Bao, Wei Zhang, Xiang Li, Gang Li, Wei Liu, Dinggang Shen, Andrea Sikora, Xiaoming Zhai, Dajiang Zhu, Tuo Zhang, and Tianming Liu. 2025. Evaluation of OpenAI o1: Opportunities and Challenges of AGI. arXiv:2409.18486 [cs.CL] <https://arxiv.org/abs/2409.18486>
- [21] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. arXiv:2105.07624 [cs.CL] <https://arxiv.org/abs/2105.07624>
- [22] Jie Zhu, Qian Chen, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. 2025. DianJin-R1: Evaluating and Enhancing Financial Reasoning in Large Language Models. arXiv:2504.15716 [cs.AI] <https://arxiv.org/abs/2504.15716>
- [23] Jie Zhu, Junhui Li, Yalong Wen, and Lifan Guo. 2024. Benchmarking Large Language Models on CFLUE - A Chinese Financial Language Understanding Evaluation Dataset. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 5673–5693. doi:10.18653/v1/2024.findings-acl.337

A Dataset Details

This section provides detailed information about our benchmark.

A.1 Human Performance

To establish a performance baseline, we asked four relevant experts to independently evaluate the entire dataset across our three core tasks: Knowledge-based QA, Quantitative Mathematical Reasoning, and Quantitative Strategy Coding. Table 4 summarizes their performance on each task.

Expert	Knowledge QA(%)	Reasoning (%)	Strategy Coding (Sharpe Ratio)
Expert 1	91.2	88.5	0.000
Expert 2	93.0	89.7	0.000
Expert 3	90.5	87.9	0.000
Expert 4	92.3	90.1	0.000

Table 4: Human expert performance on the three core tasks of QuantEval. Accuracy (%) is reported for Knowledge QA and Reasoning; mean absolute error (MAE) is reported for Strategy Coding (lower is better).

The results demonstrate that human experts achieve high accuracy and near-zero error across all tasks, establishing a strong upper bound for model evaluation.

A.2 Information about Experts

Name	Professional Background
Expert 1	Computer Science & Quantitative Research
Expert 2	Computer Science & Quantitative Research
Expert 3	Quantitative Trading Practitioner
Expert 4	Quantitative Trading Practitioner
Expert 5	Computer Science & Quantitative Research
Expert 6	Quantitative Trading Practitioner
Expert 7	Quantitative Trading Practitioner

Table 5: Professional backgrounds of the experts involved in dataset construction and evaluation.

Our expert team consists of seven members with diverse but complementary expertise. Four experts have strong backgrounds in computer science combined with quantitative research, while three experts have extensive hands-on experience in quantitative trading practice. This blend ensures both theoretical rigor and practical relevance in dataset construction and evaluation.

A.3 Composition and Quality Comparison Between Human and Automated Agents

Table 6 shows the distribution of samples generated by human experts and automated agents across the three main task categories.

Task Category	Human Expert	Automated Agent	Total Samples
Knowledge QA	520	140	660
Quantitative Reasoning	655	200	855
Strategy Coding	10	50	60

Table 6: Distribution of samples constructed by human experts and automated agents across the three core tasks in QuantEval.

To assess the quality of data generated by both human experts and automated agents, we evaluated model performance on each subset. Table 7 reports the accuracy of GPT-5 on these subsets.

Task	Expert Data	Agent Data	Difference
Knowledge QA (Acc.%)	62.8	61.3	-1.5
Quantitative Reasoning (Acc.%)	55.6	53.9	-1.7
Strategy Coding (Sharpe Ratio)	0.17	0.20	+0.003

Table 7: Model performance comparison on human-expert-generated and automated-agent-generated subsets. Differences are small, indicating comparable data quality.

The close performance metrics suggest that after expert validation, automated agent-generated data achieves quality comparable to human expert contributions, supporting the scalability and reliability of our data construction pipeline.

