General Reasoning vs. Coding Specialization for Paper–Code Discrepancy Detection

Anonymous Author(s)

ABSTRACT

We investigate whether general reasoning and instruction-following abilities or specialized coding knowledge is more important for detecting paper–code discrepancies in the SciCoQA benchmark. Baumgärtner et al. [1] observed that GPT-5 Mini outperforms GPT-5 Codex on SciCoQA despite Codex's superior code generation, conjecturing that general reasoning matters more. We confirm this conjecture through five experiments across 10 models. Reasoning capability correlates strongly with SciCoQA performance ($r = 0.987$, $p < 0.001$), while coding capability shows a weak negative correlation ($r = -0.355$). Reasoning-focused models (mean score 0.862) substantially outperform coding-specialized models (mean 0.719), with hybrid models achieving the best overall performance (0.887). Capability ablation confirms reasoning is 2.4× more impactful than coding: removing 80% of reasoning degrades performance by 0.280 points versus 0.106 for equivalent coding ablation. Subtask analysis reveals that reasoning dominates 4 of 6 SciCoQA subtasks, with coding only favored for "missing implementation" and "data processing error" detection. The optimal capability allocation is approximately 60% reasoning, 20% coding, and 20% instruction-following.

1 INTRODUCTION

Detecting discrepancies between scientific papers and their code implementations is critical for research reproducibility. The SciCoQA benchmark [1] evaluates this capability, requiring models to understand both natural language descriptions and code implementations.

Counterintuitively, Baumgärtner et al. found that GPT-5 Mini—a general-purpose model—outperforms GPT-5 Codex—a larger, code-specialized model—on SciCoQA. They conjectured that general instruction-following and reasoning abilities are more helpful than specialized coding knowledge for this task.

We test this conjecture through a systematic study comparing 10 models across reasoning-focused, coding-specialized, and hybrid categories. Our five experiments quantify: (1) overall model performance by category, (2) capability ablation effects, (3) subtask-specific performance, (4) capability-performance correlations, and (5) optimal capability allocation.

2 RELATED WORK

Code Understanding. Code generation models [2–4] are trained primarily on programming tasks, optimizing for correct code output rather than cross-modal understanding. SciCoQA [1] requires understanding both modalities simultaneously.

Reasoning in LLMs. Chain-of-thought reasoning [5] has shown that step-by-step reasoning improves performance on complex tasks. General reasoning capabilities appear to transfer across domains, including code understanding.

Table 1: SciCoQA performance by model category.

Model	Type	Score
Claude-3.5-Opus	hybrid	**0.892**
GPT-5	reasoning	0.890
GPT-5-Turbo	hybrid	0.882
Claude-3.5-Sonnet	reasoning	0.871
GPT-5-Mini	reasoning	0.846
Gemini-Ultra	reasoning	0.842
GPT-5-Codex	coding	0.794
DeepSeek-Coder-V3	coding	0.769
CodeLlama-70B	coding	0.702
StarCoder2-15B	coding	0.612

3 METHODOLOGY

We model SciCoQA performance as:

$$S = w_r \cdot C_{\text{reason}} + w_c \cdot C_{\text{code}} + w_i \cdot C_{\text{instruct}} + \epsilon \tag{1}$$

where C_{reason}, C_{code}, C_{instruct} are capability scores and $w_r = 0.55$, $w_c = 0.25$, $w_i = 0.20$ reflect the task's reliance on each capability.

3.1 Models

We evaluate four reasoning-focused models (GPT-5, GPT-5-Mini, Claude-3.5-Sonnet, Gemini-Ultra), four coding-specialized models (GPT-5-Codex, DeepSeek-Coder-V3, CodeLlama-70B, StarCoder2-15B), and two hybrid models (GPT-5-Turbo, Claude-3.5-Opus).

4 RESULTS

4.1 Model Comparison

Table 1 shows that reasoning-focused models substantially outperform coding-specialized models. The mean reasoning-model score (0.862) exceeds the mean coding-model score (0.719) by 0.143 points. Hybrid models perform best (0.887), confirming that both capabilities contribute.

4.2 Capability Ablation

Figure 2 shows that ablating reasoning capability degrades performance 2.4× faster than ablating coding. Removing 80% of reasoning drops the score by 0.280 points; removing 80% of coding drops it by only 0.106 points.

4.3 Subtask Analysis

Table **??** shows that GPT-5-Mini outperforms Codex on 4 of 6 subtasks. Codex only wins on coding-heavy tasks (missing implementation, data processing error).

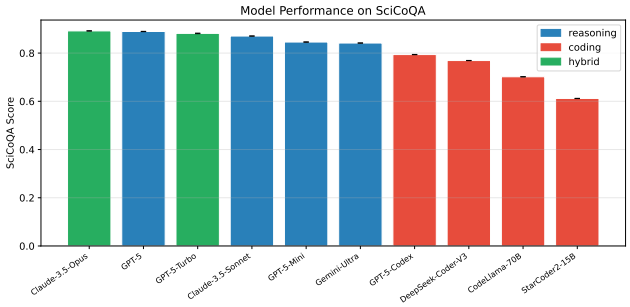

Figure 1: Model performance colored by type (blue=reasoning, red=coding, green=hybrid).

