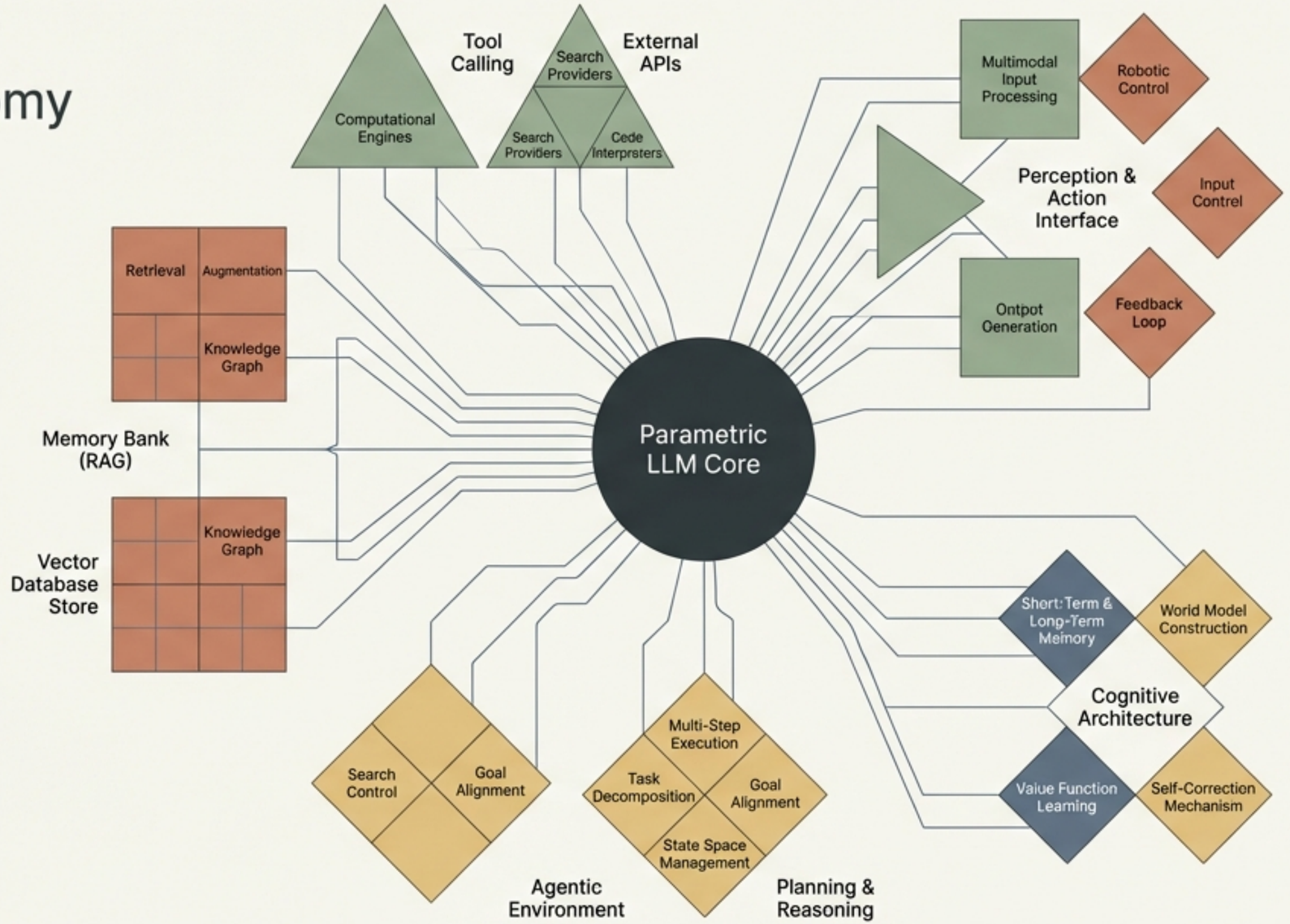
Augmented Large Language Models: The Evolution of Autonomy

A Comprehensive State-of-the-Art Treatment of RAG, Tool Calling, and Agents.



Inherent Constraints of Parametric Language Models

Knowledge Cutoff

$P_\theta(y \mid x)$ is conditioned only on training corpus D_{train} with temporal bound t_{max}.

Hallucination

Model assigns $P_\theta(y \mid x) > \epsilon$ to factually incorrect y due to distributional memorisation artifacts.

Computation Bottleneck

All reasoning must occur within fixed forward-pass depth L and hidden dimension d; no external symbolic computation.

Static Knowledge

Parameters θ encode a snapshot; updating requires retraining with immense cost.

Groundedness Gap

No mechanism to verify generated claims against authoritative sources at inference time.