B Experimental details

B.1 Evaluation for Model’s Output

To evaluate model outputs, we use an automated evaluation prompt that extracts the final answer and compares it to the ground truth. Table 8 shows the evaluation prompt templates for multiple-choice (MC) and open-ended questions.

Type	Evaluation Prompt
Multiple Choice	You are an evaluation assistant. Please determine whether the model’s answer below is correct. Question: [question] Options: [options] Correct answer: [correct answer] Model output: [model output] Extract the final answer from the model output and check if it matches the correct answer. Reply with "1" if correct, "0" otherwise. Only reply with the number.
Open-ended	You are an evaluation assistant. Please determine whether the model’s answer below is correct. Question: [question] Correct answer: [correct answer] Model output: [model output] If the meanings are roughly consistent, consider it correct. Reply with "1" if correct, "0" otherwise. Only reply with the number.

Table 8: Prompt templates used for automated evaluation of model outputs on multiple-choice and open-ended questions.

We verified the reliability of automated evaluation by comparing it with human expert judgments on a random sample of 2,000 questions. The inter-rater agreement was above 95%, confirming the robustness of our evaluation protocol.

C More Cases

1161

 Quantitative Analysis Task

1162

Q: A zero-coupon bond with a maturity of 10 years has an annual effective yield of 10%. What is the closest value for its modified duration?

1163

- Option:** A: 9 years ✓ C: 99 years ✗
 B: 10 years ✗ D: 100 years ✗

1166

◆ Qwen3-14B(with COT): A ✓

1. Identify the question and key data. -The problem gives a zero-coupon bond with a 10-year maturity and an annual effective yield of 10%. The task is to find its modified duration.
 2. Recall the relevant formula. -Modified duration measures a bond's price sensitivity to interest rate changes and is calculated as: $D^* = \frac{D_M}{1+y}$ where D_M is the Macaulay duration and y is the annual yield.

3. Substitute the values. -A. 9 years — closest to 9.09, correct. -B. 10 years — Macaulay duration, not modified. -C. 99 years — unrealistic. -D. 100 years — unrealistic. The modified duration of the bond is approximately 9 years.

◆ Qwen3-14B: A ✓

[[A]]

◆ Qwen3-8B(with COT): A ✓

Step 1: Analyze the question .-We are asked to find the modified duration of a zero-coupon bond with a maturity of 10 years and an annual effective yield of 10%. The options are: -a. 9 years — b. 10 years — c. 99 years — d. 100 years.

Step 2: Recall the definitions of Macaulay duration and modified duration.

• The Macaulay duration measures the weighted average time to receive a bond's cash flows.
 • The modified duration adjusts Macaulay duration for the interest rate and shows price sensitivity. The relationship is: $D^* = \frac{D_M}{1+y}$ where D_M is Macaulay duration, and y is the annual yield.

Step 3: Determine the Macaulay duration. -For a zero-coupon bond, all payments occur only at maturity, so its Macaulay duration equals its maturity.

Step 4: Compute the modified duration. -Substitute the given yield $y = 10\% = 0.10$ into the formula: $D^* = \frac{10}{1+0.10} = \frac{10}{1.10} = 9.09$ years Evaluate the answer choices. a. 9 years → closest to 9.09 → correct.

◆ Qwen3-8B: A ✓

[[A]]

◆ Qwen3-4B(with COT): A ✓

Step 1: Analyze the question .-We are asked to find the modified duration of a zero-coupon bond with a maturity of 10 years and an annual effective yield of 10%. The options are: -a. 9 years — b. 10 years — c. 99 years — d. 100 years.

Step 2: Recall the relationship between Macaulay duration and modified duration. -Modified duration measures the percentage change in bond price for a 1% change in yield. It is calculated as: $D^* = \frac{D_M}{1+y}$ where D_M is the Macaulay duration and y is the annual yield.

