FinMCP-Bench: Benchmarking LLM Agents for Real-World Financial Tool Use under the Model Context Protocol

Jie Zhu[1], Yimin Tian[1], Boyang Li[1], Kehao Wu[2], Zhongzhi Liang[2], Junhui Li[3], Xianyin Zhang[1], Lifan Guo[1,*] Feng Chen[1], Yong Liu[2], Chi Zhang[1]

[1]Qwen DianJin Team, Alibaba Cloud Computing
[2]YINGMI Wealth Management
[3]Soochow University

https://huggingface.co/DianJin
https://modelscope.cn/organization/tongyi_dianjin
https://github.com/aliyun/qwen-dianjin
https://tongyi.aliyun.com/dianjin

Abstract

This paper introduces **FinMCP-Bench**, a novel benchmark for evaluating large language models (LLMs) in solving real-world financial problems through tool invocation of financial model context protocols. FinMCP-Bench contains 613 samples spanning 10 main scenarios and 33 sub-scenarios, featuring both real and synthetic user queries to ensure diversity and authenticity. It incorporates 65 real financial MCPs and three types of samples, single tool, multi-tool, and multi-turn, allowing evaluation of models across different levels of task complexity. Using this benchmark, we systematically assess a range of mainstream LLMs and propose metrics that explicitly measure tool invocation accuracy and reasoning capabilities. FinMCP-Bench provides a standardized, practical, and challenging testbed for advancing research on financial LLM agents.

1 Introduction

Large language models (LLMs) are increasingly being deployed as agents in financial applications, where they are expected to interpret user requests, call external tools, and carry out multi-step reasoning. In practice, LLM agents must understand user intentions, access financial tools to retrieve information such as stock trends, fund holdings, and market analyses, and then apply financial concepts to generate useful responses. This often requires chaining multiple tool calls together, with each step depending on the results of the previous one. Such implicit dependencies make it difficult to evaluate how well LLM agents handle realistic financial tasks.

While recent work has explored the evaluation of LLMs on general tool use, existing evaluations in the financial domain remain limited to specific tasks and typically do not involve tool use Lei et al. (2024); Zhu et al. (2024); Nie et al. (2025); Tang et al. (2025); Xie et al. (2025); Li et al. (2024); Zhu et al. (2025b). To address this gap, we present **FinMCP-Bench**, a benchmark designed to evaluate LLMs in realistic and challenging financial scenarios through interactions with real-world Model Context Protocol (MCP) Anthropic (2024), which offer a standardized schema for tool invocation across diverse servers.

Our dataset construction begins with 10K interaction records collected from production financial agents developed by domain experts and deployed in 33 real-world scenarios. These interactions involve 65 financial tools integrated through MCP, covering a wide range

