CIO-Agent FAB++: A Dynamic Multi-Dimensional Benchmark for Evaluating AI Finance Agents

Team AgentBusters
AgentBeats Competition 2026
https://github.com/yxc20089/AgentBusters

January 16, 2026

Abstract

We present CIO-Agent FAB++ (Finance Agent Benchmark Plus Plus), a comprehensive evaluation framework for assessing AI agents on financial analysis tasks. Unlike static benchmarks, FAB++ dynamically generates evaluation tasks across 18 categories spanning fundamental analysis, quantitative reasoning, options trading, and risk management. The system implements a novel multi-dimensional scoring methodology that evaluates agents on macro thesis quality, fundamental accuracy, execution methodology, and adversarial robustness. We introduce the Options Alpha Challenge, a specialized evaluation track that tests agents on Black-Scholes pricing, Greeks analysis, and multi-leg strategy construction. Our framework leverages the Agent-to-Agent (A2A) protocol for standardized communication and Model Context Protocol (MCP) servers for real-time financial data access. Experimental results demonstrate the effectiveness of our evaluation methodology in distinguishing agent capabilities across diverse financial reasoning tasks.

Keywords: AI Agents, Finance Benchmark, Options Trading, Agent Evaluation, A2A Protocol, MCP

1 Introduction

The rapid advancement of large language models (LLMs) has enabled the development of sophisticated AI agents capable of performing complex financial analysis tasks [Brown et al., 2020]. However, evaluating these agents presents significant challenges: financial reasoning requires numerical precision, temporal awareness, and domain expertise that traditional NLP benchmarks fail to capture adequately.

Existing finance benchmarks suffer from several limitations:

1. **Static evaluation**: Fixed question sets become memorized by models during training, leading to inflated performance metrics.
2. **Single-dimensional scoring**: Most benchmarks evaluate only answer correctness, ignoring reasoning quality and methodology.
3. **Lack of temporal constraints**: Agents may inadvertently access future information, violating realistic trading scenarios.
4. **Limited options coverage**: Few benchmarks evaluate quantitative finance skills like derivatives pricing and risk management.

We address these limitations with CIO-Agent FAB++, a dynamic benchmark system that:

- Generates novel evaluation tasks from real financial data with temporal locking
- Evaluates agents across multiple dimensions including macro reasoning, fundamental accuracy, and execution quality
- Introduces adversarial debate to test conviction and robustness
- Provides comprehensive options trading evaluation with Black-Scholes pricing verification

2 Related Work

2.1 Financial Benchmarks

The Finance Agent Benchmark (FAB) [Bigeard et al., 2025] introduced structured evaluation of AI agents on earnings analysis tasks. BizFinBench [Lu et al., 2025] expanded coverage to include Chinese financial markets and multi-turn reasoning. However, these benchmarks use static question sets vulnerable to data contamination.

2.2 Agent Communication Protocols

The Agent-to-Agent (A2A) protocol [A2A Protocol, 2025] standardizes communication between AI agents, enabling interoperability across different implementations. The Model Context Protocol (MCP) [MCP, 2024] provides a unified interface for agents to access external tools and data sources.

2.3 Options Pricing Models

The Black-Scholes-Merton model [Black and Scholes, 1973, Merton, 1973] remains the foundation for options pricing. Extensions include stochastic volatility models [Heston, 1993] and jump-diffusion processes [Merton, 1976].

3 System Architecture

3.1 Overview

FAB++ implements a Green Agent (evaluator) and Purple Agent (finance analyst) architecture following the A2A protocol specification. Figure 1 illustrates the system components.

```
+------------------------------------------------------------------+
|                    AgentBusters System                            |
+------------------------------------------------------------------+
|  +---------------+      A2A Protocol      +---------------+       |
|  | Green Agent   |<--------------------->| Purple Agent  |       |
|  | (Evaluator)   |                       | (Analyst)     |       |
|  | Port: 9109    |                       | Port: 9110    |       |
|  +-------+-------+                       +-------+-------+       |
|          |                                       |               |
|          |     +-----------------------------+   |               |
|          |     |       6 MCP Servers         |   |               |
|          |     | SEC EDGAR  | Yahoo Finance  |   |               |
|          |     | Sandbox    | Options Chain  |   |               |
|          |     | Trading Sim| Risk Metrics   |   |               |
|          |     +-----------------------------+   |               |
+------------------------------------------------------------------+
```

