

WESTWORLD - SEASON 1



AI AGENT CO-ORDINATION

ACROSS 3 RESEARCH PAPERS



AI Researcher @ U'Calgary

Prev.Markaz (YC W22). UM'26. NUST'27



GRIND

**DID AN AI AGENT DO THAT
OR IS THIS ABOUT ME?**





1

**CAN NOT RETRIEVE CORRECT
MEMORY DETAILS SOMETIMES
JUST SOME FRAGMENTS OF
MEMORY**



**STARTED TALKING ABOUT THINGS
THAT MADE NO SENSE JUST
STRAIGHT UP RANDOM THINGS TO
A STRANGER**

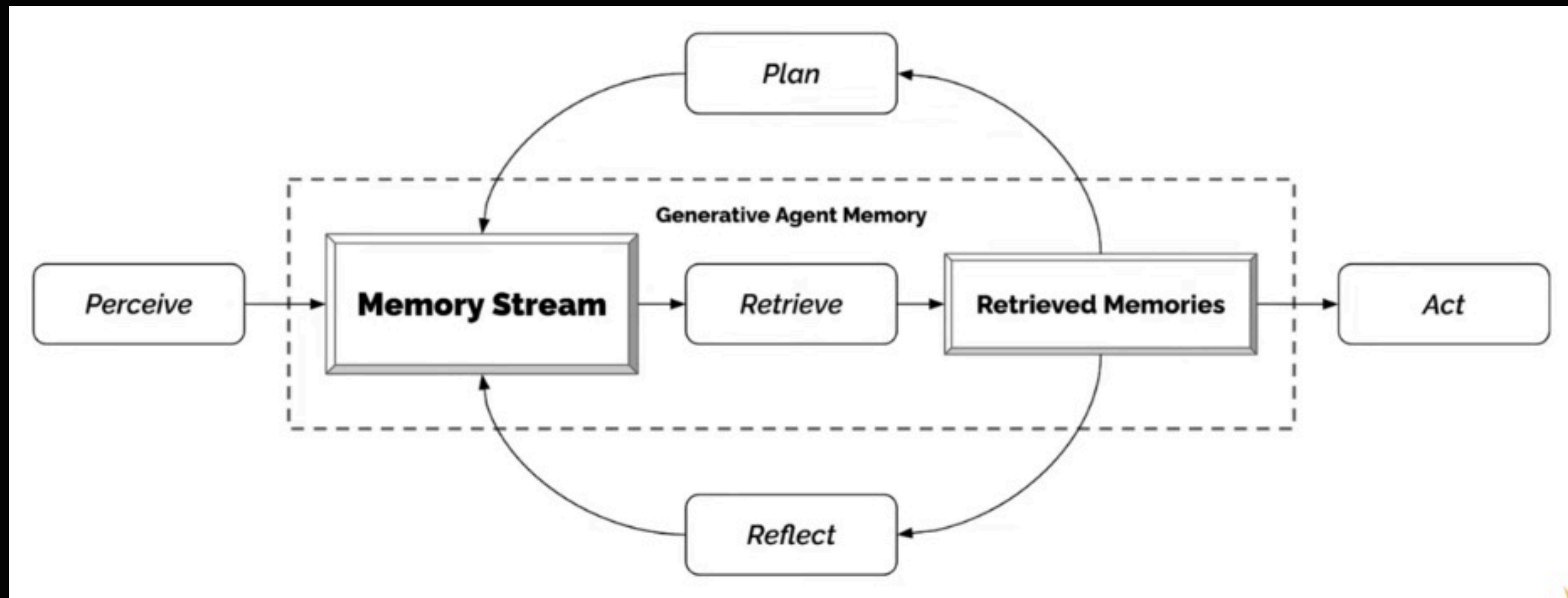
3

**CAN NOT RETRIEVE CORRECT
MEMORY DETAILS SOMETIMES
BUT RATHER JUST SOME
FRAGMENTS OF MEMORY**

PAPER 01

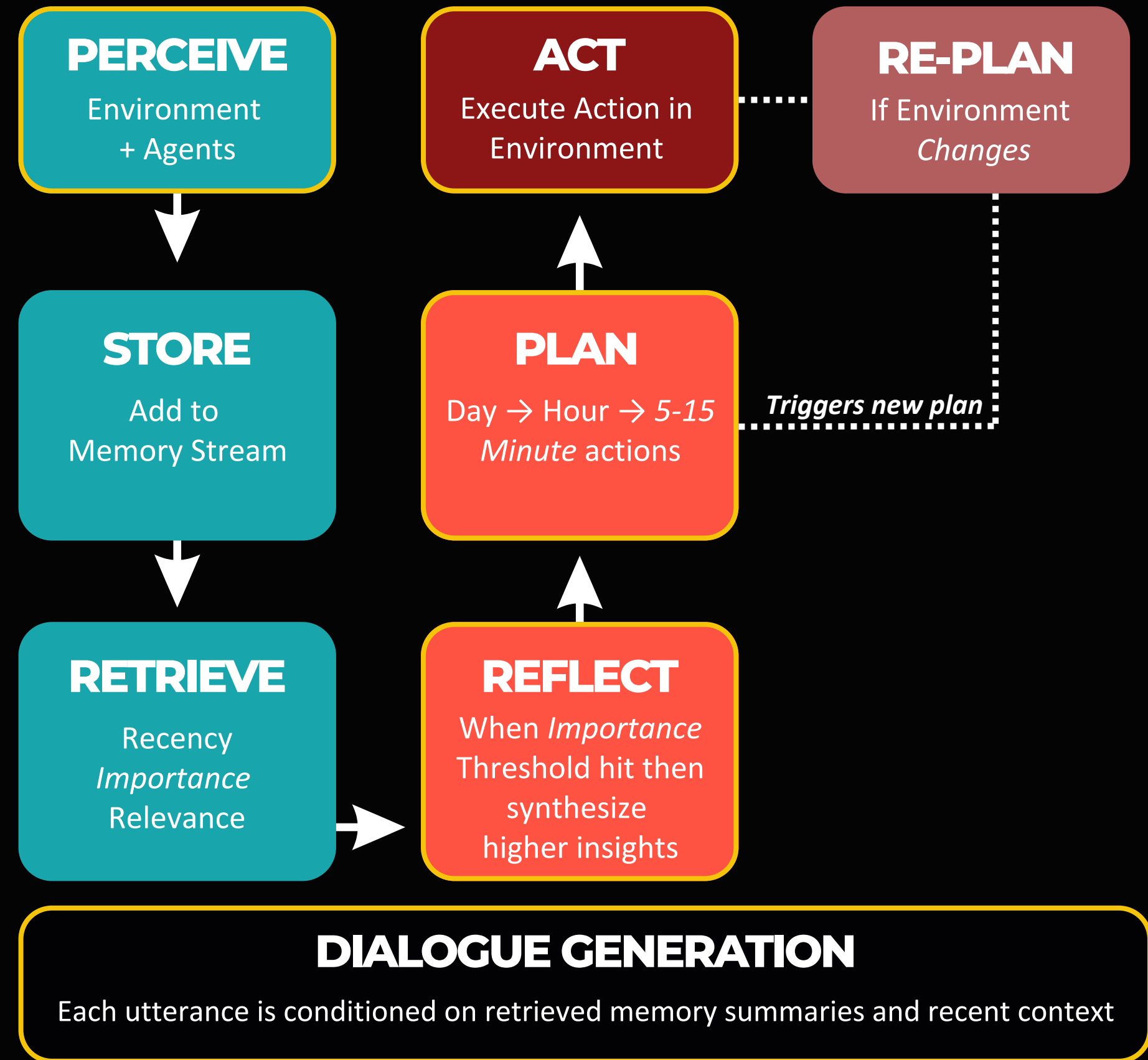
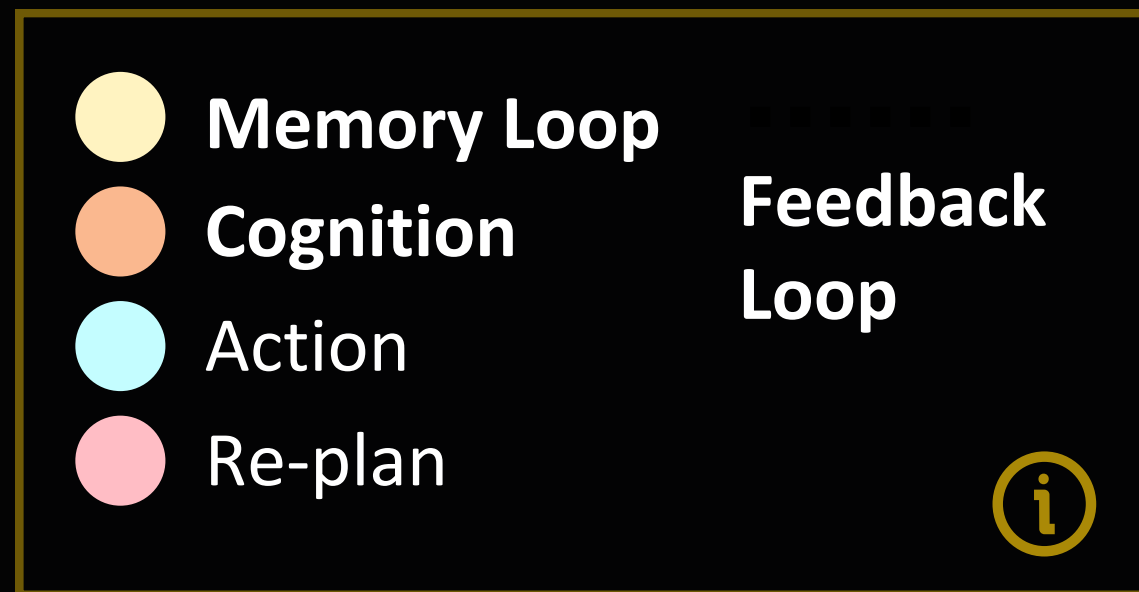