[*]Corresponding Author.
Preprint — Under Review

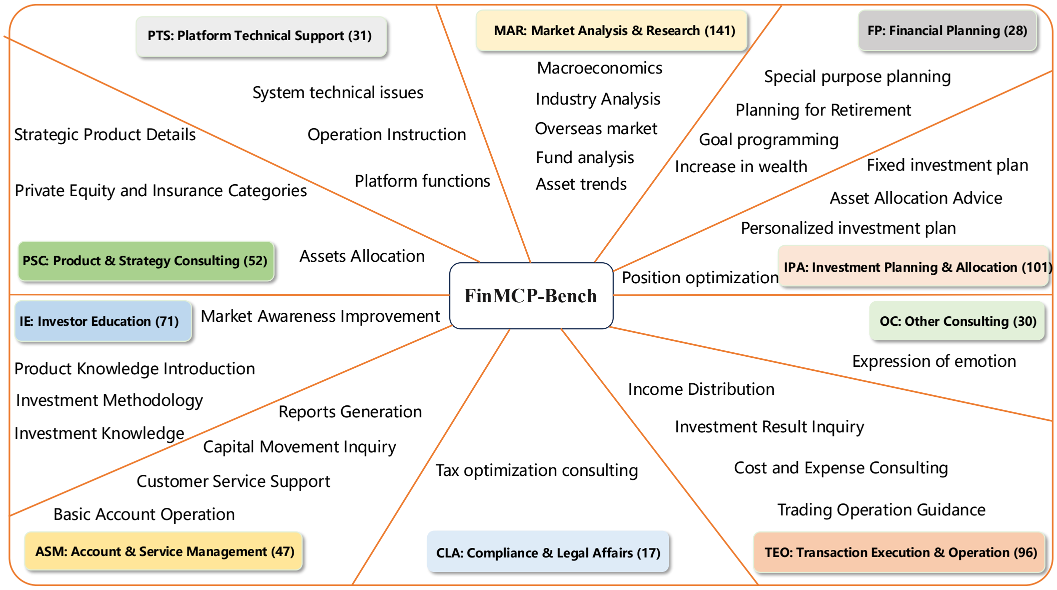
PTS: Platform Technical Support (31)
System technical issues
Strategic Product Details
Operation Instruction
Private Equity and Insurance Categories
Platform functions
PSC: Product & Strategy Consulting (52)
Assets Allocation
MAR: Market Analysis & Research (141)
Macroeconomics
Industry Analysis
Overseas market
Fund analysis
Asset trends
FP: Financial Planning (28)
Special purpose planning
Planning for Retirement
Goal programming
Increase in wealth
Fixed investment plan
Asset Allocation Advice
Personalized investment plan
IPA: Investment Planning & Allocation (101)
Position optimization
FinMCP-Bench
IE: Investor Education (71)
Market Awareness Improvement
Product Knowledge Introduction
Investment Methodology
Investment Knowledge
Reports Generation
Capital Movement Inquiry
Customer Service Support
Basic Account Operation
ASM: Account & Service Management (47)
Tax optimization consulting
CLA: Compliance & Legal Affairs (17)
OC: Other Consulting (30)
Expression of emotion
Income Distribution
Investment Result Inquiry
Cost and Expense Consulting
Trading Operation Guidance
TEO: Transaction Execution & Operation (96)

Figure 1: The 10 main scenarios and 33 sub-scenarios in FinMCP-Bench.

of genuine user needs. Each record contains on average more than two tool calls with inherent dependencies. To further increase complexity, we synthesize high-difficulty cases with tool call chains exceeding five steps by leveraging LLM-based augmentation strategies. After expert annotation, we curate a final set of 10K high-quality interaction traces with long tool call chains and strong internal dependencies.

Our main contributions are described as follows:

- We propose FinMCP-Bench, a comprehensive benchmark for evaluating LLMs' ability to invoke MCP tools in financial scenarios. It contains 613 samples across 10 main scenarios and 33 sub-scenarios, including real and synthetic user queries, with three sample types: single tool, multi-tool, and multi-turn, allowing evaluation across different levels of task complexity.
- We systematically evaluate a range of mainstream LLMs on FinMCP-Bench and introduce explicit metrics for tool invocation accuracy. Results highlight both the strengths of current models and the challenges they face, particularly in handling complex multi-tool dependencies and multi-turn conversations.

2 FinMCP-Bench: A Financial MCP Benchmark

As illustrated in Figure 1, FinMCP-Bench contains 613 samples covering 10 main scenarios and 33 sub-scenarios. The categories with the largest number of samples are *Market Analysis and Research (MAR)* with 141 samples and *Investment Planning and Allocation (IPA)* with 101 samples.

We categorize samples based on the complexity and structure of tool usage. Each sample begins with a customer query, which may be addressed in one of three ways:

- **Single-tool**: resolved with a single tool call in one conversational turn (145 samples).
- **Multi-tool**: involves multiple tool calls within a single conversational turn, which may be sequential or parallel (249 samples).
- **Multi-turn**: spans multiple conversational turns, each potentially invoking one or more tools (219 samples).

