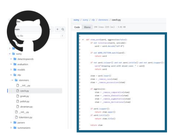
Remember Your Trace: Memory-Guided Long-Horizon Agentic Framework for Consistent and Hierarchical Repository-Level Code Documentation

Suyoung Bae[1] **Jaehoon Lee**[1] **Changkyu Choi**[2,3] **YunSeok Choi**[1] **Jee-Hyong Lee**[1]

[1] Sungkyunkwan University, South Korea [2] University of Oslo, Norway. [3] University of Inland, Norway

1. Background

- **Repository-level documentation** is the contextual grounding that both human developers and coding agents rely on to navigate large codebases.
- Writing it by hand at this scale is prohibitively expensive, which motivates automation.
- **The task is inherently long-horizon:** each document depends on the local code and on what was retrieved and written in earlier steps.



Document the "stem_word" function.

Description: This function performs Czech word stemming with optional aggressive morphological processing. … **Args**: word(str) A Czech word to be stemmed, which .. **Raises**: None explicitly raised, though mixed-case …

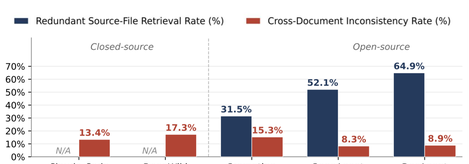
2. Existing Limitations

- Mostly on **Multi-Agent based framework**, such as RepoAgent [1], DocAgent [2], and CodeWiki [3]

≈ 50% avg. redundant source-file retrieval

≈ 13% avg. cross-document inconsistency

- **Redundant retrieval:** the same files are re-read for every component.
- **Inconsistency:** earlier outputs are never reused, so dependent documents contradict each other.
- **No hierarchy:** component docs are too local, repo docs too coarse.



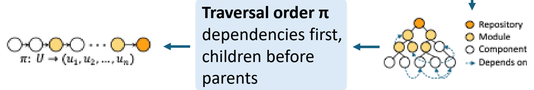
[1] Luo, Qinyu, et al. "Repoagent: An llm-powered open-source framework for repository-level code documentation generation." *EMNLP* 2024.
[2] Yang, Dayu, et al. "DocAgent: A multi-agent system for automated code documentation generation." *ACL 2025*.
[3] Hoang, Anh Nguyen, et al. "Codewiki: Evaluating ai's ability to generate holistic documentation for large-scale codebases." *ACL Findings 2026*.

3. Proposed Method : MemDocAgent

[Object] Hierarchical Repository-Level Code Documentation: Generating high-quality, complete and consistent hierarchical document in entire repository
[Key idea] Treat documentation as one long-horizon agentic task: a single agent walks the whole repository in one continuous trajectory and reads, writes and verifies against a shared memory

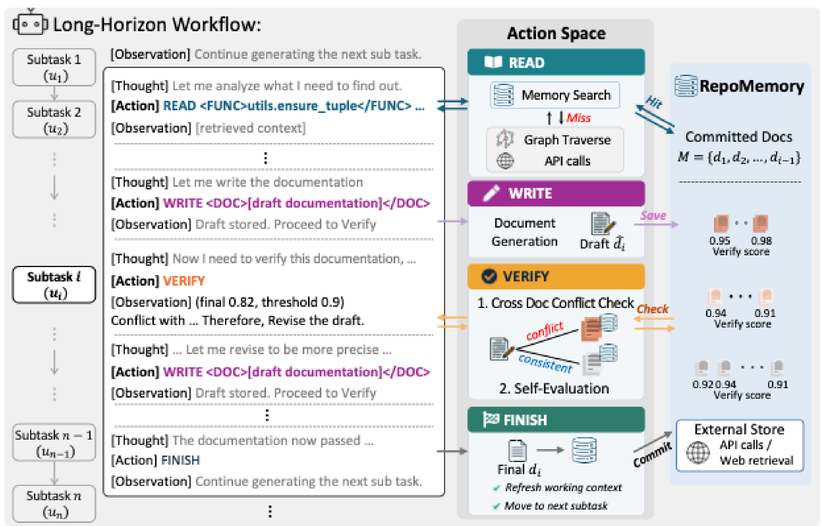
A. Dependency-Aware Traversal Guiding

AST parsing functions, methods and classes become nodes → **Dependency graph G** calls, inheritance, attribute access, imports → **Collapse cycles** Tarjan SCC into super-nodes → **Traversal order π** dependencies first, children before parents

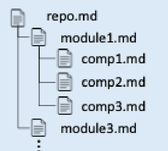


- π satisfies (1) component-level dependency order and (2) granularity-level hierarchy, so every unit is documented only after all of its dependencies and child units. This fixes structure and coverage by construction.

B. Memory-Guided Agentic Interaction



Output: Hierarchical Documentation (D)



No redundant reads
Memory lookups replace repeated raw-context retrieval

Consistent docs
Verified references and NLI checks align related documents

Full hierarchy
Traversal order covers component, module and repository levels

4. Experimental Setup

- **Data**: 20 Python repositories from DevEval
- **Backbones**: Qwen3-Coder-30B-A3B and GPT-5-mini with identical settings (up to 10 steps per sub-task, 2 revisions, threshold 0.9)
- **Baselines**: Prompting, RepoAgent, DocAgent, CodeWiki (open) / DeepWiki, Claude Code (closed)
- **Metrics**: Completeness, Truthfulness, Helpfulness (LLM judge), and the new *Information Sufficiency*

※ *Information Sufficiency*: whether the documentation contains enough information to reproduce source code that integrates and executes coherently within the repository

Remove function body → **Inject** docs in three granularity levels → **Regenerate** with an LLM generator → **Evaluate** Pass@k, CodeBLEU

5. Main Results

- Average over 20 repositories

(a) Overall results on average completeness, truthfulness, and helpfulness.