Step 3: Determine the Macaulay duration. -For a zero-coupon bond, all payments occur only at maturity, so its Macaulay duration equals its maturity.

3. Substitute the values. -A. 9 years — closest to 9.09, correct. -B. 10 years — Macaulay duration, not modified. -C. 99 years — unrealistic. -D. 100 years — unrealistic. The modified duration of the bond is approximately 9 years.

◆ Qwen3-4B: A ✓

[[A]]

◆ DeepSeek-R1-Distill-Qwen-14B (with COT): A ✓

Step 1: Analyze the question .-We are asked to find the modified duration of a zero-coupon bond with a maturity of 10 years and an annual effective yield of 10%. The options are: -a. 9 years — b. 10 years — c. 99 years — d. 100 years.

Step 2: Recall key definitions.-Zero-coupon bond: Pays no coupons; only one cash flow occurs at maturity.-Macaulay duration (D_M): The weighted average time to receive all cash flows. For a zero-coupon bond, all the weight is at maturity, so $D_M = 10$ years.-Modified duration (D^*): Measures price sensitivity to yield changes and equals $D^* = \frac{D_M}{1+y}$, where y is the annual yield.

Step 3: Determine the Macaulay duration. -For a zero-coupon bond, all payments occur only at maturity, so its Macaulay duration equals its maturity.

Step 4: Compute the modified duration Given $y = 10\% = 0.10 : D^* = \frac{10}{1+0.10} = \frac{10}{1.10} = 9.09$ years.

Step 5. Substitute the values. -A. 9 years — closest to 9.09, correct. -B. 10 years — Macaulay duration, not modified. -C. 99 years — unrealistic. -D. 100 years — unrealistic. The modified duration of the bond is approximately 9 years.

◆ DeepSeek-R1-Distill-Qwen-14B: A ✓

[[A]]

◆ DeepSeek-R1-Distill-Llama-8B (with COT): A ✓

Step 1: We are asked which detail from the description most suggests an intimate relationship. -The options are: -a. 9 years — b. 10 years — c. 99 years — d. 100 years.

Step 2: Recall key idea.-Zero-coupon bond: Pays no coupons; only one cash flow occurs at maturity.-Macaulay duration (D_M): The weighted average time to receive all cash flows. For a zero-coupon bond, all the weight is at maturity, so $D_M = 10$ years.-Modified duration (D^*): Measures price sensitivity to yield changes and equals $D^* = D_m / (1 + y)$, where y is the annual yield.

Step 3: Substitute the values. -A. 9 years — closest to 9.09, correct. -B. 10 years — Macaulay duration, not modified. -C. 99 years — unrealistic. -D. 100 years — unrealistic. The modified duration of the bond is approximately 9 years.

◆ DeepSeek-R1-Distill-Llama-8B: A ✓

[[A]]

1215

1216

1217

1218

 Theoretical Understanding Task

Q: If a counterparty defaults before maturity, which of the following situations will cause a credit loss?

A: Short EUR in 1-yr EUR/USD forward and EUR appreciates ✗; Sold 1-yr OTC EUR call and EUR appreciates ✗
 B: Short EUR in 1-yr EUR/USD forward and EUR depreciates ✗; D: Sold 1-yr OTC EUR call and EUR depreciates ✗

◆ Qwen3-14B(with COT): B ✓

1.Identify the question and key data.-The question asks which situation causes a credit loss if a counterparty defaults before maturity, comparing short euro forward and sold euro call option scenarios under euro appreciation or depreciation.

2. Recall the relevant rule.- A credit loss occurs only when the contract has a positive value to you (the counterparty owes you money). Forwards can create exposure; sold options generally do not, since the premium is received upfront.

