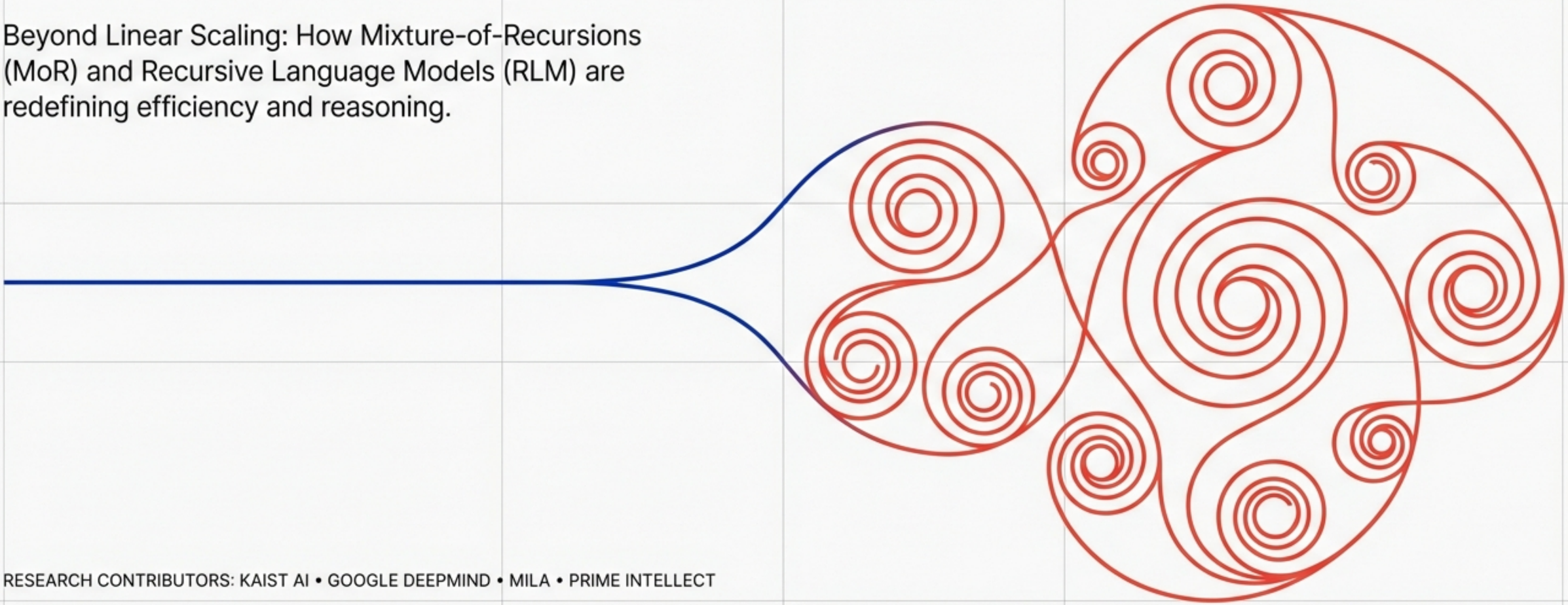
The Recursive Revolution

AI’s Paradigm Shift in 2026

Beyond Linear Scaling: How Mixture-of-Recursions (MoR) and Recursive Language Models (RLM) are redefining efficiency and reasoning.



RESEARCH CONTRIBUTORS: KAIST AI • GOOGLE DEEPMIND • MILA • PRIME INTELLECT

The Efficiency Wall

The Linear Scaling Era Has Hit a Ceiling

By 2025, simply making models bigger and contexts longer became unsustainable.
became unsustainable. Brute-force scaling is facing diminishing returns.

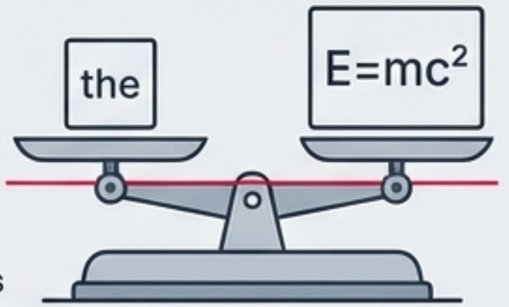
BOTTLENECKS

Context Rot

Performance degrades as context grows. Models lose coherence over thousands of tokens ("Lost in the Middle").