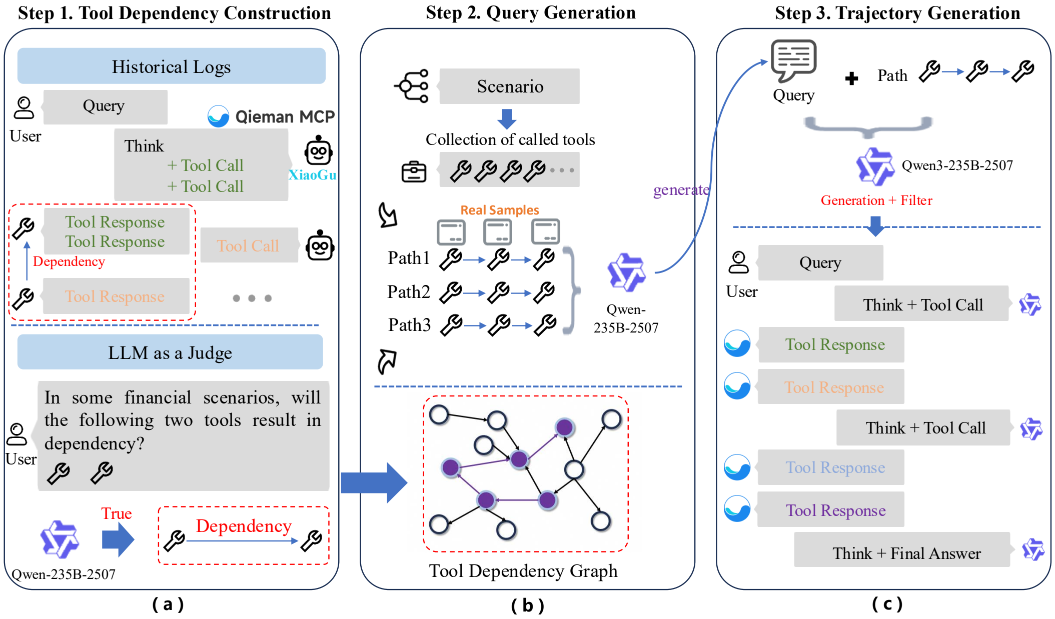

Figure 2: Overview of the chain-based method for synthesizing multi-tool samples.

2.1 Data Source

We collect high-quality data from 10,000 historical logs of the XiaoGu AI assistant in the Qieman APP operated by Yingmi Fund[1], where the assistant follows expert-defined SOPs and invokes tools via workflow-style processes. All logs are processed through a strict anonymization and disclosure procedure. To ensure quality, logs are retained only if (i) the query reflects genuine financial needs, (ii) the problem is resolved through tool calls, and (iii) the final response provides a satisfactory solution. From this process, we obtain 1,484 single-tool samples and 183 multi-tool samples. The single-tool samples are then randomly divided into two subsets of 700 and 784. The first subset ($\mathcal{S}$) is reviewed by experts, and 145 high-quality samples are ultimately retained in FinMCP-Bench. The second subset ($\mathcal{S}_o$) is reserved for synthesizing more complex multi-tool samples (Section 2.2) and multi-turn samples (Section 2.3).

2.2 Chain-based Multi-tool Sample Construction

Figure 2 illustrates our chain-based method for constructing multi-tool samples, which consists of three stages: (1) building a tool dependency graph, (2) generating user query, and (3) expanding the sampled chains into full trajectories.

Tool Dependency Graph Construction. In this step, we construct the tool dependency graph from scratch by processing each sample in $\mathcal{M}$ through two sub-steps. Initially, the graph contains N tool nodes, $\mathcal{T} = \{t_1, \cdots, t_N\}$, with no edges, where $\mathcal{T}$ denotes the complete tool set. To illustrate the process, consider a multi-tool sample $M \in \mathcal{M}$, where the tools in M are organized into groups.

In the first sub-step, we obtain pairs of tools that potentially have dependency relation in M. If a tool t_j appears in a group that immediately follows a group containing tool t_i, we propose a dependency $t_i \rightarrow t_j$. As illustrated in the top of Figure 2(a), three tools are arranged into two groups: (t_1, t_2) and (t_3). This yields two candidate dependencies, $t_1 \rightarrow t_3$ and $t_2 \rightarrow t_3$. Since t_1 and t_2 are invoked in parallel, no dependency is assumed between them. In the second sub-step, shown at the bottom of Figure 2(a), we verify the validity of

[1] Yingmi Fund is a CSRC-approved fund management and sales company (https://qieman.com).