Backbone	Method	#Doc	Completeness			Truthfulness			
			Section Score	Coverage Score	Complete.	Consistency	Relevance	Truthful.	**Helpfulness**
Open-Source Baselines									
Qwen3-Coder	Prompting	3,323	0.838	0.837	0.838	0.809	0.803	0.649	0.458
	RepoAgent	3,650	0.659	0.967	0.813	0.910	0.918	0.835	0.391
	DocAgent	3,198	0.792	0.898	0.845	0.900	0.889	0.800	0.556
	CodeWiki	80	0.770	0.473	0.622	0.906	0.852	0.772	0.628
	MemDocAgent	3,323	**0.959**	**0.998**	**0.979**	**0.968**	**0.946**	**0.916**	**0.690**
GPT-5-mini	Prompting	3,323	0.790	0.473	0.632	0.883	0.840	0.741	0.617
	RepoAgent	3,650	0.555	**1.000**	0.778	0.898	0.947	0.850	0.449
	DocAgent	3,198	0.776	0.944	0.860	0.898	0.925	0.831	0.671
	CodeWiki	80	**0.975**	0.579	0.777	0.847	0.771	0.653	0.708
	MemDocAgent	3,323	0.916	**1.000**	**0.958**	**0.983**	**0.968**	**0.952**	**0.800**
Closed-Source Baselines									
Claude-Code (`/init`)		20	1.000*	0.488	0.744	0.845	0.869	0.734	0.696
DeepWiki		404	1.000*	0.840	0.920	0.815	0.935	0.762	0.755

(b) Results separated by documentation granularity (component-, module-, and repository-level).

Backbone	Method	Component-Level			Module-Level			Repository-Level		
		Complete.	Truthful.	Helpful.	Complete.	Truthful.	Helpful.	Complete.	Truthful.	Helpful.
Open-Source Baselines										
Qwen3-Coder	Prompting	0.886	0.694	0.577	0.870	0.666	0.313	0.758	0.587	0.483
	RepoAgent	0.813	0.835	0.391	–	–	–	–	–	–
	DocAgent	0.845	0.800	0.556	–	–	–	–	–	–
	CodeWiki	–	–	–	0.630	0.769	0.606	0.613	0.775	0.650
	MemDocAgent	**0.987**	**0.962**	**0.691**	**0.975**	**0.902**	**0.702**	**0.975**	**0.883**	**0.677**
GPT-5-mini	Prompting	0.631	0.761	0.627	0.626	0.766	0.626	0.640	0.696	0.598
	RepoAgent	0.778	0.850	0.449	–	–	–	–	–	–
	DocAgent	0.860	0.831	0.671	–	–	–	–	–	–
	CodeWiki	–	–	–	0.809	0.597	0.737	0.744	0.708	0.679
	MemDocAgent	**0.952**	**0.974**	**0.750**	**0.974**	**0.924**	**0.858**	**0.949**	**0.959**	**0.791**

MemDocAgent vs. strongest baseline (Qwen3-Coder)

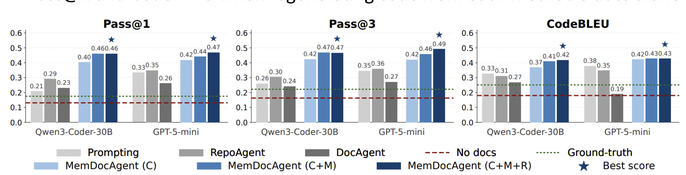
+15.9% Completeness

+9.7% Truthfulness

+9.9% Helpfulness

6. Information Sufficiency

- Pass@k and CodeBLEU when regenerating code from each method's docs alone



- No docs 0.131, human docstring 0.175. Richer hierarchy (C → C+M → C+M+R) lifts each score steadily, and generated docs beat human docstrings

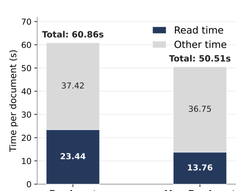
7. Ablation Study

- Removing any component hurts every metric (Truthfulness −13.5 to −14.8 pts)
- Generic vector memory gives no real gain (Truthfulness 0.760 vs. 0.916)
- Shuffled traversal drops Truthfulness 0.741 → 0.543 (10 repos)

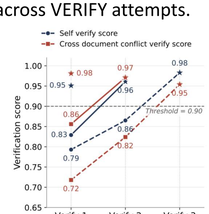
Method	Documentation Quality			Information Sufficiency		
	Completeness	Truthfulness	Helpfulness	Pass@1	Pass@3	CodeBLEU
MemDocAgent	**0.979**	**0.916**	**0.690**	**0.461**	**0.467**	**0.419**
w/o think	0.917 (−6.2)	0.777 (−13.9)	0.583 (−10.7)	0.312 (−14.9)	0.319 (−14.8)	0.225 (−19.4)
w/o memory system	0.903 (−7.6)	0.768 (−14.8)	0.593 (−9.7)	0.324 (−13.7)	0.333 (−13.4)	0.190 (−22.9)
w/o conflict verify	0.900 (−7.9)	0.781 (−13.5)	0.592 (−9.8)	0.282 (−17.9)	0.290 (−17.7)	0.233 (−18.6)

8. Efficiency and Verification

- Average read time comparison across 20 repositories



- Documentation improvements across VERIFY attempts.



−41% read time vs. DocAgent

2.62 read calls per component

63.8% of components pass VERIFY on the first try

9. Conclusion

- Repository documentation recast as a long-horizon agentic task with shared, verified memory
- Best completeness, truthfulness, helpfulness and information sufficiency, with less read time

Paper Code Profile