The Paradigm Shift to Augmented Large Language Models

Vanilla LLM:

$$P_\theta(\mathbf{y} \mid \mathbf{x}) = \prod P_\theta(\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{x})$$

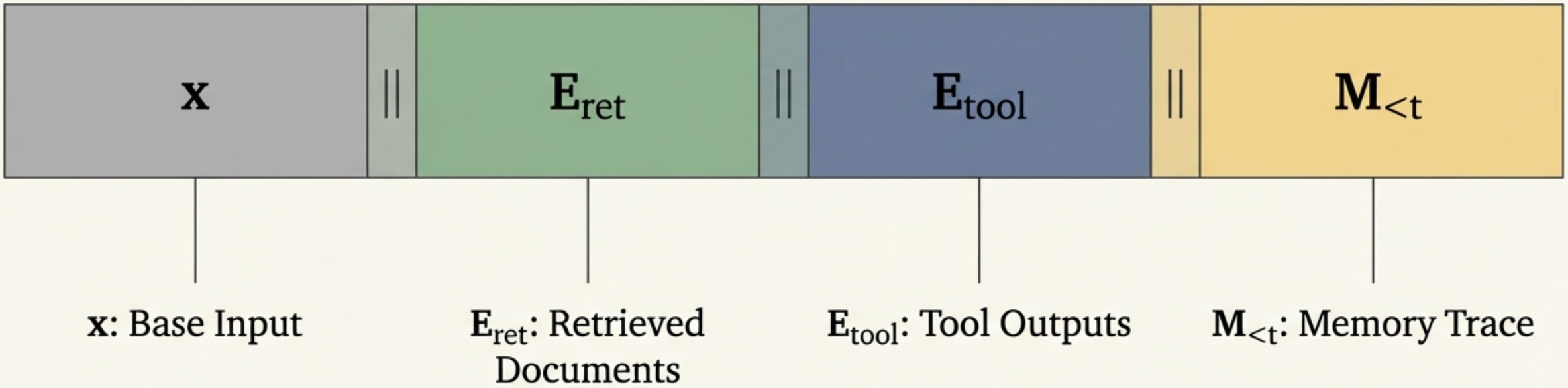
Augmented LLM (ALM):

$$P_S(\mathbf{y} \mid \mathbf{x}) = \prod P_\theta\Big(y_t \mid y_{<t}, \mathbf{x}, \bigoplus E_i(\phi_i(\mathbf{x}, y_{<t}))\Big)$$

- $\phi_i(\cdot)$: Query formulation function constructing a request to external module E_i.
- $\bigoplus$: Aggregation operator (concatenation, cross-attention fusion, interleaving, or gated mixture).
- E_i : Non-parametric or external functional modules (retriever, tool API, code interpreter, memory buffer).

Constructing the Augmented Context Window

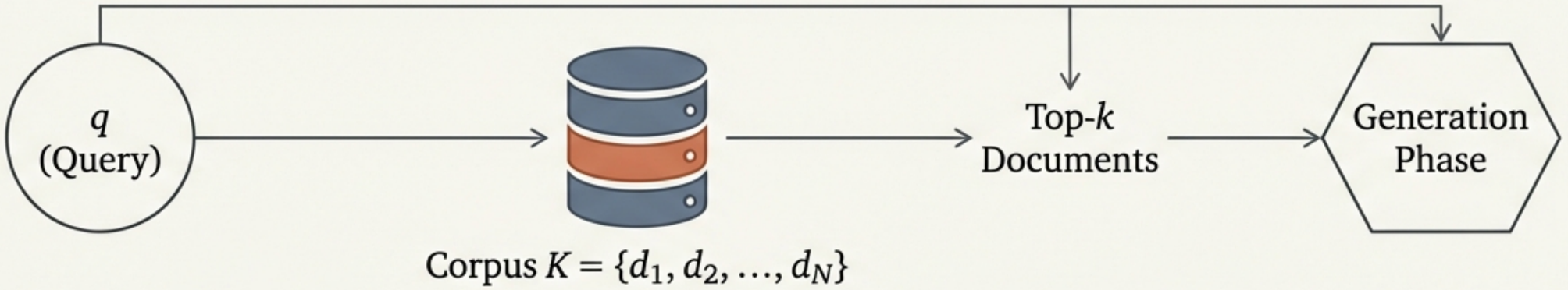
$$\mathbf{C}_{\text{aug}}(\mathbf{x}, t) = \left[\mathbf{x} \parallel \mathbf{E}_{\text{ret}}(\phi_{\text{ret}}(\mathbf{x}, \mathbf{y}_{<t})) \parallel \mathbf{E}_{\text{tool}}(\phi_{\text{tool}}(\mathbf{x}, \mathbf{y}_{<t})) \parallel \mathbf{M}_{<t}\right]$$



Generation at each step becomes: $y_t \sim P_\theta(y_t \mid \mathbf{C}_{\text{aug}}(\mathbf{x}, t))$

Stage 1: Bridging the Groundedness Gap via Retrieval

Generation is conditioned dynamically on external documents retrieved at inference time.