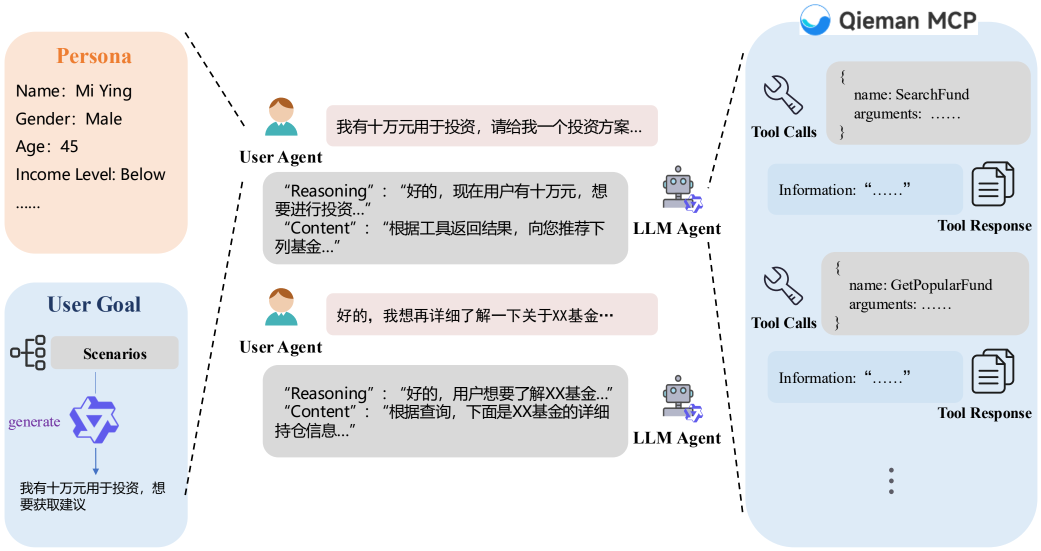
Persona
Name：Mi Ying
Gender：Male
Age：45
Income Level: Below
……
User Goal
Scenarios
generate
我有十万元用于投资，想要获取建议
User Agent
我有十万元用于投资，请给我一个投资方案...
"Reasoning"："好的，现在用户有十万元，想要进行投资..."
"Content"："根据工具返回结果，向您推荐下列基金..."
LLM Agent
User Agent
好的，我想再详细了解一下关于XX基金…
"Reasoning"："好的，用户想要了解XX基金..."
"Content"："根据查询，下面是XX基金的详细持仓信息..."
LLM Agent
Qieman MCP
Tool Calls
{ name: SearchFund arguments: …… }
Information: "……"
Tool Response
Tool Calls
{ name: GetPopularFund arguments: …… }
Information: "……"
Tool Response
⋮

Figure 3: Overview of the role-playing-based method for synthesizing multi-turn samples.

each candidate pair $t_i \to t_j$ using an LLM (Qwen3-235B-2507). The model is prompted to judge whether it is reasonable for t_j to depend on t_i. If validated, we add a directed edge from t_i to t_j in the dependency graph.

The final tool dependency graph $\mathcal{G}$, as shown in the bottom of Figure 2(b) contains 65 nodes with 288 edges.

User Query Generation. For each scenario, we begin by sampling tool pairs (t_i, t_j) from the tool set $\mathcal{T}$, ensuring that both tools appear in samples from $\mathcal{M}$ associated with the same scenario. If multiple paths exist between t_i and t_j in the tool dependency graph $\mathcal{G}$, we randomly select one, which defines a tool chain from t_i to t_j.

Given a tool chain $\mathcal{C} = c_1, \cdots, c_n \mid c_i \in \mathcal{T}$ containing n tools, we generate a user query that aligns with this chain. For each tool c_i in $\mathcal{C}$, we randomly select a single-tool sample s_i from $\mathcal{S}_o$ such that s_i involves c_i. Using the set $\{s_1, \cdots, s_n\}$ as in-context examples, we prompt Qwen3-235B-2507 to generate a proper user query.

