PAPER CLUB

DISCUSSION 01

COORDINATION IN MULTI-AGENT SYSTEMS

GENERATIVE AGENTS: INTERACTIVE SIMULACRA OF HUMAN BEHAVIOR

LLMs simulate human behavior at a single moment but lack long-term coherence. This paper builds agents that remember, reflect, and plan producing believable individual and emergent social behavior.

- **Context**: Rule-based (FSMs, behavior trees), RL, and cognitive architectures (SOAR) all fail in open worlds or require manual authoring. LLMs alone can't maintain coherence over time due to context limits.
- **Co-ordination:** 25 agents in "Smallville" perceive nearby agents/objects each timestep, communicate in natural language, and update each other's memories. Information diffusion, relationship formation, and event coordination all emerge with no scripted rules.
- **Application:** SIMs-like sandbox
- **Mathematics:**

$$score = \alpha \cdot recency + \alpha \cdot importance + \alpha \cdot relevance$$

For all **α=1**
Recency = exponential decay (factor 0.995)
Importance = LLM integer 1–10
Relevance = cosine similarity of memory embeddings

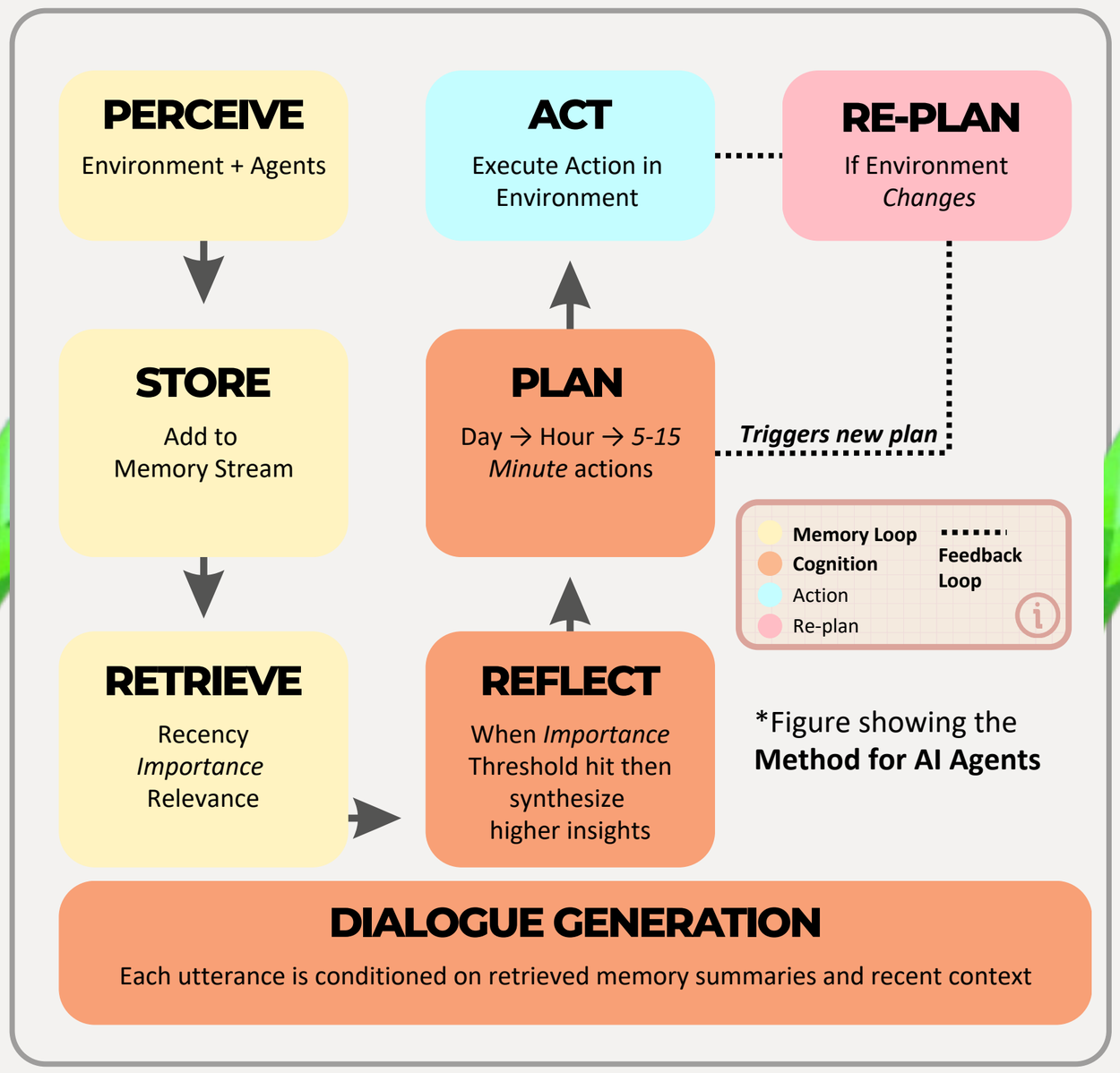


*Figure showing the **Method for AI Agents**

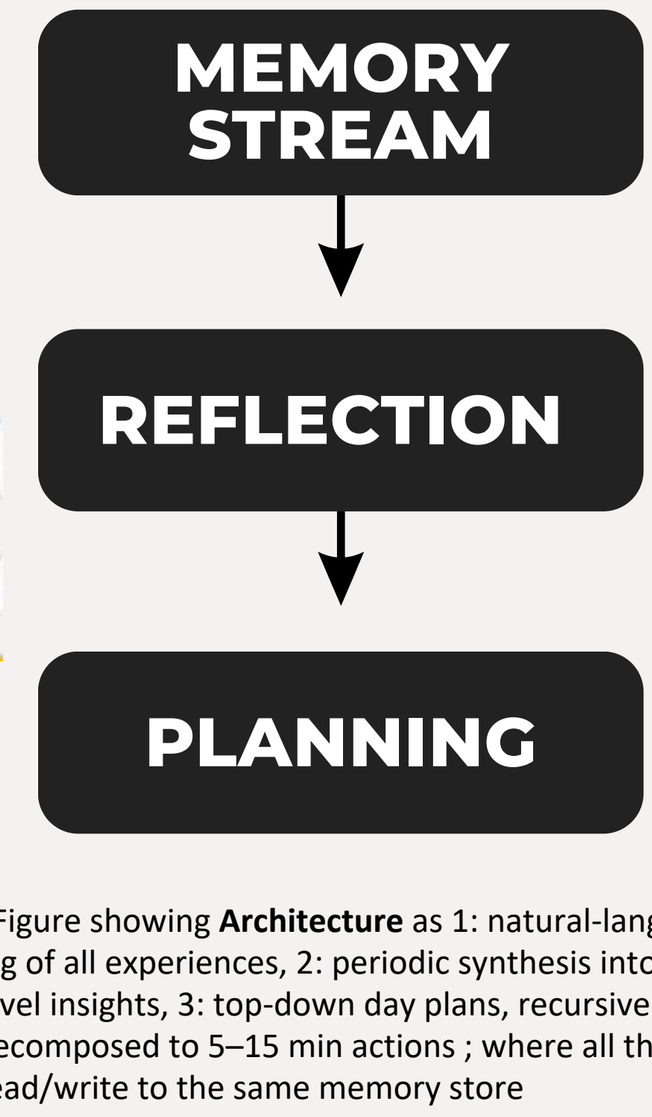


*Figure showing **Architecture** as 1: natural-language log of all experiences, 2: periodic synthesis into higher-level insights, 3: top-down day plans, recursively decomposed to 5–15 min actions ; where all three read/write to the same memory store

Memory, reflection, and planning each contribute **causally** to believable behavior and removing any one degrades performance significantly. The key failure modes **(retrieval misses, hallucinated embellishments, over-formal LLM style)** point to the limits of the underlying **model**, not the architecture itself.

