LEGOMem: Modular Procedural Memory for Multi-agent LLM Systems for Workflow Automation

Extended Abstract

Dongge Han
Microsoft
Redmond, US
donggehan@microsoft.com

Camille Couturier
Microsoft
Redmond, US
Camille.Couturier@microsoft.com

Daniel Madrigal Diaz
Microsoft
Redmond, US
danielmad@microsoft.com

Xuchao Zhang
Microsoft
Redmond, US
xuchaozhang@microsoft.com

Victor Rühle
Microsoft
Redmond, US
virueh@microsoft.com

Saravan Rajmohan
Microsoft
Redmond, US
saravan.rajmohan@microsoft.com

ABSTRACT

We introduce LEGOMem, a modular procedural memory framework for multi-agent LLM systems in workflow automation. LEGOMem distills successful executions into reusable full-task and subtask memories and allocates them to orchestrators and task agents to improve planning and execution. Across three retrieval variants, experiments on OfficeBench show consistent gains of 12–13 absolute points over memory-less and baseline methods, highlighting the importance of procedural memory for workflow automation.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent systems**; *Reasoning about beliefs and knowledge.*

KEYWORDS

Multi-agent systems; Procedural memory; LLM Agents; Workflow

ACM Reference Format:
Dongge Han, Camille Couturier, Daniel Madrigal Diaz, Xuchao Zhang, Victor Rühle, and Saravan Rajmohan. 2026. LEGOMem: Modular Procedural Memory for Multi-agent LLM Systems for Workflow Automation: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/VLUA1303

1 INTRODUCTION

LLMs are increasingly deployed as agents to automate complex multi-step workflows [1, 2, 4, 9, 11–13, 15–18, 21, 22, 24]. To manage the diversity and compositionality of such tasks, recent systems often adopt multi-agent [14, 19] designs, where multiple LLM-based agents collaborate, specialize, or delegate responsibilities across roles and tools [3, 5, 7, 20, 23]. We introduce LEGOMem, a modular procedural memory framework designed for multi-agent LLM systems. In this work, we focus on a common and practical subclass of multi-agent architectures, where a central orchestrator performs planning and delegates subtasks to specialized tool-using task agents, as exemplified by the Magentic-One framework [7, 20]. Our goal is to equip both orchestrators and task agents with memory grounded in prior task trajectories, enabling them to perform better planning, coordination, and task executions. To this end, we design LEGOMem to distill successful executions into structured memory units: full-task memories (task-level plans and reasoning traces) and subtask memories (agent behavior and tool interactions). These modular memories are stored in a memory bank, indexed by semantic embeddings, and reused at inference time to augment planning and execution. LEGOMem is instantiated as a retrieval augmentation (RAG) [6, 8, 10] layer over existing multi-agent systems. During a new task, the orchestrator receives relevant full-task memories to support task decomposition and agent selection, while each task agent is assigned subtask memories aligned with its delegated subtasks. We explore three memory retrieval strategies—*vanilla*, (where we retrieve task-level memories and allocate subtask memories to relevant agents), *dynamic retrieval*, (where we store also subtask memories and use subtask queries to find semantically similar subtask memory for agents) and *query rewriting*, (where we use an LLM to expand a task query into subtask plans for subtask memory retrieval)—to study how retrieval and memory specificity affect multi-agent performance. This framework allows us to systematically investigate key questions in multi-agent memory design, including where memory should be placed, how it should be retrieved, and which agents benefit most from it.

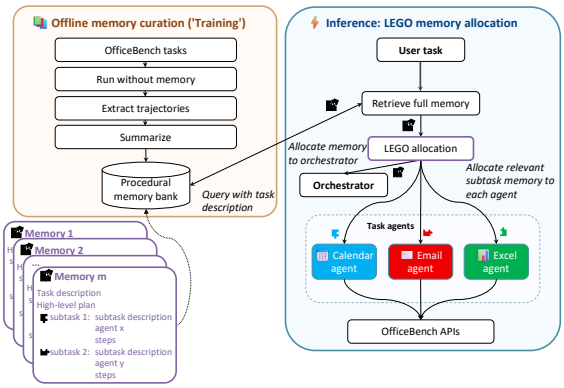

Figure 1: Overview of the LEGOMem framework

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/VLUA1303

Table 1: Performance comparison across memory variants, task levels, and multi-agent teams. Results show mean success rates across different LEGOMem variants compared with baseline methods, each data-point is averaged over three random seeds.

	LLM team				Hybrid (LLM + SLM) team				SLM team			
	Level 1	Level 2	Level 3	**Overall**	Level 1	Level 2	Level 3	**Overall**	Level 1	Level 2	Level 3	**Overall**
Baseline methods												
No memory	49.31	58.52	33.33	45.83	45.14	48.89	16.95	35.31	36.81	34.81	7.34	24.78
Synapse	**59.72**	**75.56**	43.50	58.11	46.53	**68.15**	29.94	46.49	36.81	42.22	20.90	32.24
AWM	54.17	58.52	35.03	48.03	43.75	55.56	18.64	37.50	35.42	36.30	12.99	26.97
Our methods												
LEGOMem	57.99	73.33	**47.46**	**58.44**	**49.31**	62.22	36.16	48.03	**38.89**	**54.07**	25.42	**38.16**
LEGOMem-Dynamic	56.25	**75.56**	43.79	57.12	44.44	65.93	36.16	47.59	**38.89**	50.37	**27.12**	37.72
LEGOMem-QueryRewrite	54.17	72.59	42.94	55.26	47.22	66.67	**40.11**	**50.22**	36.81	48.89	26.55	36.40

Table 2: Comparing performance with various memory placement mechanism across LEGOMem variants.