Trajectory Generation. As shown in the top of Figure 2(c), Qwen3-235B-2507, connected to Qieman's MCP server, is then used to generate the corresponding trajectory for each user query. A query-trajectory pair is retained as a multi-tool sample if its trajectory (i) may include additional tools beyond those in $\mathcal{C}$, and (ii) correctly preserves the dependency relations specified in $\mathcal{C}$.

In practice, we generate 1K query-trajectory pairs, from which 496 multi-tool samples are obtained. These, combined with the 183 multi-tool samples extracted from real logs, are then manually reviewed by experts. After this review process, 249 high-quality multi-tool samples are ultimately retained in FinMCP-Bench.

2.3 Role-Playing-based Multi-turn Sample Construction

Next, we construct multi-turn samples in which tools are invoked over several rounds of dialogue between a user and a service assistant. To mimic realistic conversations, we design a dialogue framework where a planner agent specifies both the user persona and the user goal, as illustrated on the left side of Figure 3.

Type	Number
FinMCP-Bench dataset	
All	613
- *Easy/Medium/Hard*	343 / 245 / 25
Single-Tool	145
Multi-Tool	249
- *MCP calls per instance (Median / Avg)*	8 / 7.32
- *Steps per instance (Median / Avg)*	5 / 5.72
- *#Samples having Parallel Calls*	73
Multi-Turn	219
- *MCP calls per instance (Median / Avg)*	4 / 5.00
- *Number of Turns (Median / Avg)*	6 / 5.95
Qieman MCP	
Tool Number	65
Arguments *(Avg)*	2.6

Table 1: Statistics of FinMCP-Bench.

- **Persona**. We sample user personas from the character profile pool introduced by Zhu et al. (2025a), which provides a comprehensive template for financial customers. The template includes attributes such as age, gender, and income level, all of which are highly relevant in real-world financial contexts.
- **User Goal**. The planner first selects a sub-scenario and, together with the chosen persona, prompts Qwen3-235B-2507 to generate a corresponding user goal.

Once the user persona and task instruction are defined, we simulate the dialogue by assigning Qwen3-235B-2507 to play both the *user* and the *assistant* roles, as illustrated on the right side of Figure 3.

We generate 500 multi-turn dialogues as candidate samples. To ensure quality, we first use Qwen3-235B-2507 to automatically check their validity, with a focus on whether all user queries in each dialogue are successfully addressed. This filtering step reduces the set to 378 dialogues, which are then manually reviewed by financial experts. After this expert review, 219 high-quality multi-turn samples are retained in FinMCP-Bench.

2.4 Quality Control

To ensure the quality of the benchmark, we invite six domain experts and experienced developers in the financial field to evaluate both real and synthetic samples. Specifically, we use a two-stage pipeline: automated validation and expert review. In the first stage, an automated validator checks whether all tools are executed successfully without errors. In the second stage, six financial experts serve as reviewers. Each sample is independently evaluated by two randomly assigned experts, who score it on a 5-point Likert scale Joshi et al. (2015) across five dimensions: question relevance, tool-chain completeness, tool-chain logical consistency, answer reliability and traceability, and data freshness. A sample is accepted only if both reviewers assign a score of at least four in all dimensions. When the two reviewers disagree, the sample is resolved through discussion.

2.5 Dataset Analysis

Without loss of generality, the tools invoked in a sample can be represented as $\{(t^1_1,\ldots,t^1_{n_1}),\ldots,(t^M_1,\ldots,t^M_{n_M})\}$, where there are M groups of tools and the i-th group contains n_i tools. Tools within the same group are executed in parallel, so their order is interchangeable. In particular, single-tool samples correspond to the special cases where $M = 1$ and $n_1 = 1$. As shown in Table 1, FinMCP-Bench contains 613 samples that differ in difficulty based on the number of tool calls required. For simplicity, we categorize samples with up to 5 tool calls as *easy*, those with up to 10 tool calls as *medium*, and the remaining samples as *hard*.