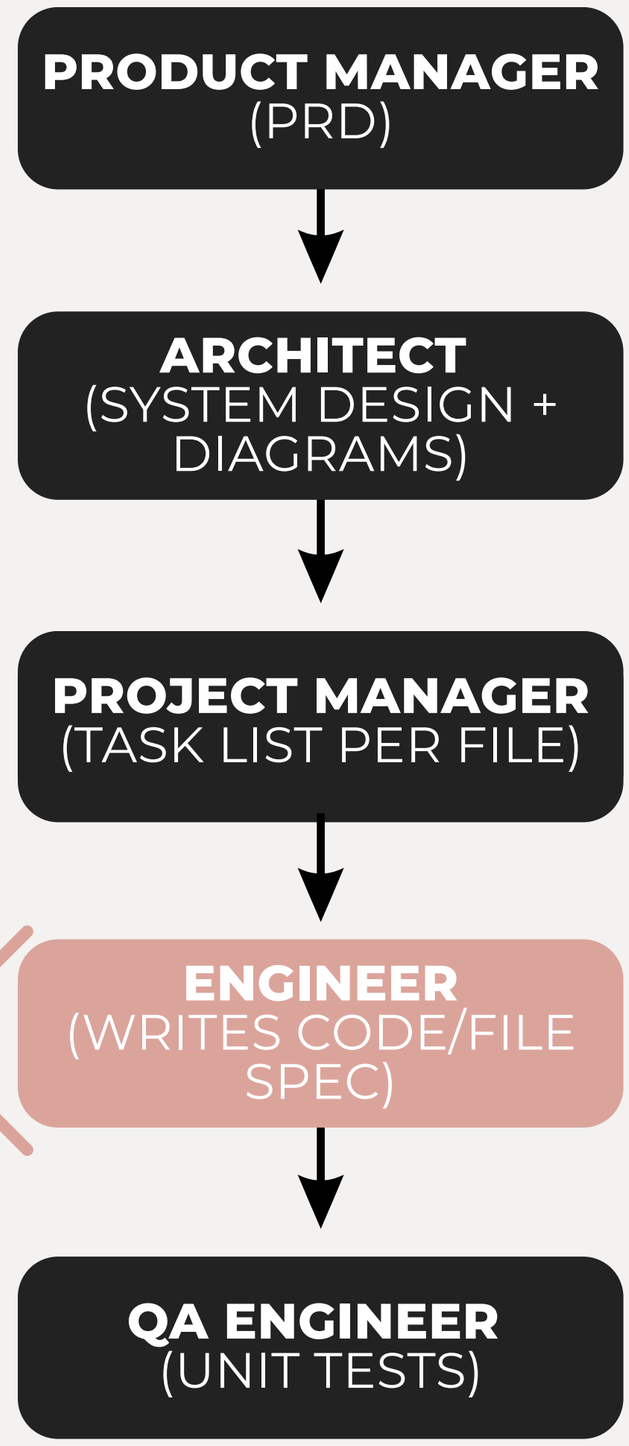
BELIEVABLE LONG-HORIZON AGENTS AREN'T JUST ABOUT A BETTER LLM BUT THEY NEED A MEMORY ARCHITECTURE THAT DECIDES WHAT TO REMEMBER, WHAT TO INFER, AND WHEN TO ACT

For more information please visit **https://arxiv.org/abs/2304.03442**

METAGPT: META *PROGRAMMING* FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK

Naively chaining LLMs causes cascading hallucinations through idle chatter. MetaGPT encodes human SOPs as prompt sequences, forcing structured artifact handoffs (PRDs, diagrams, code) instead of free-form dialogue.