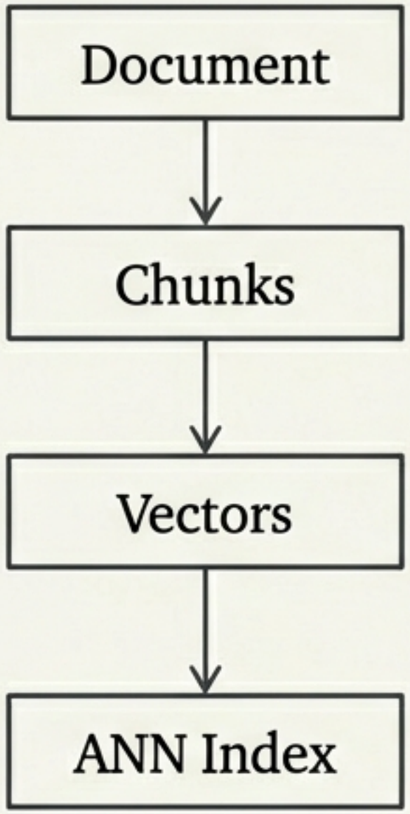
Top-k Marginalisation

$$P_{\text{RAG}}(y \mid q) \approx \sum \frac{\exp(sim(q, d)/\tau)}{\sum \exp(sim(q, d')/\tau)} \cdot P_\theta(y \mid q, d)$$

τ acts as the temperature parameter controlling retrieval distribution sharpness.

The Mechanics of the RAG Engine: Indexing and Retrieval

Offline Indexing Flow

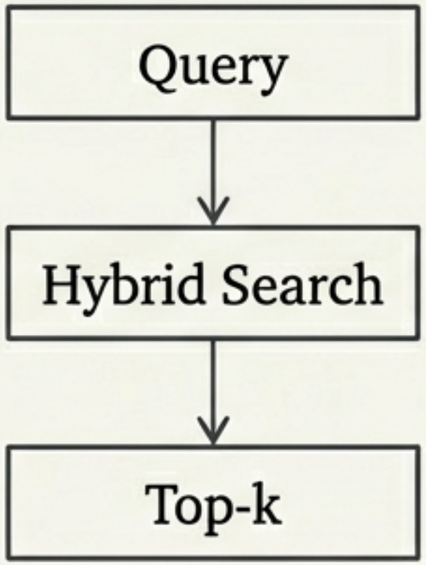


Chunking strategy:
$s_{i+1} = s_i + w - o$

Bi-encoder Embedding:
$v_i = f_d(c_i) \in \mathbb{R}^{d_{emb}}$

Sublinear Search:
$\text{Search}(I, q, k) \in O(\log N)$

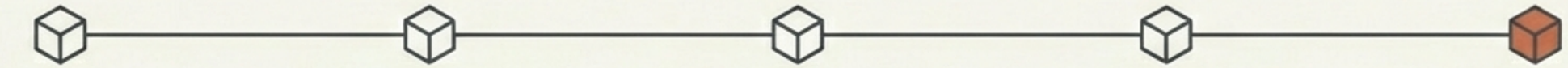
Inference Retrieval Flow



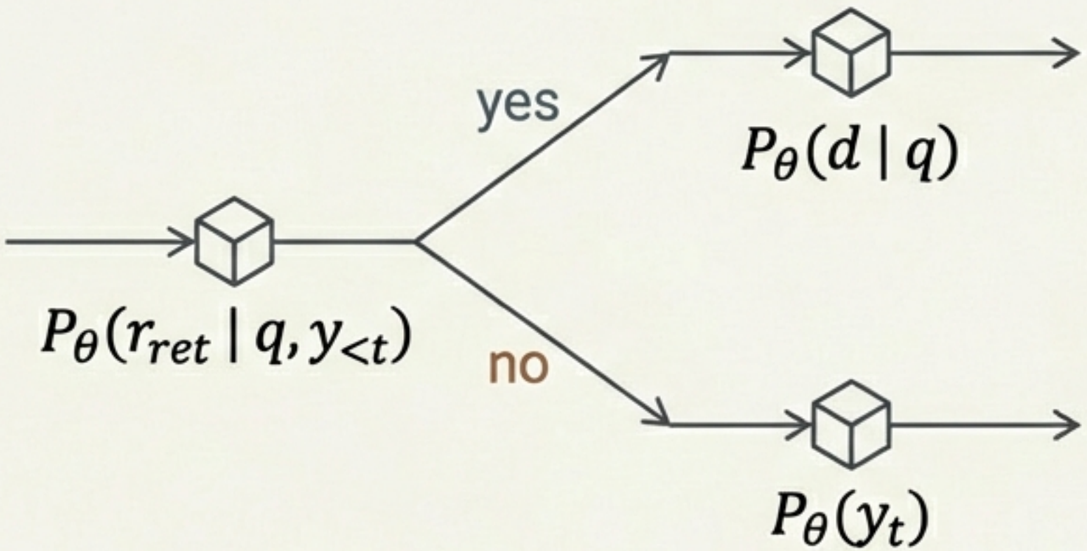
Hybrid Retrieval:

$$score_{hybrid}(q, d) = \alpha \cdot sim_{dense}(q, d) + (1 - \alpha) \cdot BM25_{norm}(q, d)$$

Evolving RAG through Self-Reflection and Modularity



Self-RAG Conditional Logic:



$$P_{Self-RAG}(y, r \mid q) = \prod_{\theta} P_\theta(r_{ret} \mid q, y_{<t}) \cdot \begin{cases} P_\theta(d \mid q) \cdots & \text{if } r_{ret} = \text{yes} \\ P_\theta(y_t \mid q, y_{<t}) & \text{if } r_{ret} = \text{no} \end{cases}$$

Reflection Tokens:	
r_{ret}: Retrieve? (yes/no)	r_{sup}: Support Level? (fully/partially/none)
r_{rel}: Relevant? (yes/no)	r_{use}: Utility? (1-5 score)

Multi-Dimensional Evaluation of Retrieval and Generation

Retrieval	Generation	End-to-End
Context Precision	Faithfulness (claims grounded in context)	Answer Correctness (Semantic Similarity + Factual Overlap)
Context Recall	Answer Relevance	
MRR	Hallucination Rate (1 - Faithfulness)	
nDCG@k		

RAGAS Framework

$$RAGAS_{\_score} = \text{HarmonicMean}(Faithfulness, AnswerRelevance, ContextPrecision, ContextRecall)$$

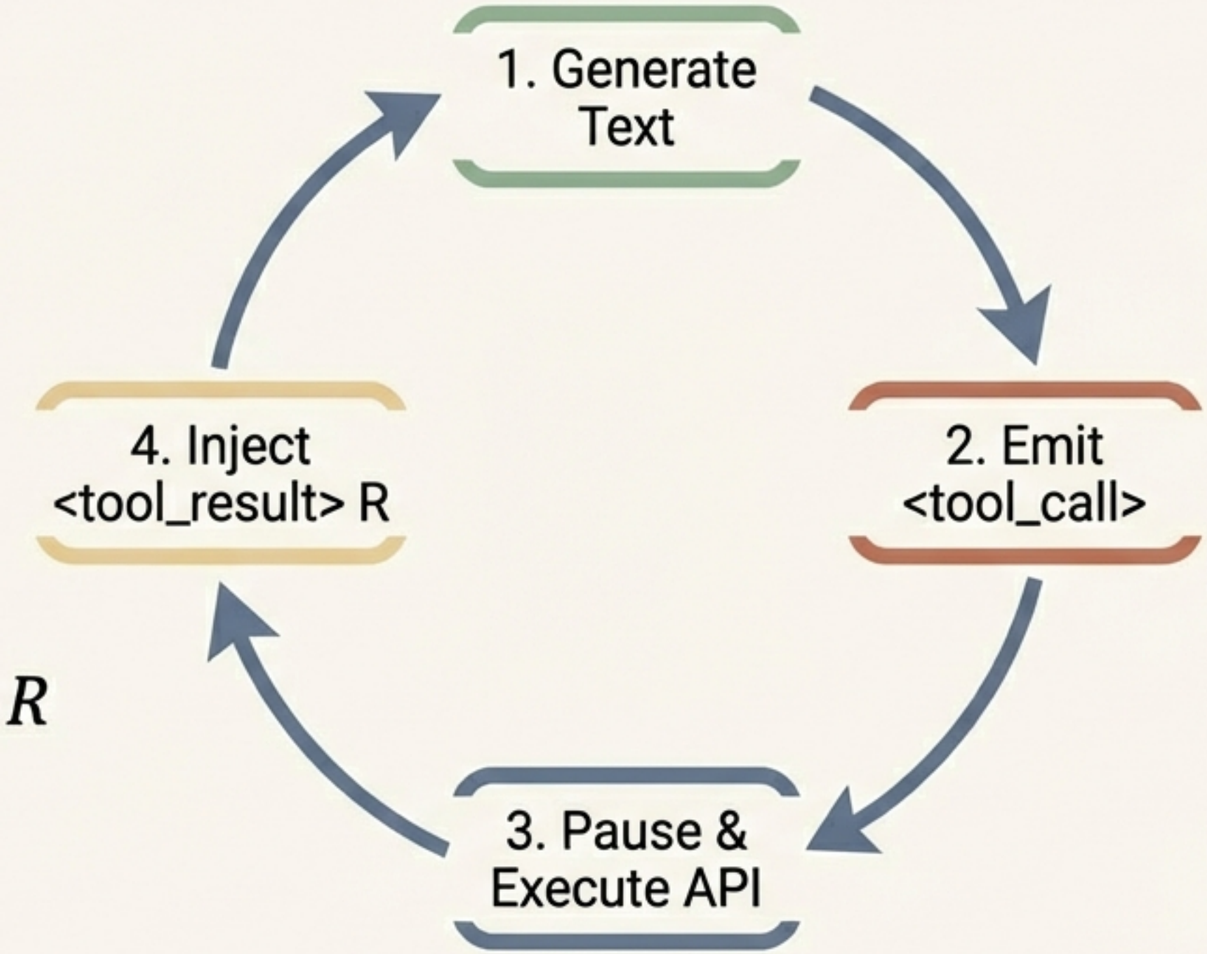
Stage 2: Enabling Computation via Tool Calling