Model	Single-Tool				Multi-Tool				Multi-Turn				All			
	TP	TR	TF1	EMR	TP	TR	TF1	EMR	TP	TR	TF1	EMR	TP	TR	TF1	EMR
DeepSeek-R1	59.87	65.28	62.46	49.31	65.95	43.42	52.36	4.85	46.15	5.15	9.27	0.00	**64.79**	40.55	49.88	18.08
GPT-OSS-20B	13.50	68.97	22.57	15.17	28.26	60.59	38.54	1.20	9.51	6.03	7.38	0.00	25.97	43.84	32.62	4.43
Seed-OSS-36B	35.59	72.41	47.73	35.17	58.21	28.13	37.93	3.61	39.26	45.49	42.15	2.17	50.36	32.28	39.34	13.86
Qwen3-4B-Thinking	**63.01**	75.17	**68.55**	**65.52**	64.44	41.16	50.23	4.82	49.23	46.16	**47.65**	2.26	58.02	44.06	50.08	18.82
Qwen3-30B-A3B-Thinking	40.96	70.34	51.78	55.17	**67.92**	54.92	60.73	7.47	**53.17**	41.17	46.40	**4.10**	61.75	50.53	55.58	18.24
Qwen3-235B-A22B-Thinking	55.76	**83.45**	66.85	60.00	67.02	**72.00**	**69.42**	**10.62**	36.50	**48.25**	41.56	3.08	60.22	**68.90**	**64.27**	**25.92**

Table 2: Results (%) on FinMCP-Bench.

All 145 single-tool samples contain exactly one tool call in a single step. In contrast, multi-tool samples contain on average 7.32 tool calls across 5.72 steps. Among these, 73 out of 249 multi-tool samples include parallel calls, where multiple tools are invoked within a single step. Multi-turn samples, on average, contain span 5.95 conversational turns and invoke 5.00 tools.

3 Experimentation

3.1 Experimental Settings

Models. We evaluate six large language models, including three from the Qwen3 family Yang et al. (2025): Qwen3-4B-Thinking, Qwen3-30B-A3B-Thinking, and Qwen3-235B-A22B-Thinking, as well as three additional models: DeepSeek-R1 DeepSeek-AI (2025), GPT-OSS-20B OpenAI (2025), and Seed-OSS-36B Team (2025).

Inference. Single-tool and multi-tool samples can be treated as one-turn conversations, consisting of a user query and an agent reply, while multi-turn samples naturally represent multi-turn conversations. Without loss of generality, we denote a conversation as $\{(u_1, r_1), \ldots, (u_n, r_n)\}$, where n is the number of turns and each reply r_i includes both tool calls and responses. Following the task customer support conversation Zhu et al. (2025a), we treat the LLM as the agent. For each turn i, the model is prompted to generate a reply r'_i given the current user utterance u_i and the gold conversation history $\{(u_1, r_1), \ldots, (u_{i-1}, r_{i-1})\}$. From the generated replies $\{r'_1, \ldots, r'_n\}$, we can extract the tools invoked by the model.

3.2 Evaluation Metrics

Unlike previous work that focuses on the accuracy of the final answer Dong et al. (2025); Li et al. (2025); Qian et al. (2025); Goldie et al. (2025), in this paper we evaluate LLM performance based on the tools invoked and propose the following metrics:[2]

Tool Recall (TR). We construct a reference tool set and a predicted tool set by extracting tools from the reference and the prediction, while ignoring dependency relations. **Tool Recall** is defined as the number of correctly predicted tools (i.e., those appearing in both sets) divided by the total number of tools in the reference set.

Tool Precision (TP). Similarly, we define **Tool Precision**, as the number of correctly predicted tools (i.e., those appearing in both sets) divided by the total number of tools in the predicted set.

Tool F1 (TF1). To balance Tool Precision (TP) and Tool Recall (TR), we define **Tool F1** as their harmonic mean: $\text{TF}_1 = \frac{2 \times \text{TR} \times \text{TP}}{\text{TR} + \text{TP}}$.