- **Context**: AutoGPT, LangChain, ChatDev all lack systematic decomposition and structured output formats. Human software teams succeed because SOPs define roles, deliverables, and quality gates but MetaGPT imports this directly
- **Co-ordination:** Agents publish structured outputs and each subscribes only to role-relevant messages. An agent activates only when all its prerequisite dependencies have arrived which eliminates chatter and information overload simultaneously.
- **Application:** Mini-games (*Flappy Bird*, Snake, 2048) since it generates complete multi-file Python applications from a single natural-language prompt



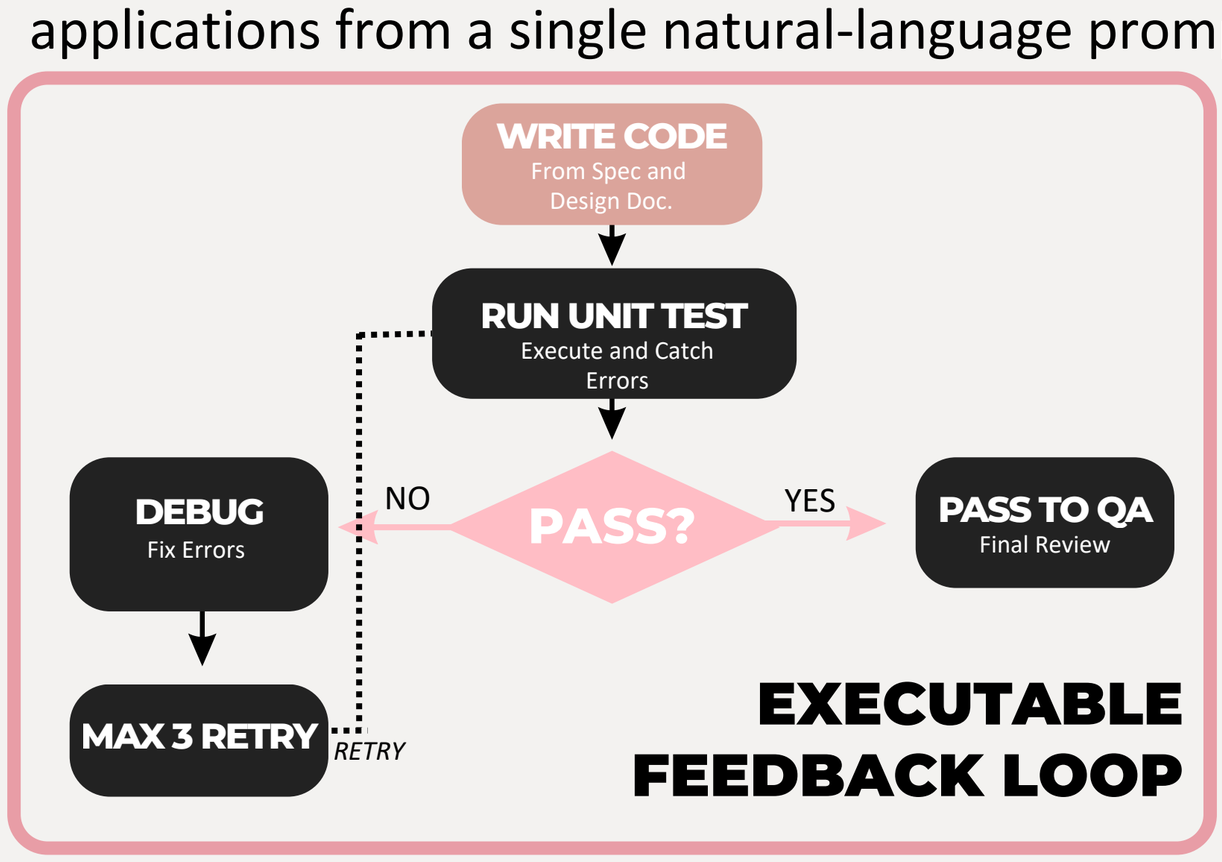


*Figure showing **ASSEMBLY LINE PIPELINE** where each agent outputs a **structured** document; the next agent's input is that document, not raw dialogue