Figure 1: FAB++ System Architecture

3.2 Green Agent (Evaluator)

The Green Agent serves as the benchmark orchestrator, responsible for:

- Dynamic task generation from financial data templates
- Multi-dimensional response evaluation
- Adversarial counter-argument generation
- Alpha Score computation

3.3 Purple Agent (Finance Analyst)

The Purple Agent represents the system under test, implementing:

- Financial data retrieval via MCP servers
- LLM-powered analysis generation
- Options strategy construction
- Risk assessment and position sizing

3.4 MCP Server Infrastructure

We deploy six MCP servers providing specialized financial capabilities:

Table 1: MCP Server Specifications

Server	Port	Capabilities
SEC EDGAR	8101	10-K/10-Q filings, XBRL parsing, temporal locking
Yahoo Finance	8102	Real-time quotes, historical data, lookahead detection
Python Sandbox	8103	Secure code execution for numerical computations
Options Chain	8104	Black-Scholes pricing, Greeks calculation, IV surface
Trading Simulator	8105	Paper trading, slippage modeling, P&L tracking
Risk Metrics	8106	VaR computation, Sharpe/Sortino ratios, stress testing

4 Evaluation Methodology

4.1 Task Categories

FAB++ evaluates agents across 18 categories organized into three tiers:

4.1.1 Core Finance (6 categories)

- **Beat or Miss**: Earnings surprise detection against analyst consensus
- **Macro Analysis**: Economic trend interpretation and market impact
- **Fundamental Analysis**: Financial statement interpretation
- **Quantitative Reasoning**: Numerical calculations from financial data
- **SEC Filing Analysis**: Information extraction from regulatory documents
- **Trend Analysis**: Historical pattern recognition and forecasting

4.1.2 Options Alpha (6 categories)

- **Options Pricing**: Black-Scholes valuation and fair value assessment
- **Greeks Analysis**: Sensitivity calculations and hedging strategies
- **Strategy Construction**: Multi-leg options strategies
- **Volatility Trading**: IV rank/percentile analysis
- **P&L Attribution**: Return decomposition by Greek exposure
- **Risk Management**: VaR-based position sizing

4.1.3 Advanced (6 categories)

- **Copy Trading**: Strategy replication and signal generation
- **Race to 10M**: Capital growth optimization under constraints
- **Strategy Defense**: Adversarial robustness testing
- **Financial Data Description**: Structured data interpretation
- **Multi-turn Perception**: Context maintenance across interactions
- **Sentiment Analysis**: Market sentiment extraction

4.2 Dynamic Task Generation

Unlike static benchmarks, FAB++ generates tasks dynamically using templates populated with real financial data:

Algorithm 1 Dynamic Task Generation

Require: Template T, Financial Lake $\mathcal{F}$, Simulation Date d
1: Select ticker s from universe $\mathcal{S}$
2: Lock temporal context to date d
3: Retrieve fundamental data $F_s = \mathcal{F}(s, d)$
4: Generate ground truth G from F_s
5: Instantiate task $\tau = T(s, F_s, G, d)$
6: Compute rubric criteria R for τ
7: **return** Task (τ, G, R)

4.3 Multi-Dimensional Scoring

We evaluate responses across three primary dimensions:

4.3.1 Role Score

The Role Score combines weighted subscores:

$$\text{RoleScore} = 0.30 \cdot S_{\text{macro}} + 0.40 \cdot S_{\text{fundamental}} + 0.30 \cdot S_{\text{execution}} \tag{1}$$

where:

- S_{macro}: Macro thesis quality (semantic similarity + theme coverage)
- $S_{\text{fundamental}}$: Numerical accuracy against ground truth
- $S_{\text{execution}}$: Methodology quality and tool usage

4.3.2 Adversarial Debate

We introduce adversarial debate to test agent conviction:

Algorithm 2 Adversarial Debate Protocol

Require: Agent response A, Task τ
1: Generate counter-argument C challenging A
2: Request rebuttal R from agent
3: Evaluate conviction: maintained, weakened, or collapsed
4: Compute debate multiplier $m \in [0.8, 1.2]$
5: **return** Multiplier m

4.3.3 Alpha Score

The final Alpha Score combines all dimensions:

$$\alpha = \frac{\text{RoleScore} \times \text{DebateMultiplier}}{\ln(1 + \text{Cost}) \times (1 + \text{LookaheadPenalty})} \tag{2}$$

This formulation rewards accurate, robust responses while penalizing expensive computation and temporal violations.