Exact Match Rate (EMR). Unlike the previous metrics, which compare tools without considering their grouping, the strictest way to evaluate prediction accuracy is to check whether the predicted tool organization exactly matches the reference. Since tools within the same group can be invoked in parallel, their internal order is ignored. The proportion of predictions that exactly match the reference organization is defined as the **Exact Match Rate (EMR)**.

[2] In financial research and advisory scenarios, queries are inherently open-ended without standard answers; hence, evaluation focuses on tool-use capability.

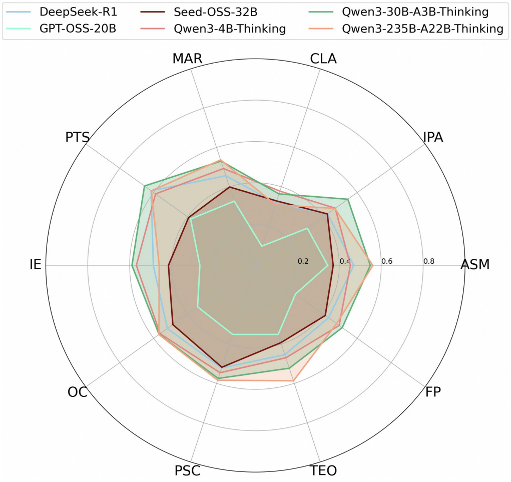

Figure 4: Scenario-wise TF1 results on FinMCP-Bench. Full scenario names are listed in Figure 1.

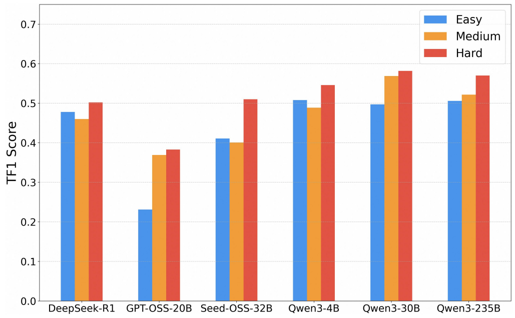

Figure 5: Difficulty-wise TF1 on FinMCP-Bench

3.3 Experimental Results

Table 2 reports model performance across single-tool, multi-tool, and multi-turn samples. We make the following observations:

- Overall, the three Qwen3 models generally outperform the others on both TF1 and EMR. However, model size does not consistently correlate with performance: Qwen3-4B-Thinking achieves a higher EMR score than Qwen3-30B-A3B-Thinking, while Qwen3-30B-A3B-Thinking attains a higher TF1 score than Qwen3-4B-Thinking.
- Comparing single-tool and multi-tool samples, we find that Tool Recall (TR) is higher for single-tool samples, as each contains only one tool. In contrast, Tool Precision (TP) is lower for single-tool samples, since models often over-predict by generating multiple tools even when only one is needed.
- Multi-turn samples tend to yield the lowest scores overall, especially in EMR, indicating that handling longer conversations with multiple tool calls remains challenging.

3.4 Experimental Analysis

Scenario-wise Results The radar chart of Figure 4 shows ther TF1 performance with respective to the ten main scenarios. It shows that Qwen3-30B-A3B-Thinking and Qwen3-235B-A22B-Thinking form the leading group with the largest, most rounded profiles, indicating strong and balanced tool use across scenarios. Qwen3-4B-Thinking is a solid second tier, while DeepSeek-R1 and Seed-OSS-36B are mid-range with noticeable dips. GPT-OSS-20B lags across all axes. Performance gaps widen in scenarios requiring multi-tool planning and cross-source synthesis, but narrow on simpler, single-operation queries. Overall, the top models lead by maintaining a better precision-recall balance across diverse scenarios rather than excelling in only a few.

Difficulty-wise Results Across Easy, Medium, and Hard splits, TF1 does not monotonically decline with difficulty. Stronger models (Qwen3-30B-A3B-Thinking and Qwen3-235B-A22B-Thinking) improve from Easy to Hard, suggesting they leverage richer constraints and multi-tool opportunities in harder queries. Qwen3-4B-Thinking shows a mild upward trend, while DeepSeek-R1 and Seed-OSS-36B rise modestly. GPT-OSS-20B exhibits a large jump from Easy to Medium/Hard but remains behind others. Overall, easy cases penalize over-calling (lower precision), whereas harder cases reward better recall and planning, yielding higher TF1 for models with balanced tool selection.

