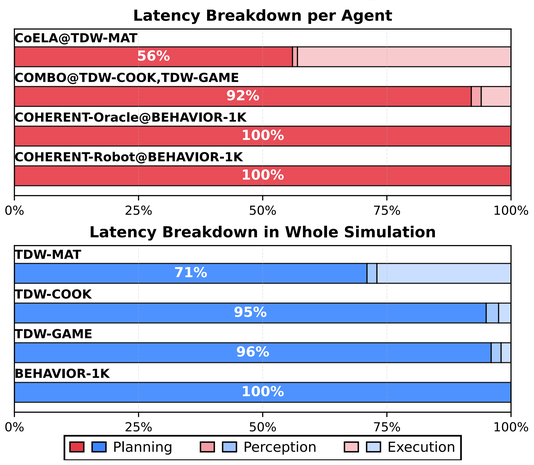
AgenticCache: Cache-Driven Asynchronous Planning for Embodied AI Agents

Hojoon Kim[1], Yuheng Wu[2], Thierry Tambe[2]
[1]Seoul National University [2]Stanford University

MLSys

Motivation & Problem

LLM-driven embodied AI agents are slow.

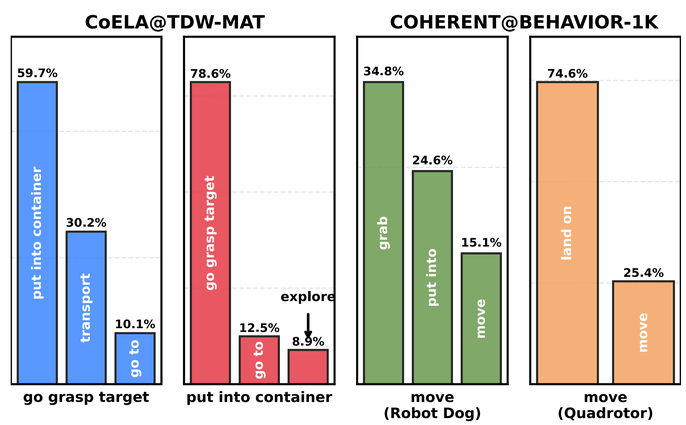


=> How to can we make AI Agents faster?

Key Insight: Plan Locality

Next plan is predictable from the current one

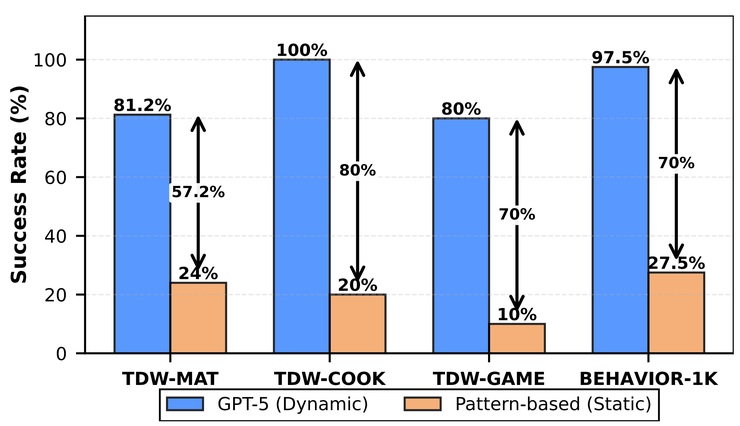
→ *Plans can be cached & reused*



Challenges of Caching Plan

But cached plans go stale

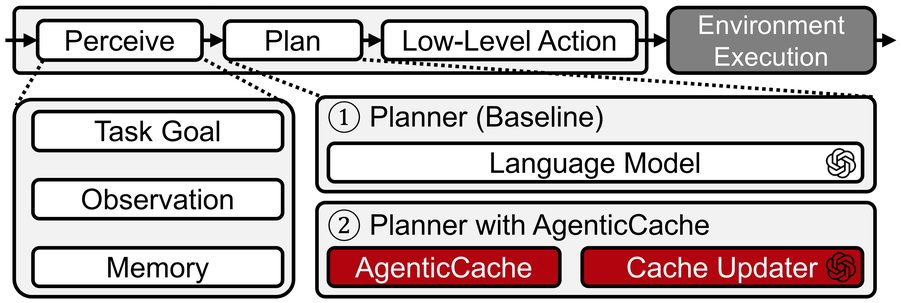
→ *Refresh the cache without blocking the agent*