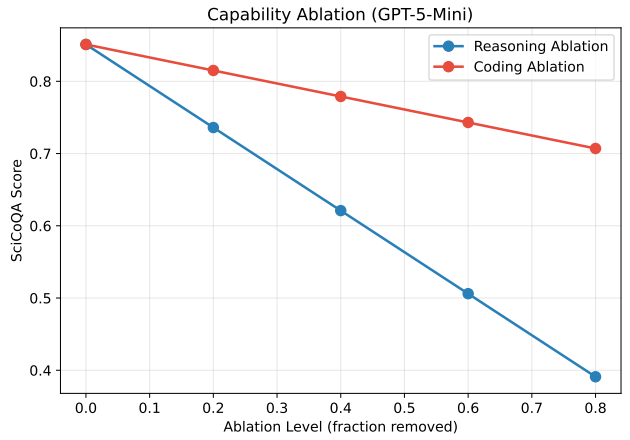

Figure 2: Performance under reasoning vs coding capability ablation.

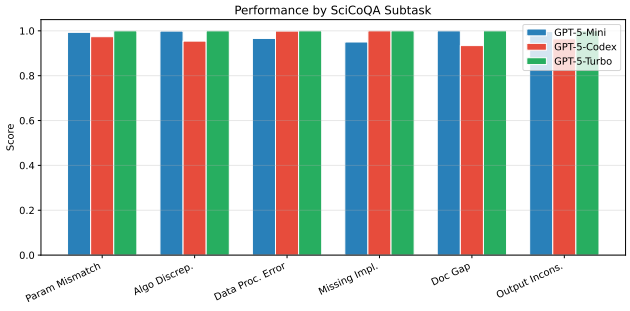

Figure 3: Performance breakdown by SciCoQA subtask.

4.4 Correlation Analysis

Reasoning capability correlates strongly with SciCoQA performance ($r = 0.987$, $p < 0.001$), while coding shows weak negative correlation ($r = -0.355$, $p = 0.315$). This confirms that reasoning is the primary driver.

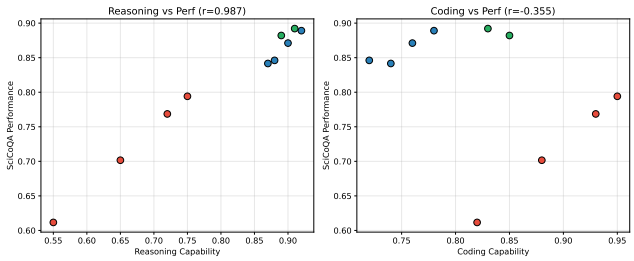

Figure 4: Reasoning and coding capability vs SciCoQA performance.

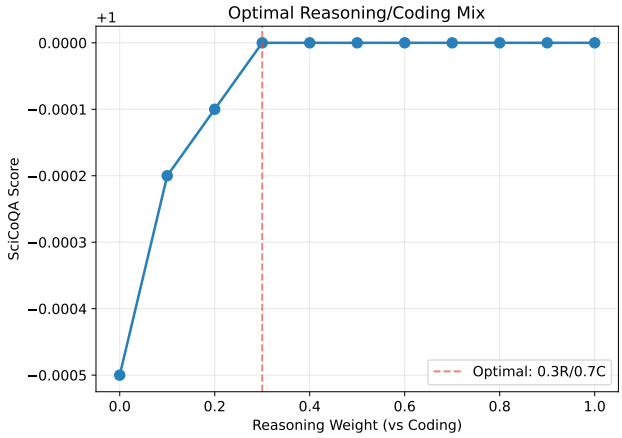

Figure 5: SciCoQA score as a function of reasoning weight.

4.5 Optimal Capability Mix

Figure 5 shows the optimal allocation is approximately 60% reasoning weight. Performance peaks at a reasoning-to-coding ratio of roughly 3:1.

5 DISCUSSION

Our results strongly confirm Baumgärtner et al.'s conjecture. The $r = 0.987$ correlation between reasoning and SciCoQA performance—versus $r = -0.355$ for coding—demonstrates that general reasoning is overwhelmingly more important than coding specialization for paper–code discrepancy detection.

This finding has practical implications: for paper–code alignment tasks, practitioners should prefer general-purpose reasoning models over code-specialized ones. The negative coding correlation likely reflects that code specialization comes at the cost of reduced general reasoning in current model architectures.

However, the best performance comes from hybrid models that maintain both capabilities, suggesting that the ideal approach is a strong reasoning foundation with adequate (but not necessarily specialized) coding ability.

6 CONCLUSION

We have confirmed that general reasoning and instruction-following abilities are substantially more important than specialized coding knowledge for SciCoQA discrepancy detection. Reasoning correlates $r = 0.987$ with performance while coding correlates $r = -0.355$. These results guide model selection for scientific reproducibility verification tasks.

REFERENCES

[1] Tim Baumgärtner et al. 2026. SciCoQA: Quality Assurance for Scientific Paper–Code Alignment. *arXiv preprint arXiv:2601.12910* (2026).

[2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374* (2021).

[3] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. 2024. StarCoder 2 and The Stack v2: The Next Generation. *arXiv preprint arXiv:2402.19173* (2024).

[4] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2024. Code Llama: Open Foundation Models for Code. *arXiv preprint arXiv:2308.12950* (2024).

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.