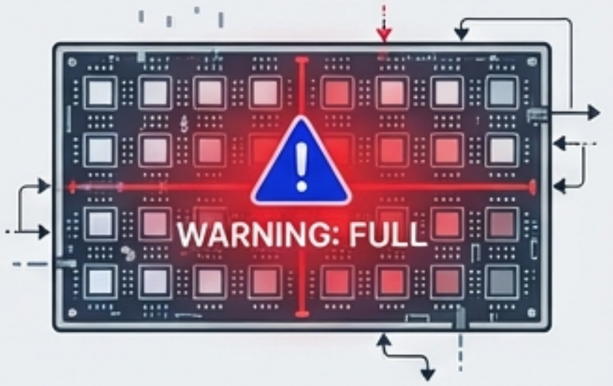
Compute Waste

Static compute budget. Standard Transformers spend equal compute on function words as they do on complex reasoning.



Memory Explosion

KV caches saturate GPU memory, limiting batch sizes and slowing inference.



Statistic: Per-token costs rise linearly with context, while performance creates a divergence between cost and capability.

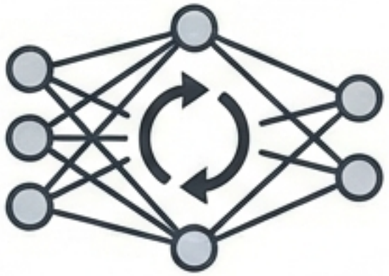
Introducing The Recursive Paradigm

Recursion Decouples Quality from Cost

The Solution: Recursion

Internal Recursion (Architecture)

Mixture-of-Recursions (MoR)



Reusing model weights to create variable 'thinking depth' per token.
Goal: Parameter efficiency.

External Recursion (Scaffolding)

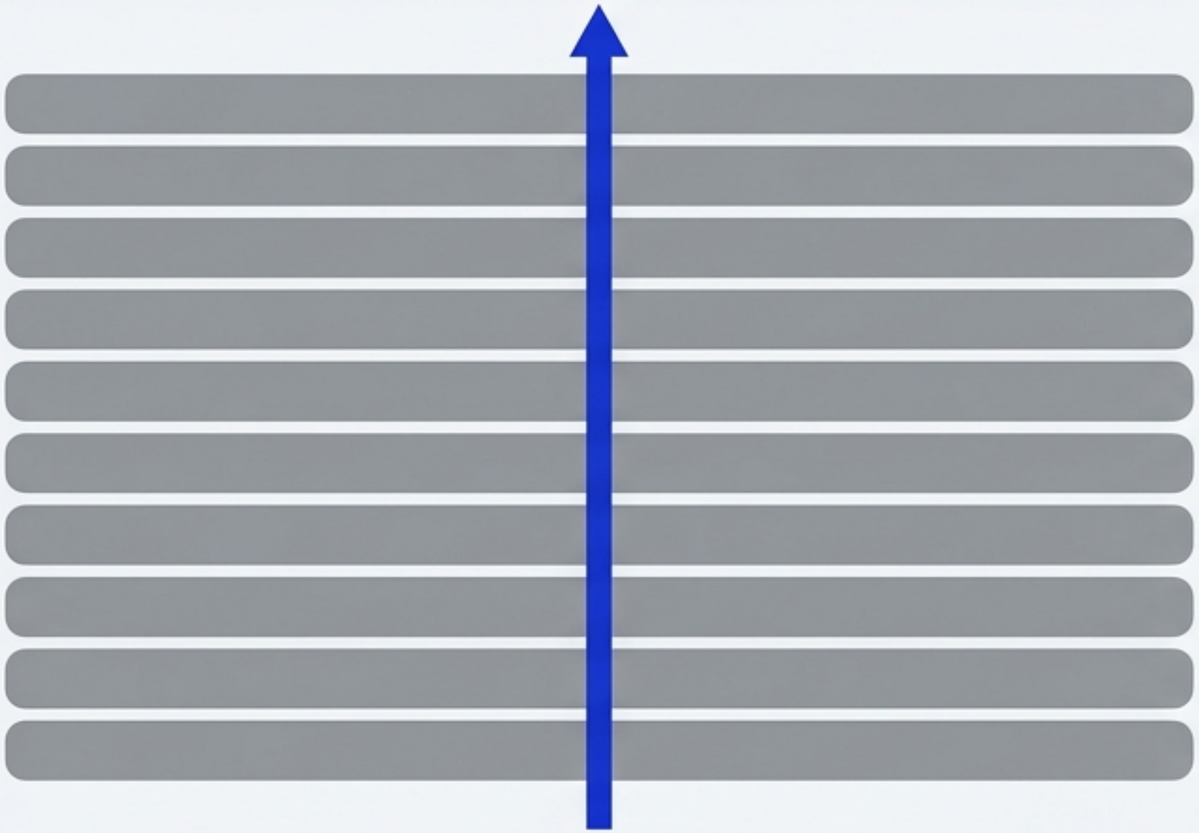
**Recursive Language Models (RLM)*

Using code and sub-agents to 'fold' context and manage long horizons.
Goal: Infinite context management.

