Agent Team Result Analysis

Agent-based results

• ***Proprietary agent:*** This agent achieves the first place on the agent scoreboard for every model variation and the second place on the human scoreboard with Sonnet-4.5 with a score of 4900. This score represents a 70.59% success rate. Looking at the proprietary agent configurations, there are small differences in solving performance between Sonnet-4.5 and Opus-4.1, but a large drop when using Haiku-3.5. We note a large gap in performance between the proprietary agent and the CTF specific agents. According to the developers, their agent is prompted to encourage long term planning and they do not provide the agent with custom tool wrappers, but rather allow it to run various tools interactively. This gives the agent flexibility as needed and resolves its own issues when they occur. Further, they believe that LLMs are more familiar with direct tool usages rather than custom function calls due to the baseline training data.

• ***Claude Code:*** Claude Code achieves the third place on the agent scoreboard for Sonnet-4.5 with a score of 4600, sharing the same place with one human team. Claude Code also significantly outperforms agents built specifically for solving CTFs, except the proprietary agent. As with the proprietary agent, there is a dramatic drop in performance when using Haiku-3.5. With this model choice, all of the agents perform similarly poor. Therefore, while framework choice influences the agent's thinking patterns and tooling capabilities, model choice is still important in shaping the baseline reasoning capabilities.

• ***NYU Autonomous Framework:*** The NYU Agent exhibits inconsistent performance across different models choices. As shown in Figure 4 in the paper, it trails behind both proprietary agent and Claude Code on Sonnet-4.5 and Opus-4.1, with 4300 points as the best. And it matches both agents on Haiku-3.5 (all for 1100 points). The jumps in performance across model choices suggests that the NYU Agent is dependent on advanced LLM reasoning capabilities to navigate its custom tooling layer.

• ***Cybench LM Agent:*** By contrast, Cybench ranked lowest across all model configurations, with scores of 900, 600, and 300 points respectively. Its performance drop, particularly under weaker models, underscoring its limited capability of leveraging the advanced reasoning of the model. Moreover, the consistently low success rates (more details in Table 2 in paper) indicate that Cybench may lack adaptive mechanisms to compensate for model limitations.

Difficulty-based results

Model	Agent	Overall Success Rate	Easy Challenges			Medium Challenges			Hard Challenges		
			Success Rate	Time	Token	Success Rate	Time	Token	Success Rate	Time	Token
Sonnet-4.5	Proprietary Agent	12/17 (70.59%)	8/9 (88.89%)	32.78	2.35	3/6 (50.00%)	69.85	6.33	1/2 (50.00%)	83.43	9.16
	Claude Code	11/17 (64.71%)	7/9 (77.78%)	16.05	2.44	3/6 (50.00%)	20.67	5.22	1/2 (50.00%)	22.00	6.69
	NYU Agent	7/17 (41.18%)	5/9 (55.56%)	14.97	3.04	1/6 (16.67%)	32.48	7.90	1/2 (50.00%)	33.64	11.78
	Cybench	3/17 (17.65%)	3/9 (33.33%)	13.10	2.16	0/6 (0%)	15.35	1.97	0/2 (0%)	31.82	1.17
Opus-4.1	Proprietary Agent	10/17 (58.82%)	6/9 (66.67%)	48.46	20.19	3/6 (50.00%)	65.88	23.57	1/2 (50.00%)	69.9	25.50
	Claude Code	8/17 (47.06%)	5/9 (55.56%)	20.40	10.72	2/6 (33.33%)	33.50	15.47	1/2 (50.00%)	48.50	15.10
	NYU Agent	4/17 (23.53%)	3/9 (33.33%)	41.14	22.23	1/6 (16.67%)	21.99	10.37	0/2 (0%)	52.00	41.42
	Cybench	2/17 (11.76%)	2/9 (22.22%)	14.15	5.74	0/6 (0%)	24.34	10.85	0/2 (0%)	29.90	6.68
Haiku-3.5	Proprietary Agent	3/17 (17.65%)	2/9 (22.22%)	201.74	7.23	1/6 (16.67%)	195.57	7.54	1/2 (50.00%)	216.97	9.00
	Claude Code	3/17 (17.65%)	2/9 (22.22%)	13.80	0.44	1/6 (16.67%)	13.80	0.67	0/2 (0%)	63.00	0.65
	NYU Agent	3/17 (17.65%)	2/9 (22.22%)	111.31	1.69	1/6 (16.67%)	106.20	2.50	0/2 (0%)	71.93	2.83
	Cybench	1/17 (5.88%)	1/9 (11.11%)	17.44	0.30	0/6 (0%)	32.00	0.47	0/2 (0%)	24.84	0.39

Table1: Performance comparison across models, agents, and **difficulty levels**. Time represents the average time spent for one challenge in minutes; Token denotes the average token usage for one challenge in dollars.

Table1 below illustrates the performance and cost of agents for challenges in difficulty levels.
• ***Easy Challenges:*** In the easy tier, proprietary agent with Sonnet-4.5 achieved the highest success rate (88.89%), outperforming other agent-model combinations. It also maintained relatively low token cost (avg. 2.35 dollars per challenge). For Claude Code, although its success rate on this level was slightly lower than that of proprietary agent, it spent only a half time and very close tokens to solving these challenges, demonstrating both efficiency and effectiveness. The NYU Autonomous Framework under Sonnet showed decent performance (solved 5 out of 9 challenges) but consumed more tokens and time. For Cybench, despite the cost saving,it only solved one third challenges on easy challenges set, which is not ideal for a CTF-oriented agent. For Opus-4.1, the success rate of all agents dropped to the same extent (by 22%), while token usage significantly increased. Besides the factor of per-token pricing, it still shows the low time- and token- effectiveness. As mentioned in overall result in Table1, Haiku-3.5 performed poorly across all agents, with all agents' success rate decreased to 22%, except Cybench agent, suggesting that smaller models may offer cost benefits but at the expense of performance.

• ***Medium Challenges:*** As challenge complexity increased, overall success rates declined, with both the proprietary agent and Claude Code (Sonnet-4.5) remaining the top success rate at 50%, however, the proprietary agent required much more time on them. Other agents struggled, with NYU agent solving only one challenge (the same challenge) across all models and Cybench failing to solve any tasks. This suggests that larger models offer necessary reasoning capabilities for medium challenges, and well-designed agents like Claude Code can further enhance performance with modest time and cost overhead.

• ***Hard Challenges:*** At the highest difficulty, a half agent and model combination achieved some success (solving one challenge), while all other combinations failed. Claude Code (Sonnet-4.5)

maintained a competitive balance (with avg. 22 minutes and avg. 6.69 dollars per challenge), while the proprietary agent consistently asked for more token usage and longer time. Other agents underperformed significantly: NYU Framework could only see some progress when using Sonnet-4.5, and Cybench failed again. In summary, only the strong models and agents can handle difficult CTF challenges, and that inefficient model or agent limitations can lead to high cost without any benefit.