Generative Agents: Interactive Simulacra of Human Behavior



PAPER 01

Generative Agents: Interactive Simulacra of Human Behavior



PAPER 01



Generative Agents: Interactive Simulacra of Human Behavior

$$\textit{score} = \alpha \cdot \textit{recency} + \alpha \cdot \textit{importance} + \alpha \cdot \textit{relevance}$$

Recency = exponential decay
(factor 0.995)

Importance = LLM integer 1–10

Relevance = cosine similarity of
memory embeddings



DISCUSSION

**FIND OUT WHY THE
UNDERGROUND WATER
TUNNELS IN HAWAII ARE
GIVING OUT PICKLE
FLAVORED WATER **RIGHT NOW****

TIME: 1 MINUTE TO DISCUSS

NAIVELY CHAINED LLMS

**Put a bunch of random LLMs and chain
them together and let them figure out
the solution themselves**

~~NAIVELY CHAINED LLMS~~

~~Put a bunch of random LLMs and chain them together and let them figure out the solution themselves~~

SO BAD

PAPER 02

METAGPT: META PROGRAMMING FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK

PRODUCT MANAGER
(PRODUCT
REQUIREMENT
DOCUMENT)



ARCHITECT
(SYSTEM DESIGN +
DIAGRAMS)



PROJECT MANAGER
(TASK LIST PER FILE)



ENGINEER
(WRITES CODE/FILE
SPEC)