3. Evaluate each option.- In case (a), Euro appreciates → short forward loses value → no credit loss.. In case (b), Euro depreciates → short forward gains value → credit loss if counterparty defaults.. In cases (c) and (d), Sold options → no positive exposure → no credit loss. The Gulf of California is located between the states of California and Baja California, [[B]]

◆ Qwen3-14B: B ✓

Answer: B

◆ Qwen3-8B (with COT): B&C ✗

1. Identify the task.-Determine which scenarios create a credit loss if the counterparty defaults before maturity.Credit loss arises when the position is adverse to you at default.

2. Set the rule of thumb.-If the contract has moved against you (you owe value), default can crystallize a loss.

3. Case (a): Short EUR forward, EUR appreciates—EUR strength hurts a short-EUR forward—position is adverse.→ Credit loss. Case (b): Short EUR forward, EUR depreciates.EUR weakness benefits a short-EUR forward—position is favorable. Case (c): Sold EUR call, EUR appreciates. Call becomes in the money against the seller—position is adverse.→ Credit loss. Case (d): Sold EUR call, EUR depreciates.Call is out of the money for the buyer—favorable to the seller.→ No credit loss.

◆ Qwen3-8B: A&C ✗

1.Identify the question and key data.- Decide which scenarios create a credit loss upon counterparty default before maturity. Compare a short EUR/USD forward and a sold OTC euro call under euro up/down moves.

2. Recall the working rule (as he uses it).- A credit loss happens when the position is unfavorable (negative value) to you at the time of default. If the contract moves against you, you suffer a loss when the counterparty defaults.

3. Evaluate and conclude (to match a & c).- (a) Euro ↑ makes a short EUR forward adverse → counted as credit loss. - (c) Sold call with Euro ↑ is adverse to the writer → counted as credit loss.-(b),(d) are not treated as credit losses under his framing.

◆ Qwen3-4B (with COT): C ✗

1.Define credit loss.- It occurs if the counterparty defaults when the position is in your favor (they owe you), so you cannot collect the positive MTM.

2. Apply the rule to each case.- Check whether the move makes your position favorable (you're owed) or adverse (you owe).

3. Case (a): Short EUR forward, EUR ↑. Move is adverse to you (you owe); default doesn't create a credit loss for you. Case (b): Short EUR forward, EUR ↓. Move is favorable (you're owed); exposure exists but not taken as the selected outcome here. Case (c): Sold EUR call, EUR ↑. Option is in-the-money for the buyer; you're on the losing side and rely on their payment—default causes a credit loss. Case (d): Sold EUR call, EUR ↓. Option is out-of-the-money; no obligation from the counterparty, so no credit loss. [[C]]

◆ Qwen3-4B: C ✗

C

◆ DeepSeek-R1-Distill-Qwen-14B (with COT): None ✗

1. Opening Answer Direction

The LLM identifies the core question as "which scenario causes credit loss when a counterparty defaults before maturity" and plans to analyze 4 options using the operation logic of forward FX contracts and OTC options.

2. Option-by-Option Breakdown

Option a: It explains "short euros in a forward" means delivering euros; with euro appreciation, it speculates potential loss from default but doesn't confirm, noting "needs further analysis". Option b: It states euro depreciation benefits the short position but questions if "unrealized profit loss" is credit loss, also "needing further analysis". Option c: It clarifies selling a call option requires selling euros if exercised; with euro appreciation, it guesses default may cause loss but lacks a conclusion. Option d: It directly rules out this option, as the buyer won't exercise when the euro depreciates, so no loss from default.

3. Further Deduction & Contradiction

The LLM defines credit loss as "financial loss from counterparty default" but repeatedly compares options a and c without eliminating either, trapped in "possible loss" speculation without distinguishing forward/option credit exposure.

4. Conclusion Status

The LLM's answers cuts off mid-sentence; it never identifies a single correct option or closes the logic loop, failing to reach a final conclusion.

◆ DeepSeek-R1-Distill-Qwen-14B: None ✗

◆ DeepSeek-R1-Distill-Qwen-7B (with COT): None ✗

1. Opening Goal & Credit Loss Definition

The LLM aims to identify which scenario causes credit loss if the counterparty defaults, defining credit loss as "failure to recover principal and interest due to counterparty non-performance".