5 Options Alpha Challenge

5.1 Black-Scholes Implementation

The Options Chain MCP server implements the Black-Scholes-Merton model with dividend yield:

$$d_1 = \frac{\ln(S/K) + (r - q + \sigma^2/2)T}{\sigma\sqrt{T}} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{4}$$

Call and put prices:

$$C = Se^{-qT}N(d_1) - Ke^{-rT}N(d_2) \tag{5}$$

$$P = Ke^{-rT}N(-d_2) - Se^{-qT}N(-d_1) \tag{6}$$

where S is spot price, K is strike, r is risk-free rate, q is dividend yield, σ is volatility, and T is time to expiration.

5.2 Greeks Calculation

We compute the standard Greeks for evaluation:

Table 2: Options Greeks Formulas

Greek	**Call**	**Put**
Delta (Δ)	$e^{-qT}N(d_1)$	$-e^{-qT}N(-d_1)$
Gamma (Γ)	$\frac{e^{-qT}n(d_1)}{S\sigma\sqrt{T}}$	Same as call
Theta (Θ)	$-\frac{Se^{-qT}n(d_1)\sigma}{2\sqrt{T}} - rKe^{-rT}N(d_2)$	Complex
Vega (ν)	$Se^{-qT}\sqrt{T}n(d_1)$	Same as call
Rho (ρ)	$KTe^{-rT}N(d_2)$	$-KTe^{-rT}N(-d_2)$

5.3 Options Evaluation Scoring

The Options Evaluator uses a four-dimensional scoring rubric:

$$S_{\text{options}} = 0.25 \cdot S_{\text{P\&L}} + 0.25 \cdot S_{\text{Greeks}} + 0.25 \cdot S_{\text{Strategy}} + 0.25 \cdot S_{\text{Risk}} \tag{7}$$

Table 3: Options Scoring Dimensions

Dimension	Evaluation Criteria
P&L Accuracy	Max profit/loss calculations, breakeven points, probability of profit
Greeks Accuracy	Delta, gamma, theta, vega values within 5% tolerance
Strategy Quality	Correct leg identification, strike selection rationale, structure validity
Risk Management	Position sizing, hedging strategy, exit criteria definition

6 Experiments

6.1 Experimental Setup

We evaluated a baseline Purple Agent using GPT-4o as the underlying LLM. The agent was tested on:

- Synthetic questions generated from the Financial Lake
- Public CSV finance Q&A dataset
- Options trading tasks (iron condor, volatility, Greeks, risk)

6.2 Results

Table 4: Baseline Agent Performance

Dataset	Tasks	Score	Metric
Synthetic Questions	2	60,418	Alpha Score
Public CSV	3	66.67%	Accuracy
Options (Iron Condor)	1	50.8/100	Options Score

6.2.1 Options Evaluation Breakdown

Table 5: Options Task Performance (Iron Condor on SPY)

Dimension	Score
P&L Accuracy	68.0/100
Greeks Accuracy	0.0/100
Strategy Quality	50.0/100
Risk Management	85.0/100
Final Options Score	**50.8/100**

The results reveal that while the baseline agent performs well on risk management concepts (85/100), it struggles with explicit Greeks calculations (0/100), indicating a gap between conceptual understanding and numerical precision.

6.3 Public CSV Detailed Results

Table 6: Public CSV Dataset Performance

Question Category	Correct	Score
Market Analysis (US Steel Merger)	4/4	100%
Trends (Netflix ARPU)	9/9	100%
Beat or Miss (TJX Margin)	0/2	0%
Average	13/15	**66.67%**

7 Discussion

7.1 Key Findings

1. **Conceptual vs. Computational**: Agents demonstrate strong conceptual understanding but struggle with precise numerical calculations, particularly in options pricing.
2. **Category Variance**: Performance varies significantly across categories, with fundamental analysis outperforming quantitative tasks.
3. **Temporal Awareness**: Agents occasionally exhibit lookahead bias, accessing information beyond the simulation date.
4. **Risk Management Strength**: Agents excel at qualitative risk discussions but underperform on quantitative risk metrics.