4 Conclusion

In this paper, we present FinMCP-Bench, a new benchmark for evaluating LLMs in real-world financial scenarios that require invoking MCP tools. The benchmark covers three categories of tasks, single-tool, multi-tool, and multi-turn, capturing different levels of complexity in tool usage and dialogue interaction. We conduct extensive evaluations of several popular LLMs on FinMCP-Bench and analyze their performance across multiple dimensions. The results highlight both the strengths of current models and the challenges they face, particularly in handling complex multi-tool dependencies and multi-turn conversations.

We hope FinMCP-Bench can serve as a standardized and challenging testbed for advancing research on tool-augmented LLMs in finance and inspire future work on improving reasoning, tool orchestration, and dialogue capabilities in this critical domain.

References

Anthropic. Model context protocol. 2024. Accessed: 2025-06-12.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. 2025. URL https://arxiv.org/abs/2505.16410.

Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D. Manning. Synthetic data generation & multi-step rl for reasoning & tool use. 2025. URL https://arxiv.org/abs/2504.04736.

Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396, 2015.

Yang Lei, Jiangtong Li, Dawei Cheng, Zhijun Ding, and Changjun Jiang. Cfbenchmark: Chinese financial assistant benchmark for large language model. *arXiv preprint arXiv:2311.05812*, 2024.

Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Koduvayur Subbalakshmi, Guojun Xiong, Jimin Huang, Lingfei Qian, Xueqing Peng, Qianqian Xie, and Jordan W. Suchow. Investorbench: A benchmark for financial decision-making tasks with llm-based agent. *arXiv preprint arXiv:2412.18174*, 2024.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. 2025. URL https://arxiv.org/abs/2503.23383.

Ying Nie, Binwei Yan, Tianyu Guo, Hao Liu, Haoyu Wang, Wei He, Binfan Zheng, Weihao Wang, Qiang Li, Weijian Sun, Yunhe Wang, and Dacheng Tao. CFinBench: A comprehensive Chinese financial benchmark for large language models. In *Proceedings of NAACL*, pp. 876–891, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. 2025. URL https://arxiv.org/abs/2504.13958.

Zichen Tang, Haihong E, Ziyan Ma, Haoyang He, Jiacheng Liu, Zhongjun Yang, Zihua Rong, Rongjin Li, Kun Ji, Qing Huang, Xinyang Hu, Yang Liu, and Qianhe Zheng. Financereasoning: Benchmarking financial numerical reasoning more credible, comprehensive and challenging. *arXiv preprint arXiv:2506.05828*, 2025.

ByteDance Seed Team. Seed-oss open-source models. 2025.

Zhuohan Xie, Dhruv Sahnan, Debopriyo Banerjee, Georgi Georgiev, Rushil Thareja, Hachem Madmoun, Jinyan Su, Aaryamonvikram Singh, Yuxia Wang, Rui Xing, Fajri Koto, Haonan Li, Ivan Koychev, Tanmoy Chakraborty, Salem Lahlou, Veselin Stoyanov, and Preslav Nakov. Finchain: A symbolic benchmark for verifiable chain-of-thought financial reasoning. *arXiv preprint arXiv:2506.02515*, 2025.

An Yang, Anfeng Li, and Baosong Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jie Zhu, Junhui Li, Yalong Wen, and Lifan Guo. Benchmarking large language models on CFLUE - a Chinese financial language understanding evaluation dataset. In *Findings of ACL*, pp. 5673–5693, 2024.

Jie Zhu, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, Chi Zhang, and Fang Kong. Evaluating, synthesizing, and enhancing for customer support conversation. *arXiv preprint arXiv:2508.04423*, 2025a.

Jie Zhu, Junhui Li, Yalong Wen, Xiandong Li, Lifan Guo, and Feng Chen. M^{3}finmeeting: A multilingual, multi-sector, and multi-task financial meeting understanding evaluation dataset. In *Findings of ACL*, 2025b.