AgenticCache

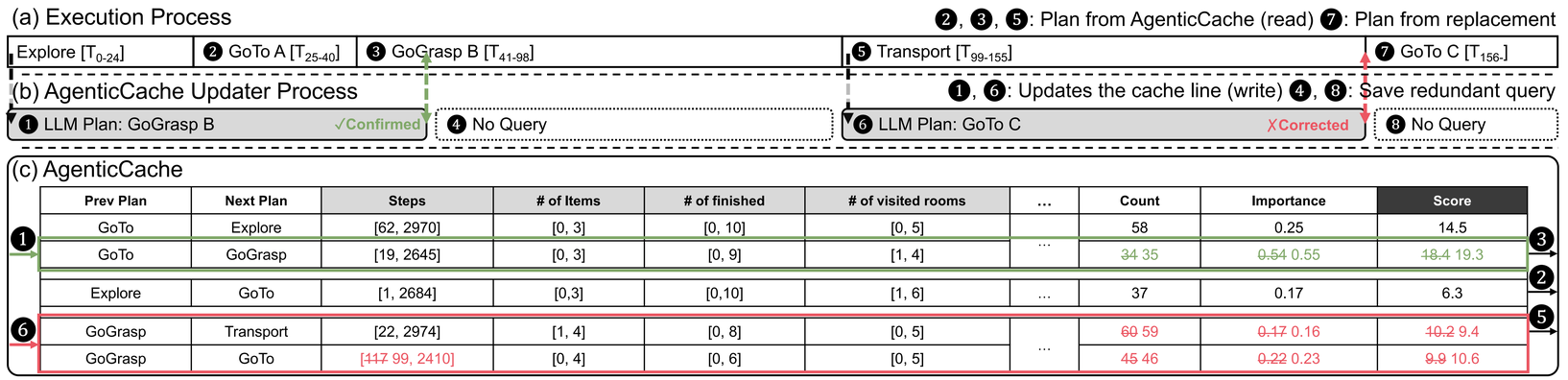
Cache + Async LLM = Branch Predictor for Plans

- A runtime **plan cache** answers each per-step query in microseconds
- A background **Cache Updater** validates & refines cached entries asynchronously
- Moves the LLM **off the critical path** of the agent loop
  → *Hit the cache, skip the wait*



① Cache as Local Planner

- 2-gram entries $\langle P_i \rightarrow P_j \rangle$ keyed by task-state metadata
- Mirrors **hybrid branch predictors** (local + global history)



② Async Cache Updater

- **Confirm** → bump count, suppress redundant LLM calls
- **Correct** → swap stale plan immediately, refresh entry

Results

Across 4 multi-agent benchmarks, AgenticCache cuts latency 1.85–7.4× while matching or beating LLM-only baseline accuracy.

Strategy	TDW-MAT		TDW-COOK		TDW-GAME		BEHAVIOR-1K	
	SR	L	SR	L	SR	L	SR	L
Baseline	90%	41.3	94%	12.9	100%	7.9	97%	3.4
Parallelized	89%	43.7	100%	14.7	0%	15.8	97%	3.0
Speculative	81%	36.4	83%	6.1	11%	6.1	94%	3.8
AgenticCache	**89%**	**22.3**	**100%**	**1.8**	**100%**	**1.1**	**100%**	**1.6**

Cold start, no prefilling: AgenticCache still cuts latency 1.3–1.9× while matching or beating baseline accuracy.

Setting	Strategy	GPT-5		GPT-5-mini	
		SR	L	SR	L
Standard	Baseline	90.0%	5.06	85.0%	4.11
	AgenticCache	**93.3%**	**2.63**	**85.0%**	**2.67**
Long-Horizon	Baseline	82.2%	11.57	62.8%	8.17
	AgenticCache	**80.6%**	**6.19**	**69.4%**	**6.29**

Headline numbers

- **+22%** task success rate (avg)
- **−65%** simulation latency (avg)
- **−50%** token usage (avg)
- Up to **86% latency / 79% cost** reduction (TDW-COOK, GPT-5)

Why it works

- Plan locality is exploitable in embodied tasks
- Cache footprint <1KB per agent
- Complementary to KV cache, vLLM, SGLang

→ *Cache-driven planning = low-latency, low-cost agents*