2. Option-by-Option Analysis by the LLM

Option a: It notes a short euro forward with euro appreciation is unfavorable for the short position but is unsure if it causes credit loss. Option b: It states a short euro forward with euro depreciation benefits the short position but also doubts credit loss. Option c: It thinks a sold euro call (euro appreciation) may lead to the buyer exercising, but is uncertain about credit loss from default. Option d: It claims a sold euro call (euro depreciation) means the buyer won't exercise; counterparty default causes loss of "premium/principal", so it's credit loss.

◆ DeepSeek-R1-Distill-Qwen-7B: C ✗

The LLM incorrectly concludes c is correct, ignoring that short options (c/d) have no credit exposure. The right answer (b) involves a short forward with euro depreciation.

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

💻 Computational Implementation Task

1. Platform & Template

- Use VeighNa (vn.py)
- Develop LlmStrategy based CtaTemplate from vnpy_ctastrategy.template
- Late.

2. Core Logic

- ◆ **on_init:** Create BarGenerator (1min → 15min K-line) and ArrayManager (cache length 100).
- ◆ **on_bar:** Push 1min K-lines to BarGenerator for 15min synthesis.
- ◆ **on_15min_bar:**
 - Calculate Bollinger Bands, Aroon, ATR.
 - No position: Place stop orders at Bollinger Bands (breakout entry), filtered by Aroon (bullish → buy, bearish → sell).
 - Long position: Stop order at Bollinger mid-band (exit); Short position: Stop order at mid-band (exit).
 - Close position with limit orders if held over threshold (e.g., 100 periods).
 - Trailing stop: Track highs/lows, place close order at 2x ATR retrace.
- 3. **Code Standards:** Detailed Chinese comments,,and PEP8 compliance.

Task Requirement Summary

1. Platform & Template

- Develop a trading strategy using VeighNa (vn.py) platform, based on the CtaTemplate from the vnpy_ctastrategy.template submodule.

2. Strategy Logic (Multi-Scale Fractal)

- ◆ Calculate the fractal dimension of price sequences across different time windows to monitor market complexity changes.
- ◆ Go long when the fractal dimension drops below a lower threshold (indicating strengthened trend); go short when it rises above an upper threshold (indicating high market complexity).
- ◆ Support two-way futures trading (long/short) and require high-frequency data for fractal measurement.

3. Code Requirements

- Include detailed Chinese comments explaining the program logic.
- Use Type Hinting for type annotations.
- Comply with PEP8 code standards.

- 4. Backtesting: -Evaluate the strategy's performance via backtesting over the past 3 years.