Vocabulary Augmentation

$$V_{aug} = V \cup V_{tool}$$

Conditional Loop

$$y_t \sim \begin{cases} P_\theta(y_t \mid y_{<t}, x, R) \text{ if } y_{t-1} \notin V_{tool} \\ \textsc{Execute}(\text{parse\_tool\_call}(y_{<t})) \rightarrow R \\ \text{if } y_{t-1} = \text{</tool\_call>} \end{cases}$$



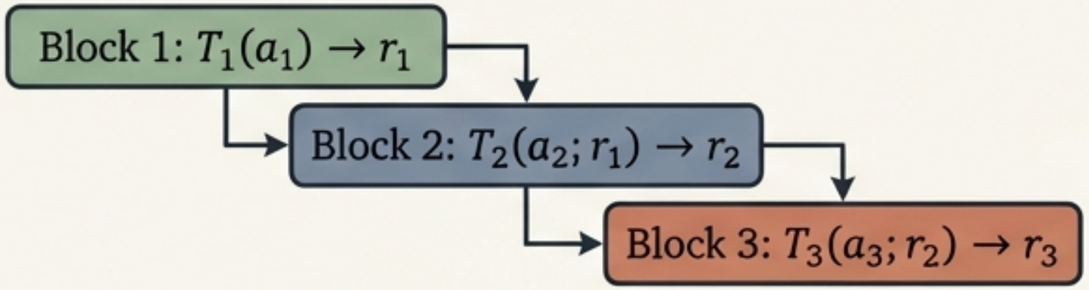
Structuring and Optimising Tool Execution

Tool Specification Tuple:

$T = (\text{name, description, parameters, returns})$

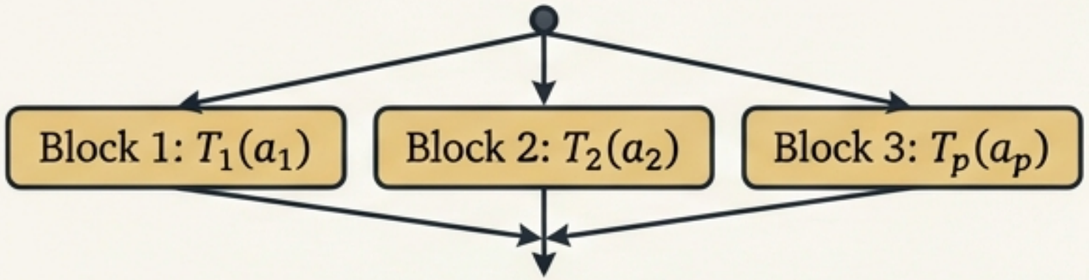
- name
- description
- parameters
- returns

Sequential Execution



$$y = M_\theta(x, r_1 = T_1(a_1), r_2 = T_2(a_2; r_1), \dots)$$

Parallel Execution



$$\{r_1, \dots, r_p\} = \text{PARALLEL\_EXEC}(T_1(a_1), \dots, T_p(a_p))$$

Training Paradigm: Tool-use is learned via Supervised Fine-Tuning ($\mathcal{L}_{\text{SFT}}$) strictly on model-generated tokens, or DPO optimising for correct tool trajectories.