	LLM variants				Hybrid (LLM + SLM) variants			
	Level 1	Level 2	Level 3	**Overall**	Level 1	Level 2	Level 3	**Overall**
Orchestrator + Agent memory								
LEGOMem	**57.99**	73.33	**47.46**	**58.44**	**49.31**	62.22	36.16	48.03
LEGOMem-Dynamic	56.25	75.56	43.79	57.12	44.44	65.93	36.16	47.59
LEGOMem-QueryRewrite	54.17	72.59	42.94	55.26	47.22	66.67	**40.11**	**50.22**
Orchestrator memory (planning) + Agent memory								
LEGOMem	54.86	**76.30**	35.03	53.51	45.14	63.70	30.51	44.96
LEGOMem-Dynamic	54.86	73.33	41.81	55.26	46.53	64.44	32.77	46.49
LEGOMem-QueryRewrite	51.39	70.37	42.94	53.73	49.31	59.26	35.59	46.93
Orchestrator memory								
LEGOMem	51.39	74.07	38.98	53.29	45.83	**68.89**	32.77	47.59
Task Agent memory								
LEGOMem	50.00	63.70	38.98	49.78	44.44	46.67	19.21	35.31
LEGOMem-Dynamic	49.31	62.96	38.98	49.34	47.22	55.56	23.16	40.35
LEGOMem-QueryRewrite	54.86	66.67	35.03	50.66	44.44	54.81	24.29	39.69
No memory								
No memory	49.31	58.52	33.33	45.83	45.14	48.89	16.95	35.31

2 EXPERIMENTS

We evaluate LEGOMem on the OfficeBench benchmark, comparing its variants with strong baselines across LLM-only, hybrid, and SLM-only multi-agent teams. Table 1 compares the performance of LEGOMem with baseline methods across different task levels and agentic team configurations. LEGOMem variants consistently outperform baseline methods in terms of overall success rate. All three LEGOMem variants show similar, consistent performance. Compared with memory-less teams, LEGOMem improves overall task success rate by +12.61%, +12.72%, and +13.38% absolute points on LLM, Hybrid and SLM teams, respectively. Importantly, LEGOMem enables smaller models to close the gap with, and sometimes outperform, larger ones. For example, the Hybrid team with LEGOMem-QueryRewrite achieves 50.22%, surpassing the memory-less LLM team (45.83%). Table 2 summarizes our ablation results across different memory retrieval variants, memory allocation strategies, and memory placement settings. Across retrieval variants, vanilla LEGOMem, LEGOMem-Dynamic, and QueryRewrite achieve similar overall performance, indicating that memory placement and allocation matter more than retrieval strategy in full-memory settings. However, in task-agent-only configurations—especially for Hybrid teams with smaller task agents—fine-grained subtask retrieval (LEGOMem-Dynamic, LEGOMem-QueryRewrite) outperforms vanilla LEGOMem by 4–5%, highlighting its benefit when global planning signals are weaker. Ablations further show that joint allocation of memory to both orchestrator and task agents yields the best results: orchestrator memory is essential for effective planning and delegation, while agent memory improves execution precision. Task-agent-only memory, although better than no memory, remains insufficient without orchestrator-level coordination.

3 CONCLUSION

We introduced LEGOMem, a modular procedural memory framework that enables multi-agent LLM systems to reuse prior task executions through role-aware full-task and subtask memories for reliable multi-agent workflow automation. Experiments on workflow automation tasks show that LEGOMem consistently improves success rates over memory-less and baseline method.

REFERENCES

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691* (2022).

[2] Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Wenjing Hu, Yuchen Mao, et al. 2024. Spider2-v: How far are multimodal agents from automating data science and engineering workflows? *Advances in Neural Information Processing Systems* 37 (2024), 107703–107744.

[3] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848* 2, 4 (2023), 6.

[4] Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, et al. 2024. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428* (2024).

[5] Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, et al. 2025. Multi-Agent Collaboration via Evolving Orchestration. *arXiv preprint arXiv:2505.19591* (2025).

[6] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281* (2024).

[7] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468* (2024).

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* 2, 1 (2023).

[9] Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. An llm compiler for parallel function calling. In *Forty-first International Conference on Machine Learning*.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[11] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

[12] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135* (2023).

[13] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2998–3009.

[14] Peter Stone and Manuela Veloso. 2000. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots* 8, 3 (2000), 345–383.

[15] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 186345.

[16] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091* (2023).

[17] Weixuan Wang, Dongge Han, Daniel Madrigal Diaz, Jin Xu, Victor Rühle, and Saravan Rajmohan. 2025. OdysseyBench: Evaluating LLM Agents on Long-Horizon Complex Office Application Workflows. *arXiv preprint arXiv:2508.09124* (2025).

[18] Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. 2024. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056* (2024).

[19] Michael Wooldridge. 2009. *An Introduction to MultiAgent Systems* (2nd ed.). Wiley Publishing.

[20] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

[21] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems* 37 (2024), 52040–52094.

[22] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

[23] Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, et al. 2024. Ufo: A ui-focused agent for windows os interaction. *arXiv preprint arXiv:2402.07939* (2024).

[24] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854* (2023).