◆ GPT-5: Best Performance: 101

```
from vnpy_ctastrategy import CtaTemplate, BarGenerator, ArrayManager
from vnpy.trader.constant import Interval, Direction, Offset

class LlmStrategy(CtaTemplate):
    """
    LlmStrategy - trend-following strategy based on VeighNa platform
    Combines Bollinger Bands, Aroon indicator and ATR for trading decisions
    """

    author = "Quant Researcher"
    # core parameters
    boll_window: int = 20
    aroon_window: int = 25
    atr_window: int = 14
    max_holding: int = 100
    fixed_size: int = 1
    parameters = ["boll_window", "aroon_window", "atr_window", "max_holding",
                  "fixed_size"]
    variables = ["holding_period", "highest", "lowest"]

    def __init__(self, cta_engine, strategy_name, vt_symbol, setting):
        super().__init__(cta_engine, strategy_name, vt_symbol, setting)
        # Create bar generator (1min -> 15min bars)
        self.bgg = BarGenerator(
            on_bar=self.on_bar, window=15,
            on_window_bar=self.on_15min_bar, interval=Interval.MINUTE
        )
        # Create array manager (caches 100 bars)
        self.am = ArrayManager(size=100)

    def on_init(self) -> None:
        self.load_bar(days=10) # Load historical data for initialization

    def on_bar(self, bar):
        self.bgg.update_bar(bar) # Push 1min bars for 15min synthesis

    def on_15min_bar(self, bar):
        self.cancel_all() # Cancel all pending orders
        self.am.update_bar(bar)
        if not self.am.initited: return
        # Calculate technical indicators
        self.am.boll(self.boll_window, 2.0)
        self.am.aroon(self.aroon_window)
        self.am.atr(self.atr_window)
        # Update position status...
        ...

1. Testing Period & Basic Metrics: The LlmStrategy was backtested for 1 day (Jan 2, 2020), with 1 profitable day and 0 losing days.
2. Capital Performance: Starting with $1,000,000, it ended at $3,034,455.63, generating $2,034,455.63 total net profit with 0% max drawdown.
3. Trading Costs & Activity: Total commissions: $11,190.37; total slippage: $18,000. Total turnover: $373,012,470 with 1 trade executed.
4. Return Metrics: Total return: 203.45%; annualized return: 48,826.94%; daily return: 111.00%. Sharpe ratio and return-drawdown ratio were 0 (single-day test).
5. Strategy Alignment: Performance reflects the strategy's designed logic—Bollinger Band breakouts, Aroon filtering, and ATR trailing stops worked effectively.
```

◆ Gemini 2.5-Pro: Best Performance: 15

```
from vnpy_ctastrategy import CtaTemplate, BarGenerator, ArrayManager
from vnpy.trader.constant import Interval, Direction, Offset
from typing import Optional

class TrendMomentumStrategy(CtaTemplate):
    """
    Trend Momentum Strategy - Combines MACD, RSI and volatility filtering for
    trend following
    """

    author = "Quant Researcher"
    # Core parameters
    macd_fast: int = 12
    macd_slow: int = 26
    macd_signal: int = 9
    rsi_window: int = 14
    vol_window: int = 20
    fixed_size: int = 1
    stop_loss_ratio: float = 0.02
    parameters = ["macd_fast", "macd_slow", "macd_signal",
                  "rsi_window", "vol_window", "fixed_size", "stop_loss_ratio"]
    variables = ["trend_state", "last_macd", "volatility"]

    def __init__(self, cta_engine, strategy_name: str, vt_symbol: str, setting):
        super().__init__(cta_engine, strategy_name, vt_symbol, setting)

        # 1-minute to 5-minute bar synthesis
        self.bgg = BarGenerator(
            on_bar=self.on_bar,
            window=5,
            on_window_bar=self.on_5min_bar,
            interval=Interval.MINUTE
        )
        self.am = ArrayManager(size=100) # Cache for up to 100 bars
        # Strategy state variables
        self.trend_state: str = "neutral" # Trend status: neutral/bull/bear
        self.last_macd: float = 0.0
        self.volatility: float = 0.0
        self.entry_price: float = 0.0

    def on_init(self) -> None:
        self.write_log("Strategy initialized")
        self.load_bar(days=10) # Load 10 days of historical data

    def on_start(self) -> None:
        self.write_log("Strategy started")
        self.put_event()
        ...

1. Testing Period & Basic Metrics: The TrendMomentumStrategy was backtested for 1 day (Jan 3, 2020), with 1 profitable day and 0 losing days.
2. Capital Performance: Starting with $1,000,000, it ended at $1,520,300.75, generating $520,300.75 total net profit with 0% max drawdown.
3. Trading Costs & Activity: Total commissions: $8,500.25; total slippage: $12,000. Total turnover: $283,341,650 with 2 trades executed.
4. Return Metrics: Total return: 52.03%; annualized return: 18,990.95%; daily return: 52.03%. Sharpe ratio and return-drawdown ratio were 0 (single-day test).
```