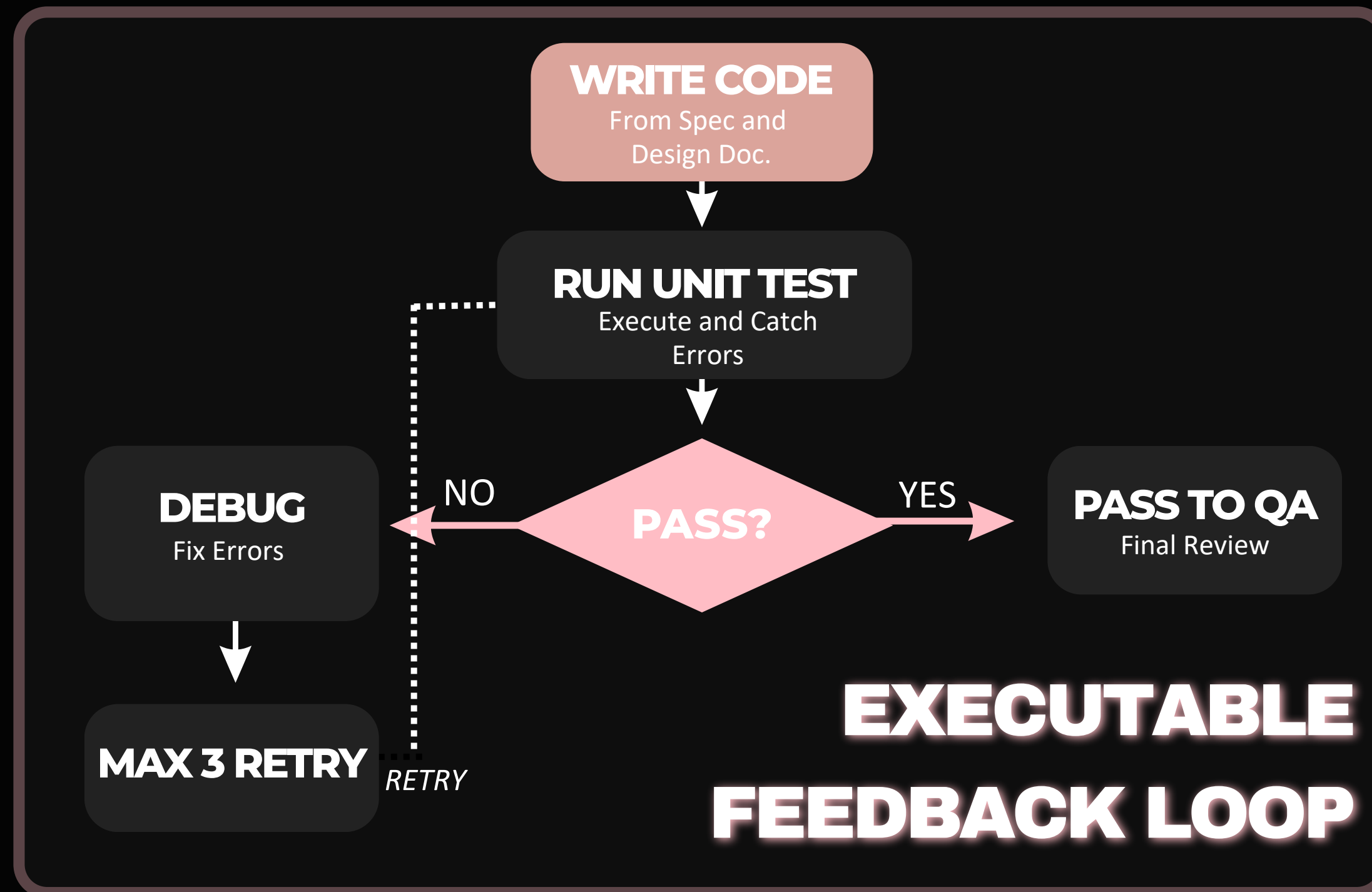


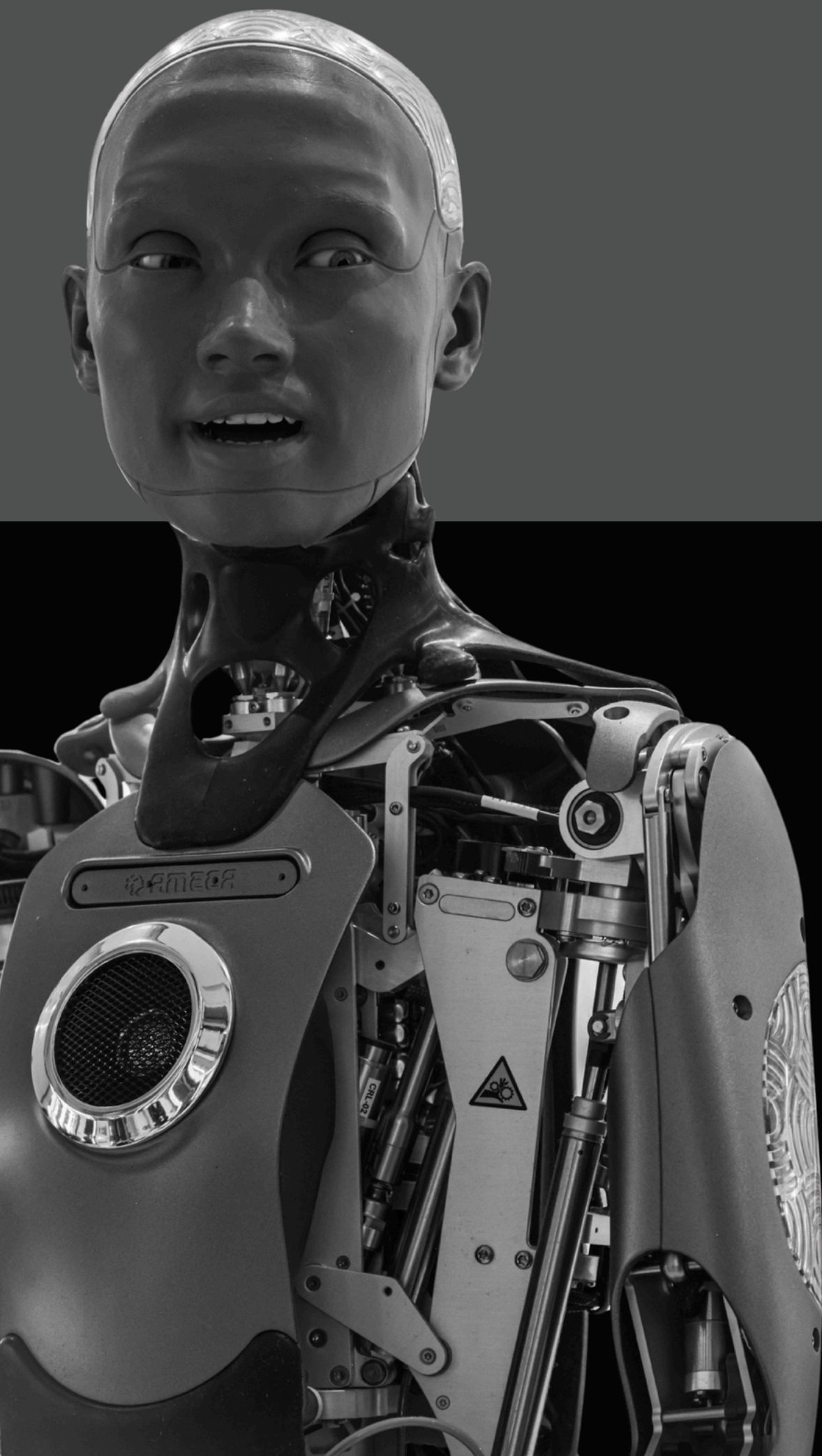
QA ENGINEER
(UNIT TESTS)