- **Standard Unbiased Estimator for Functional Correctness:**

$$Pass@k = E[1 - C(n-c,k)/C(n,k)]$$

SOPs transform multi-agent collaboration from a hallucination-prone chat loop into a disciplined engineering pipeline where each role producing verifiable, structured artifacts the next can build on. The executable feedback loop is what closes the gap between code that looks right and code that actually runs

STRUCTURE ISN'T A CONSTRAINT ON LLM CREATIVITY BUT RATHER IT'S WHAT MAKES MULTI-AGENT SYSTEMS RELIABLE ENOUGH TO SHIP.

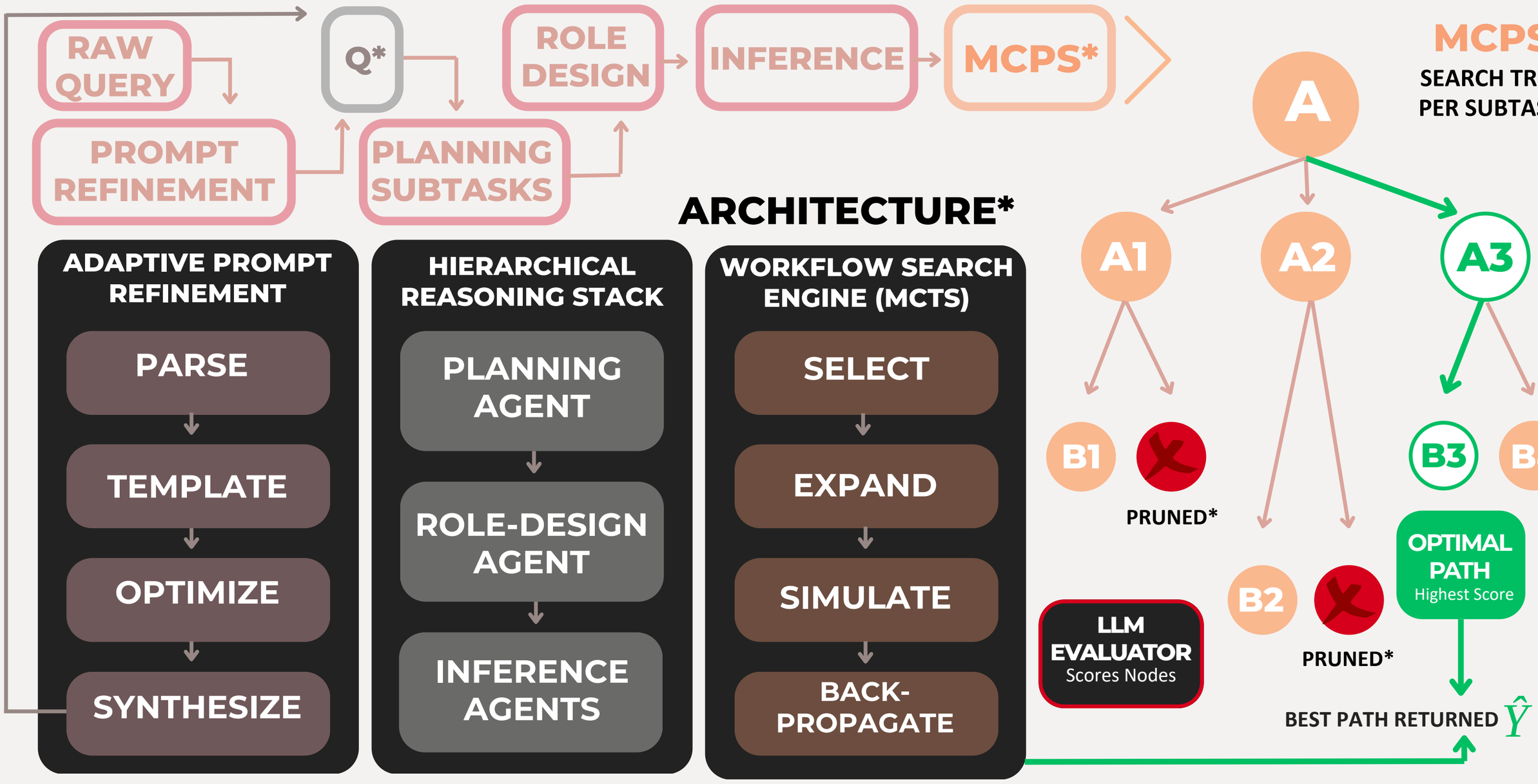
For more information please visit **https://arxiv.org/abs/2308.00352**