Stage 3: LLM Agents as Self-Directing Cognitive Engines

An autonomous system

$$A = (M_\theta, P, T, E, S)$$

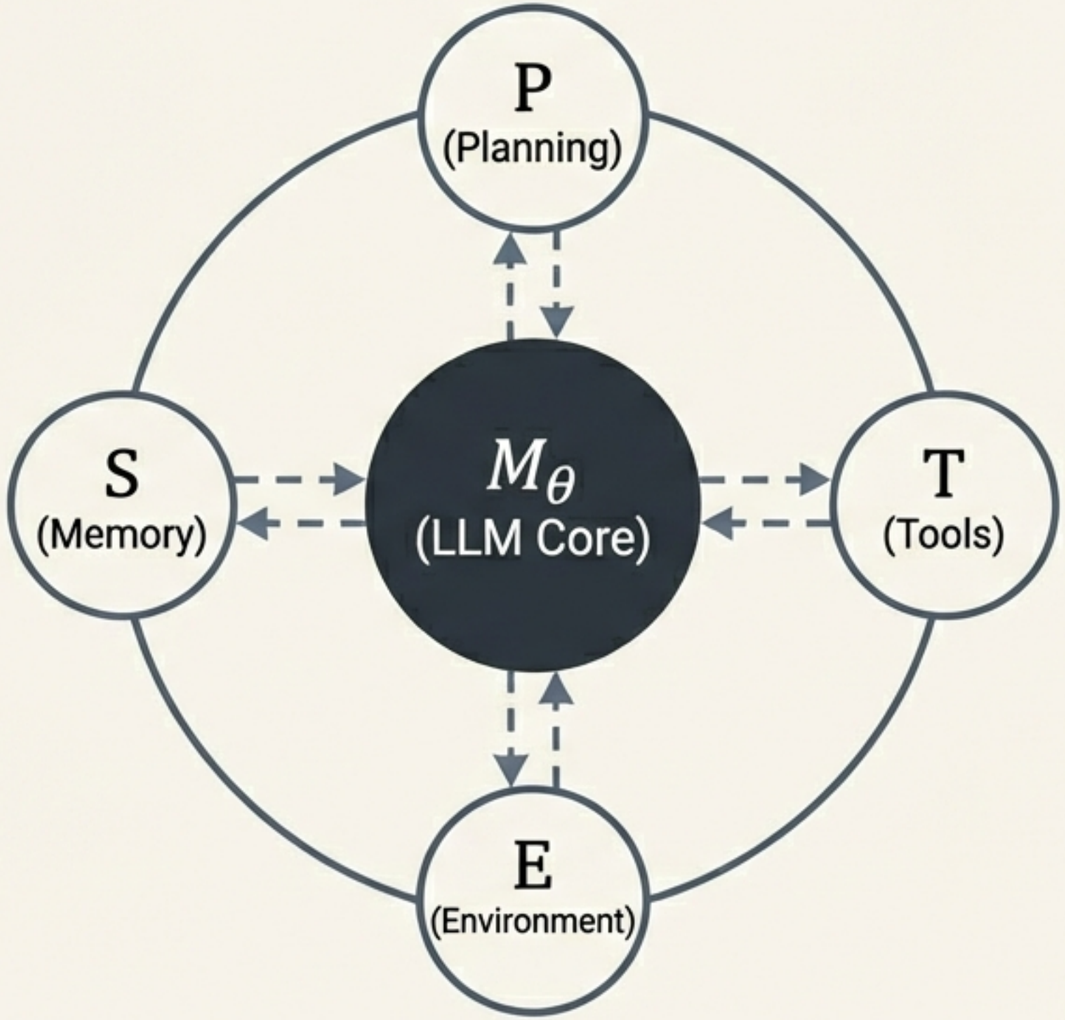
Action Mapping Equation:

$$a_t = \pi_\theta(o_{\leq t}, m_t, g) =$$

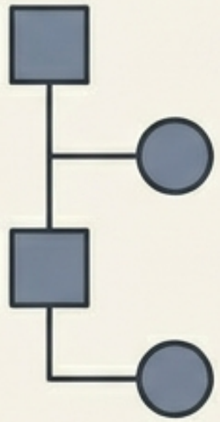
$$= M_\theta\Big(\text{PROMPT}(o_{\leq t}, m_t, g, T)\Big)$$

$o_{\leq t}$: observation history
m_t : memory state
g : goal
a_t : action (think, respond, tool_call, delegate)

Cognitive Architecture



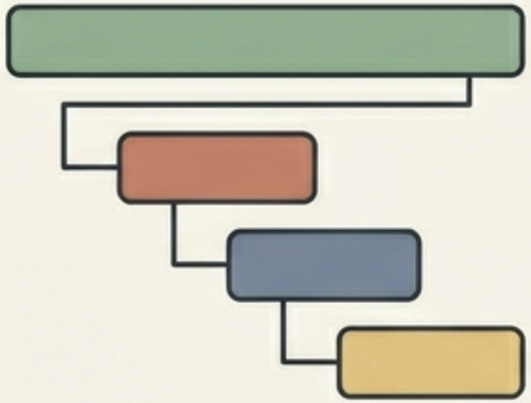
Architectures for Autonomous Agent Reasoning



1. ReAct

Interleaves reasoning and actions.

