Evaluating AGENTS.md: Are Repository-Level Context Files Helpful for Coding Agents?

Paper: 2602.11988

Authors: Thibaud Gloaguen, Niels Mündler, Mark Müller, Veselin Raychev, Martin Vechev

Published: February 2026

Summary

This paper evaluates whether repository-level context files (like) actually help coding agents perform better on real-world software engineering tasks.

Key Findings

Performance Impact

- **LLM-generated context files**: Decrease task success rates by ~3% on average
- **Developer-provided context files**: Marginally improve performance by ~4% on average
- **No context files**: Baseline performance

Cost Impact

- Context files increase inference costs by **over 20%** on average
- More steps required to complete tasks (2.45-3.92 additional steps)

Behavioral Changes

- **More testing and exploration**: Agents run more tests, search more files, read more files

- **Instruction following**: Agents generally follow instructions in context files
- **Redundant documentation**: Context files are often redundant with existing documentation
- **No effective overviews**: Context files don't provide useful repository overviews

AGENTBENCH

The authors created a new benchmark called **AGENTBENCH** consisting of:

- **138 unique instances** from 12 repositories
- Real GitHub issues (bug-fixing and feature addition tasks)
- Developer-written context files
- Python software engineering tasks

AGENTBENCH complements SWE-BENCH LITE (which uses popular repositories without context files).

Experimental Setup

Coding Agents Evaluated

- **CLAUDE CODE** with SONNET-4.5
- **CODEX** with GPT-5.2 and GPT-5.1 MINI
- **QWEN CODE** with QWEN3-30B-CODER

Datasets

- **SWE-BENCH LITE**: 300 tasks from 11 popular Python repositories (no context files)
- **AGENTBENCH**: 138 tasks from 12 repositories with developer-provided context files

Settings Evaluated

1. **NONE**: No context files
2. **LLM**: LLM-generated context files (using agent-developer recommendations)
3. **HUM**: Developer-provided context files

Key Insights

1. Context Files Make Tasks Harder

Instructions in context files increase reasoning tokens by 14-22%, suggesting tasks become more complex.

2. Context Files Are Redundant

When documentation files are removed from repositories, LLM-generated context files actually improve performance by 2.7% on average.

3. Stronger Models Don't Generate Better Context Files

Using GPT-5.2 to generate context files improves SWE-BENCH LITE performance by 2% but degrades AGENTBENCH performance by 3%.

4. Context Files Encourage Exploration

Agents use more repository-specific tools (e.g., ,) and run more tests when context files are present.

Recommendations

1. **Omit LLM-generated context files** for now, contrary to agent-developer recommendations
2. **Include only minimal requirements** in context files (e.g., specific tooling to use)
3. **Human-written context files** should describe only essential information
4. **Future work**: Improve automatic generation of concise, task-relevant guidance

Limitations

- Evaluation focused heavily on Python (a language well-represented in training data)
- Context files are a recent development (August 2025)
- Popular repositories used in benchmarks may not be representative of most codebases

Related Work

- **SWE-BENCH**: Repository-level coding agent evaluation
- **AGENTBENCH**: New benchmark for context file evaluation
- **Context files**: AGENTS.md, CLAUDE.md, README files for agents

Conclusion

Context files have only a **marginal effect** on coding agent behavior. While they encourage broader exploration and instruction following, they don't provide effective repository overviews and often make tasks harder. The authors recommend omitting LLM-generated context files and including only minimal requirements in human-written ones.

Tags: #agents #context-files #evaluation #SWE-bench #LLM-agents

---

Revision #7
Created 3 March 2026 09:24:52 by Clive
Updated 3 March 2026 09:59:42 by James