HALO: HIERARCHICAL AUTONOMOUS *LOGIC*-ORIENTED ORCHESTRATION FOR MULTI-AGENT LLM SYSTEMS

Existing MAS rely on predefined role spaces and static communication graphs, failing on expert-level tasks. HALO replaces both with dynamic role generation and MCTS-guided workflow search for best reasoning trajectories.

- **Context**: MetaGPT's static roles are brittle in open-ended settings. Centralized schedulers bottleneck at scale. No prior work combines hierarchical dynamic role instantiation with search-based execution optimization.
- **Co-ordination:** Planning agent decomposes the task sequentially, passing history H_k forward to each subtask. Mid-level agents generate role profiles on-the-fly per subtask. MCTS coordinates low-level agents: nodes = agent responses, edges = state transitions, backpropagation updates node quality scores.
- **Application:** same architecture handles code generation (HumanEval), general reasoning (MMLU), and arithmetic reasoning (MATH) without task-specific engineering so a lot of different kinds of reasoning problems

*END-TO-END PIPELINE



HALO shows that the bottleneck in multi-agent reasoning isn't the LLM but it's how tasks are decomposed and how agent collaboration is searched rather than scripted. Dynamic role generation and MCTS together let the system adapt to task complexity rather than assume it in advance.

TREAT MULTI-AGENT WORKFLOWS AS A SEARCH PROBLEM NOT A FIXED PIPELINE TO GET GOOD PERFORMANCE ON EXPERT-LEVEL DIFFICULT TASKS

For more information please visit **https://arxiv.org/abs/2505.13516**

CONCLUSION

	GENERATIVE AGENTS	METAGPT	HALO
MEMORY	Persistent stream	Structured artifacts	History H_k (Subtask outputs chained forward)
COORDINATION	**None (emergent)**	**Fixed SOP pipeline**	**Hierarchical + Search Plan → Design roles → MCTS**
ROLES	None (All Identical autonomous agents)	Fixed, predefined	Dynamic, per subtasK
TASK EXECUTION	Perceive → reflect → act loop	Linear handoff (Write → test → debug loop)	MCTS search (Select → expand → backprop)
DOMAIN	Social simulation	Software engineering	Domain-agnostic
KEY LIMIT	Expensive to run + no real-time	Rigid domain (Breaks outside software)	No cross-task learning (Each query from scratcH)

- **PROBLEM**

LLMs fail at ***sustained, multi-agent task***s not because of model capability but because of missing ***infrastructure*** like memory, coordination, and execution structure. All three papers are architectural responses to the same root cause.

- **SOLUTION**

Each paper directly solves the prior paper's binding constraint. Generative Agents adds ***memory and reflection***. MetaGPT replaces unstructured dialogue with ***enforced artifact*** handoffs. HALO replaces fixed roles with ***dynamic instantiation*** and ***search-based execution***. Each represents a structural shift in how agents are organized.

- **GAP**

None of the three systems ***learn across runs***. Performance resets to zero on every new task. Until agents can accumulate and transfer knowledge over time, the architecture is sophisticated but the system is still disposable.