Trajectory = $(t_1, a_1, o_1, t_2, a_2 \ldots)$



2. Plan-and-Execute

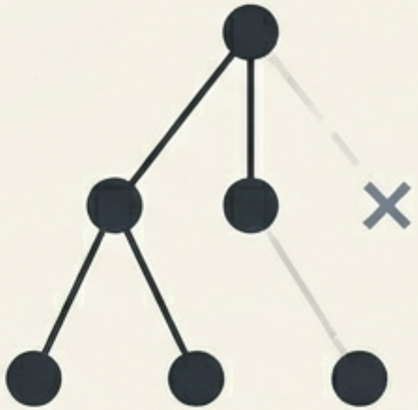
Separates planning phase from execution.

$G = \text{PLANNER}(M_\theta, q)$

3. Reflexion

Explicit self-evaluation loop.

$S_{memory} \leftarrow S_{memory} \cup \{Reflection_t\}$



4. Tree of Thoughts (ToT)

Explores multiple paths via search.

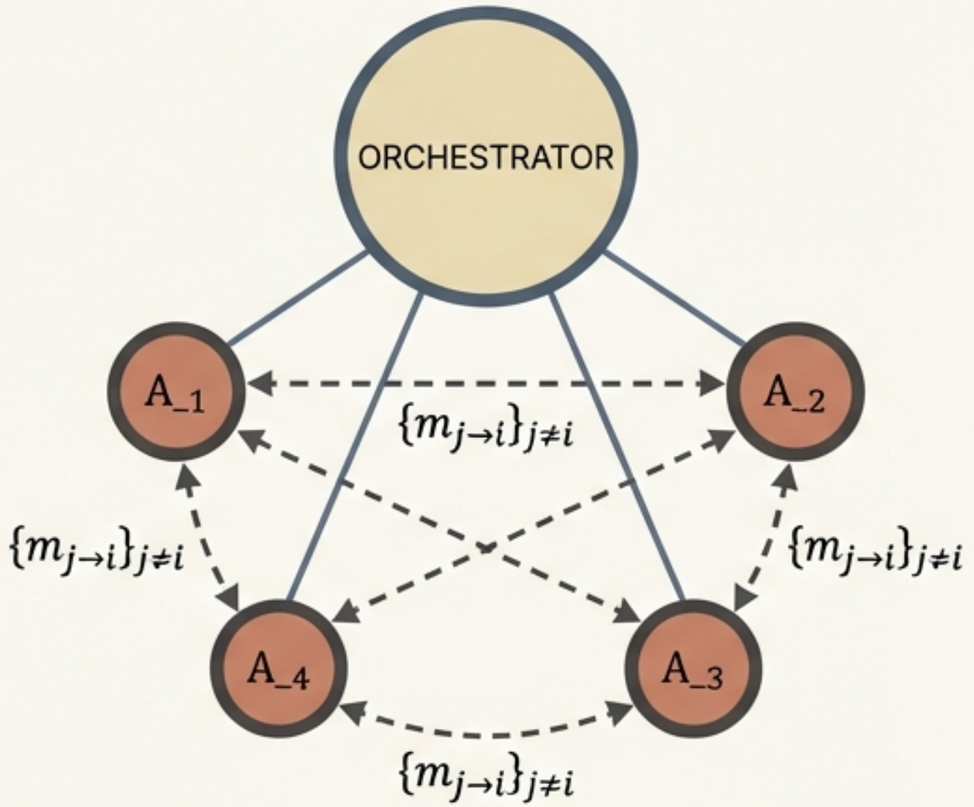
Prunes where $V(s) < \delta$

Sustaining Context and Scaling via Multi-Agent Systems

Memory Taxonomy

Working Memory	Context window (Per-session)
Episodic Memory	Action-observation traces (Cross-session)
Semantic Memory	Vector database of facts (Permanent)
Procedural Memory	Fine-tuned weights/prompts (Permanent)

Multi-Agent Coordination



$$y = \text{ORCHESTRATOR}\left(\{r_i\}_{i=1}^{n}\right)$$

$$\text{where } r_i = A_i\left(x_i, shared\_state, \{m_{j\to i}\}_{j\neq i}\right)$$

Quantifying Autonomous Performance and Efficiency

Task Success Rate

$$\frac{\text{Number of goals achieved}}{\text{Total number of goals}}$$

Average Steps to Completion

$$E[\text{ Length of valid trajectory } \tau_{success}]$$

Tool Call Accuracy

$$\frac{\text{Validly formatted tool executions}}{\text{Total tool calls emitted}}$$