PAPER 02

METAGPT: META PROGRAMMING FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK





STRUCTURED OUTPUTS

Agents exchange fixed-schema documents (PRDs, design specs, diagrams) instead of free-form chat

PUBLISH SUBSCRIBE MODEL

Every agent publishes its output to one shared message pool rather than messaging peers directly in a chain

Standard Unbiased Estimator for Functional Correctness:

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

Combinatorics-based metric for **code correctness**

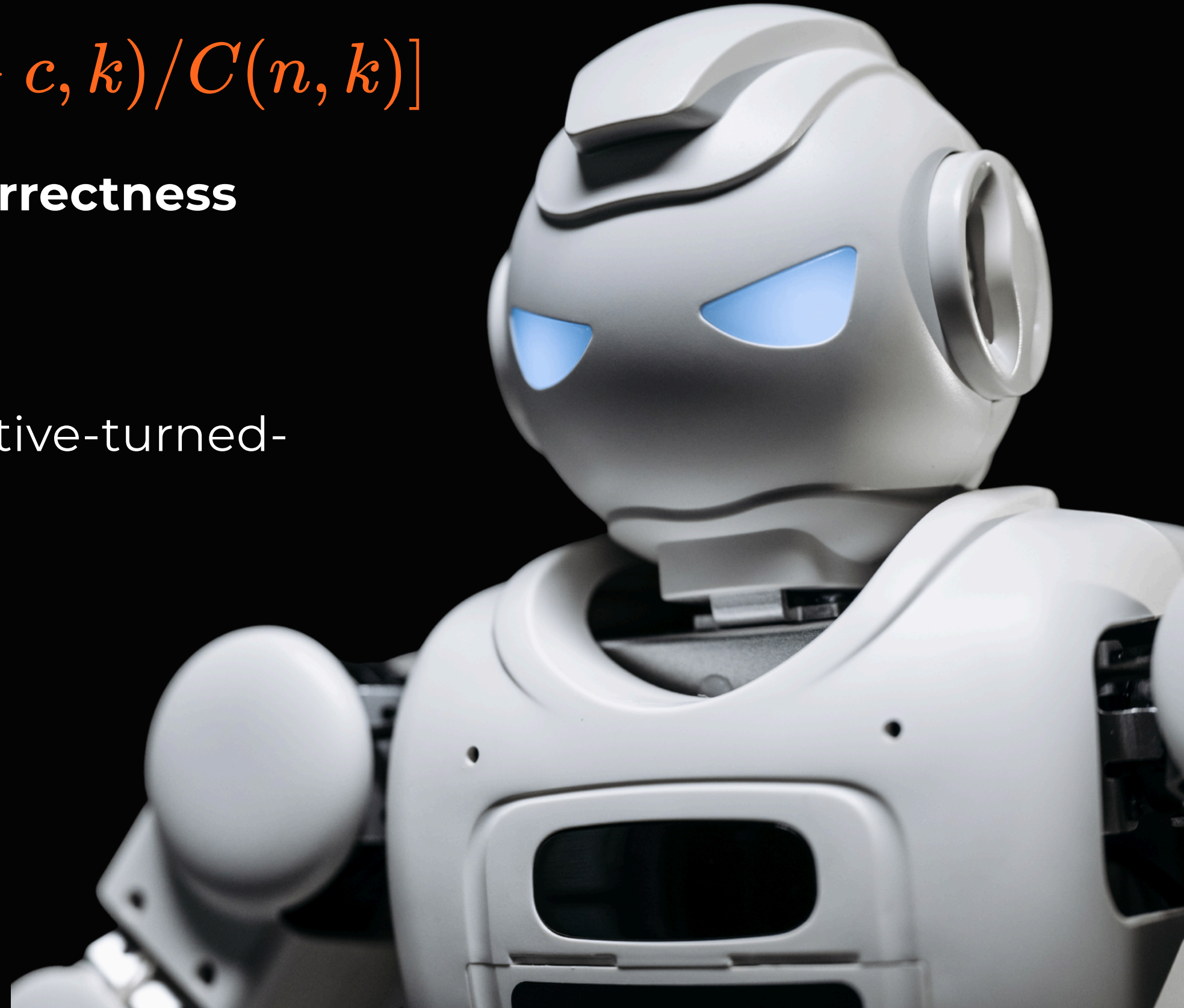
Also:

- Token efficiency / productivity ratios
- 4-point executability scale for qualitative-turned-quantitative evaluation.

n = total samples generated per problem

c = number of correct samples

k = number shown to user



METAGPT VS CHATDEV



METAGPT VS CHATDEV



3.75 vs 2.25

Executability

124.3 vs 248.9

Tokens used per Line

0.83 vs 2.5

Revisions

METAGPT X - A PROJECT BUILT ON METAGPT



A REAL PLATFORM

DISCUSSION

CHINESE WHISPER



TELEPHONE GAME

if agents just pass free-form natural-language messages to each other in sequence, small hallucinations and misunderstandings compound at each hop so by the time the last agent acts, it's working off a distorted picture of the actual task

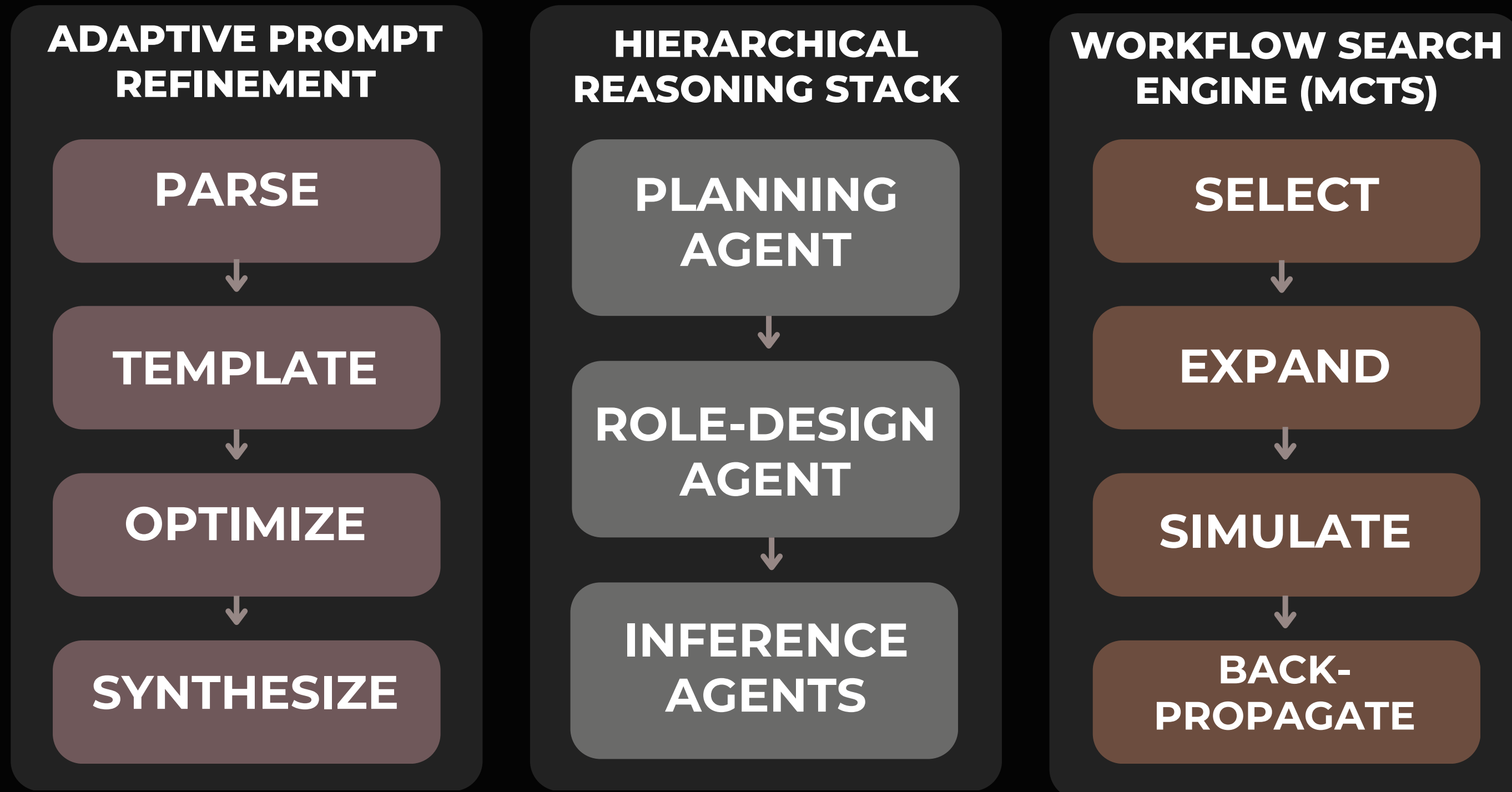
PAPER 03

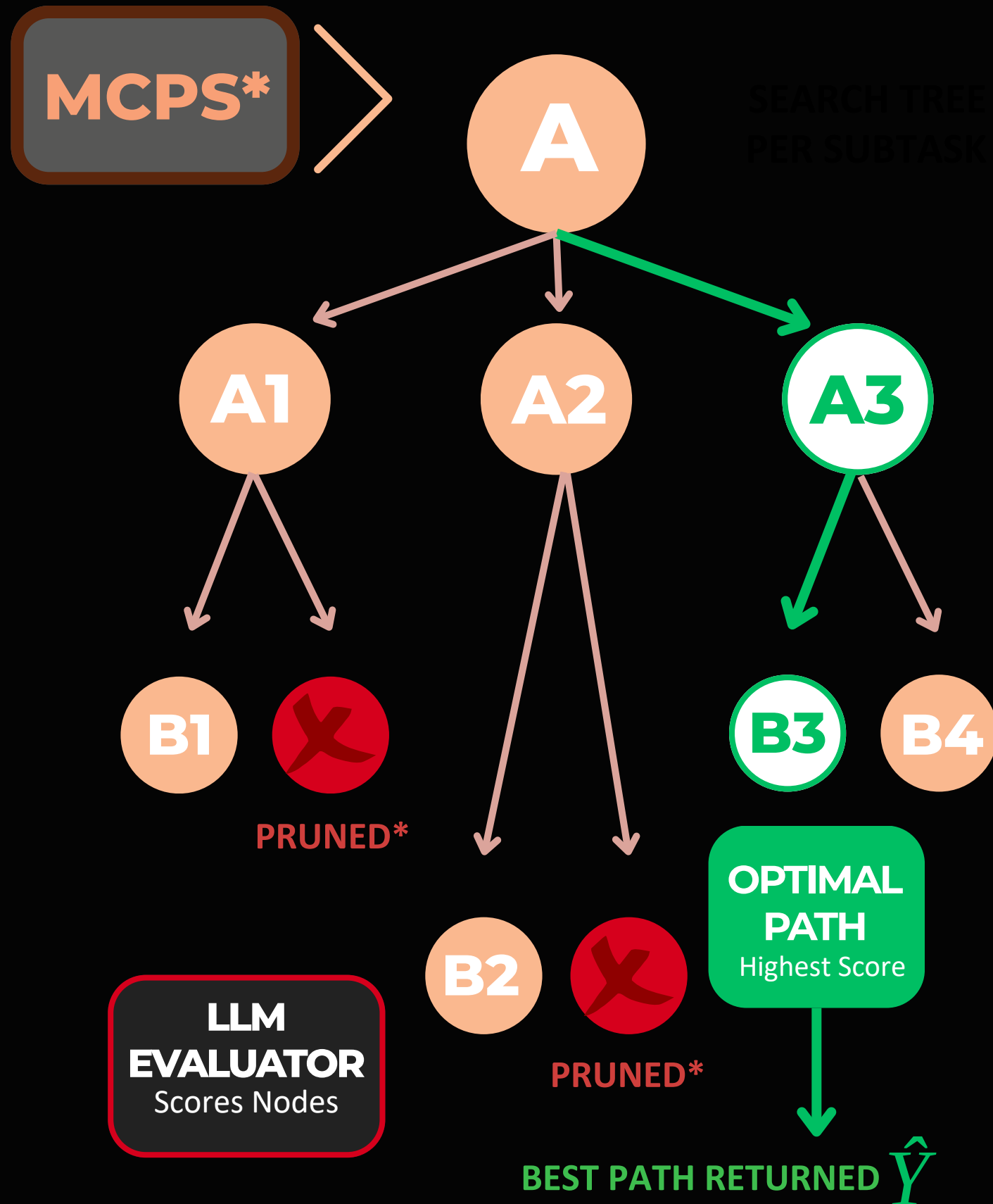
HALO: Hierarchical Autonomous Logic-Oriented Orchestration for Multi-Agent LLM Systems