7.2 Limitations

- Ground truth for subjective tasks (macro analysis) relies on reference summaries
- Options pricing assumes Black-Scholes model validity
- Adversarial debate quality depends on counter-argument generation

7.3 Future Work

- Extend to multi-agent trading simulations
- Incorporate stochastic volatility models
- Add real-time market data integration
- Develop specialized evaluators for emerging asset classes

8 Conclusion

We presented CIO-Agent FAB++, a comprehensive benchmark for evaluating AI finance agents across 18 categories spanning fundamental analysis, options trading, and risk management. Our multi-dimensional scoring methodology, combined with adversarial debate testing, provides nuanced assessment of agent capabilities beyond simple accuracy metrics. The Options Alpha Challenge introduces rigorous quantitative evaluation using Black-Scholes pricing and Greeks verification. Experimental results demonstrate the benchmark's effectiveness in revealing agent strengths and weaknesses across diverse financial reasoning tasks.

The system is publicly available at `https://github.com/yxc20089/AgentBusters` with Docker images for immediate deployment.

Acknowledgments

We thank the AgentBeats Competition organizers at Berkeley RDI for inspiring this work. We acknowledge the contributions of the A2A Protocol and MCP communities for enabling standardized agent communication.

References

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.

Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4(1):141–183.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125–144.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343.

Bigeard, A., Nashold, L., Krishnan, R., and Wu, S. (2025). Finance Agent Benchmark: Benchmarking LLMs on Real-world Financial Research Tasks. *arXiv preprint arXiv:2508.00828*. `https://arxiv.org/abs/2508.00828`.

Lu, G., Guo, X., Zhang, R., Zhu, W., and Liu, J. (2025). BizFinBench.v2: A Unified Dual-Mode Bilingual Benchmark for Expert-Level Financial Capability Alignment. *arXiv preprint arXiv:2601.06401*. `https://arxiv.org/abs/2601.06401`.

Google and Linux Foundation (2025). Agent-to-Agent Protocol: An open protocol enabling communication and interoperability between opaque agentic applications. `https://github.com/a2aproject/A2A`.

Anthropic (2024). Model Context Protocol. `https://modelcontextprotocol.io/`.

A Alpha Score Derivation

The Alpha Score is designed to reward accurate, robust, and efficient agent responses:

$$\alpha = \frac{R \cdot D}{C \cdot P} \tag{8}$$

where:

- $R = \text{RoleScore} \in [0, 100]$
- $D = \text{DebateMultiplier} \in [0.8, 1.2]$
- $C = \ln(1 + \text{Cost})$ (logarithmic cost penalty)
- $P = 1 + \text{LookaheadPenalty}$ (temporal violation penalty)

The logarithmic cost penalty ensures diminishing returns for expensive computations, while the lookahead penalty harshly penalizes agents that access future information.

B MCP Server API Reference

B.1 Options Chain Server

Listing 1: Options Chain MCP Tools

```python
# Get options chain for a ticker
get_options_chain(ticker: str, expiration: str) -> dict

# Calculate Black-Scholes price
calculate_option_price(
    spot: float, strike: float, rate: float,
    volatility: float, time_to_expiry: float,
    option_type: str, dividend_yield: float
) -> dict  # Returns price and all Greeks

# Get implied volatility surface
get_iv_surface(ticker: str) -> dict

# Analyze multi-leg strategy
analyze_strategy(legs: list[dict]) -> dict
```

B.2 Risk Metrics Server

Listing 2: Risk Metrics MCP Tools

```python
# Calculate portfolio Greeks
calculate_portfolio_greeks(positions: list[dict]) -> dict

# Calculate Value at Risk
calculate_var(
    returns: list[float], confidence: float,
    method: str  # "historical", "parametric", "monte_carlo"
) -> dict

# Run stress test
run_stress_test(
    portfolio: dict,
    scenarios: list[dict]  # e.g., {"name": "crash", "spot_change": -0.20}
) -> dict
```

C Evaluation Configuration

Listing 3: Sample Evaluation Config (YAML)

```yaml
name: "FAB++ Full Evaluation"
datasets:
  - type: synthetic
    path: data/synthetic_questions/questions.json
    limit: 50
  - type: bizfinbench
    path: data/BizFinBench.v2
    task_types: [event_logic_reasoning, financial_quantitative_computation]
    languages: [en]
```

```yaml
    limit_per_task: 20
  - type: public_csv
    path: finance-agent/data/public.csv
    limit: 100
sampling:
  strategy: stratified
  total_limit: 100
  seed: 42
```