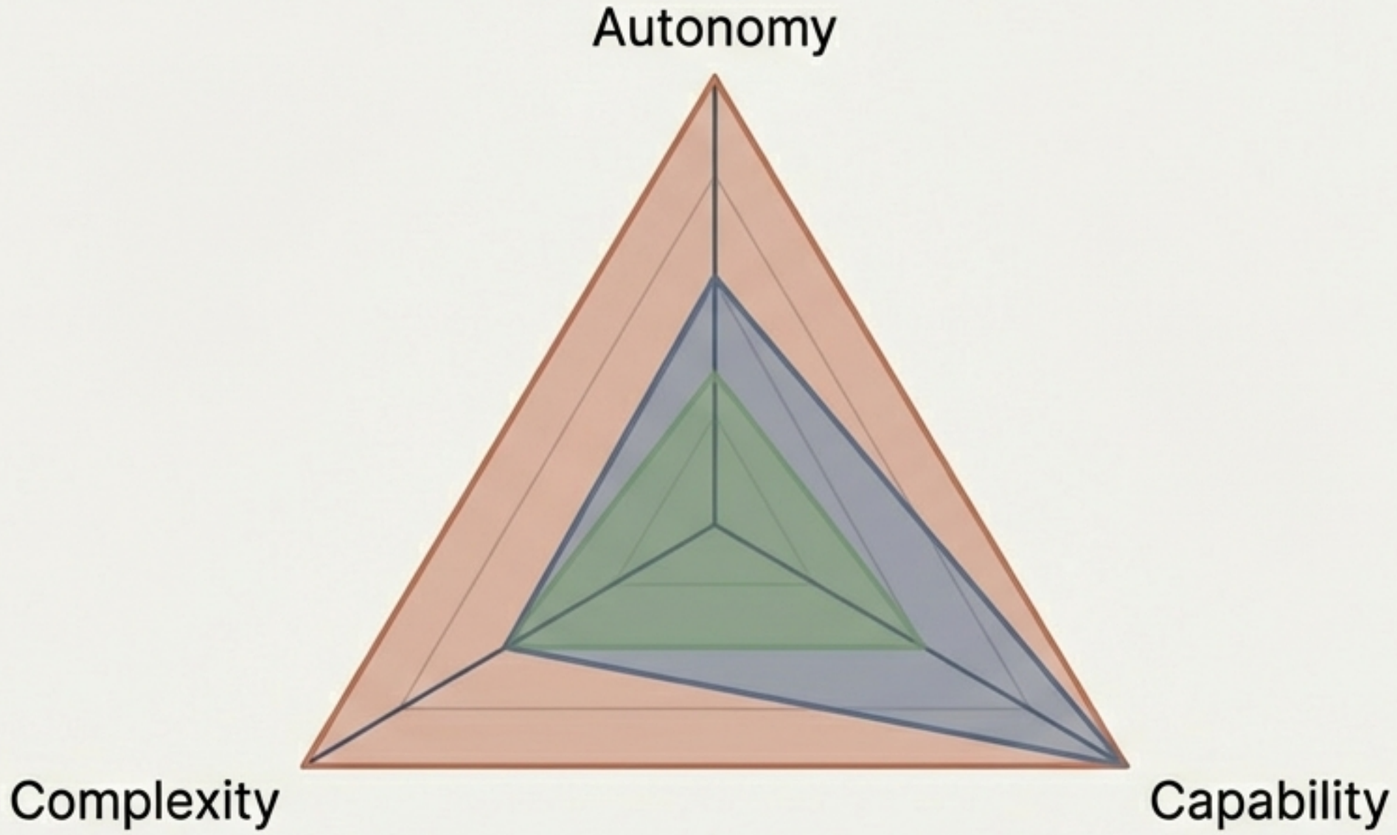
Reflection Improvement Rate

$$\frac{(\text{score}_{trial_{t+1}} - \text{score}_{trial\_t})}{\text{score}_{trial\_t}}$$

Cost Efficiency

$$\frac{\text{task\_success\_rate}}{\text{total\_tokens\_consumed}}$$

The Unified View of Augmented Architectures



RAG:
Low Autonomy
Moderate Capability
Moderate Complexity

Tool Calling:
Moderate Autonomy
High Capability
High Complexity

Agents:
High Autonomy
Maximum Capability
Maximum Complexity

The progression from Retrieval to Agents represents exponential gains in autonomy and capability, traded against engineering complexity. The optimal operating point is dictated by the specific requirements for task accuracy, inference latency, compute cost, and the required degree of deterministic trust.