ARCHITECTURE

PAPER 03





1. **Selection:** Pick the next node using UCB1:

$$Q(n)/N(n) + C \cdot \sqrt{(\ln N_{\text{parent}} / N(n))}$$

2. **Expansion:** Spawn a new agent at the frontier; add its output as a child node.

3. **Evaluation:** An LLM judge scores the output: quality score $v \in [0,1]$ + a pass/fail label.

4. **Backpropagation:** Push v up to every ancestor: $N(n) += 1$, $Q(n) += v$.

Good paths get chosen more next time.

Node = an agent-generated response or reasoning step.

Edge = a transition between reasoning states.

A root-to-leaf path = one candidate reasoning trajectory.



PAPER 03

HALO: Hierarchical Autonomous Logic-Oriented Orchestration for Multi-Agent LLM Systems

Algorithm 1 Algorithm of HALO

Input: User query Q , prompt refinement agents $\mathcal{P}_1 \sim \mathcal{P}_4$, planning agent $\mathcal{A}_{\text{plan}}$, role-design agent $\mathcal{A}_{\text{role}}$, inference agent pool $\mathcal{A}_k$, subtask budget K

Output: Final answer $\hat{Y}$

```
1:  $\mathcal{F} \leftarrow \mathcal{P}_1(Q)$                                 ▷ Parse task type  $\mathcal{T}$ , intent  $\mathcal{I}$ , and details  $\mathcal{D}$ 
2:  $Q_0 \leftarrow \mathcal{P}_2(Q, \mathcal{F})$                         ▷ Build initial prompt template
3:  $Q_{\text{opt}} \leftarrow \mathcal{P}_3(Q_0, \mathcal{F})$                 ▷ Optimize with prompting strategies
4:  $Q^* \leftarrow \mathcal{P}_4(Q_{\text{opt}}, \mathcal{F})$                 ▷ Synthesize final prompt
5:  $H_0 \leftarrow \emptyset$                                 ▷ Initialize subtask execution history
6: for  $k = 1$  to  $K$  do
7:    $T_k \leftarrow \mathcal{A}_{\text{plan}}(Q^*, \mathcal{F}, H_{k-1})$       ▷ Decompose query into subtask
8:   if  $T_k = \text{stop}$  then
9:     break                                           ▷ Terminate if planning agent signals task completion
10:  end if
11:   $\mathcal{A}_k \leftarrow \mathcal{A}_{\text{role}}(T_k, Q^*, \mathcal{F})$         ▷ Generate role-specific agents
12:  for each agent  $a_k^{(i)} \in \mathcal{A}_k$  do
13:     $y_k^{(i)} \leftarrow a_k^{(i)}(T_k, Q^*, \mathcal{F})$       ▷ Generate intermediate output by executing subtask  $T_k$ 
14:     $\ell_k^{(i)}, v_k^{(i)} \leftarrow \text{Evaluate}(y_k^{(i)})$   ▷ Evaluate status label and quality score for output
15:     $\text{MCTS\_Backpropagate}(a_k^{(i)}, \ell_k^{(i)}, v_k^{(i)})$   ▷ Propagate execution feedback
16:  end for
17:   $H_k \leftarrow H_{k-1} \cup \{T_k, \hat{Y}_k\}$               ▷ Update subtask execution history
18:  if  $\text{early-stop}(H_k)$  is True then
19:    break                                           ▷ Terminate if early-stopping mechanism is triggered
20:  end if
21: end for
22:  $\hat{Y} \leftarrow \text{Aggregate}(\{y_k^{(i)}\})$               ▷ Select final answer
23: return  $\hat{Y}$ 
```

DISCUSSION