Deep Dive I: Mixture-of-Recursions (MoR)

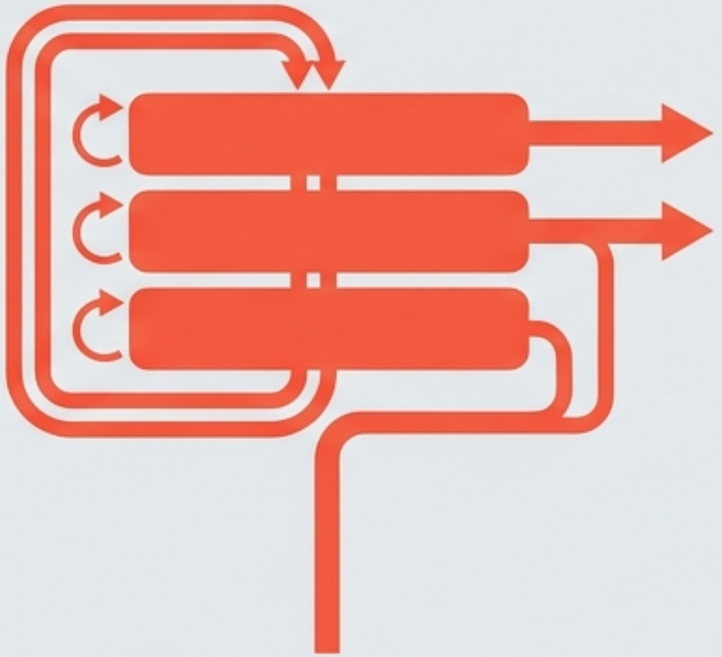
Big Model Quality without Big Model Cost

Standard Transformer



L distinct layers. Every token passes through every layer. Static depth.

Mixture-of-Recursions

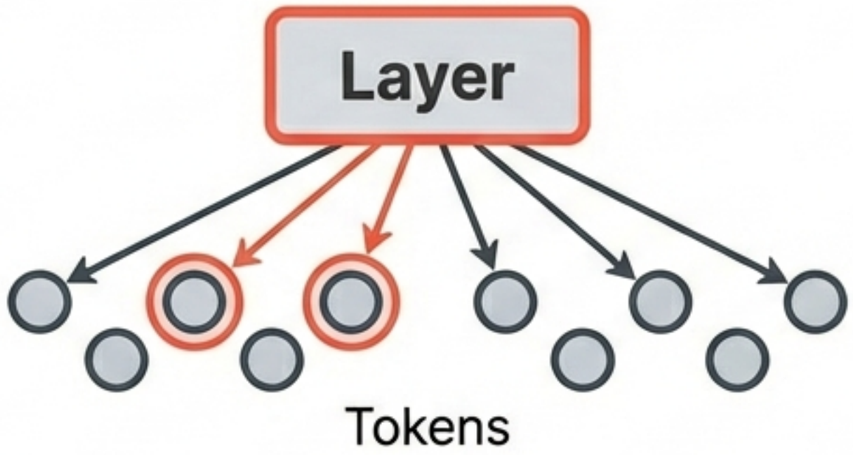


Shared layers (Φ') applied cyclically. Adaptive Depth: Easy tokens exit early; hard tokens loop more.

The Router: Learning Dynamic Thinking Depth

Treating model depth as a dynamic resource, not a static constraint.

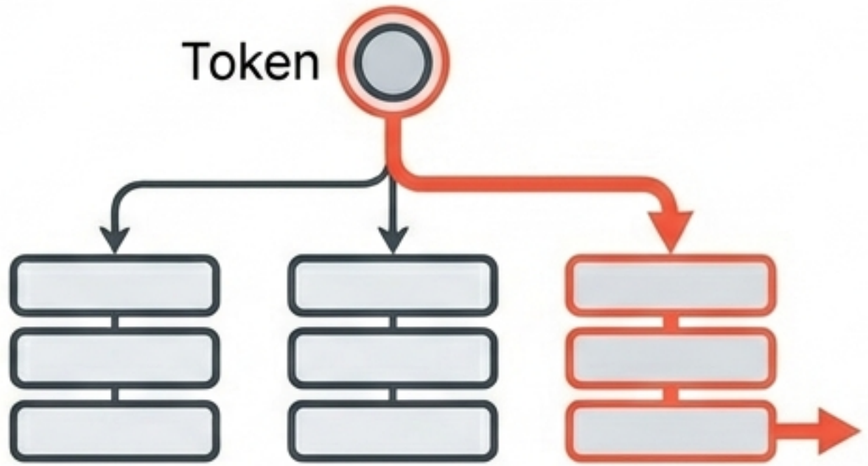
Expert-Choice Routing



The Layer chooses top-k tokens to process.

- Guarantees perfect load balancing.
- Risks information leakage (requires auxiliary loss).

Token-Choice Routing



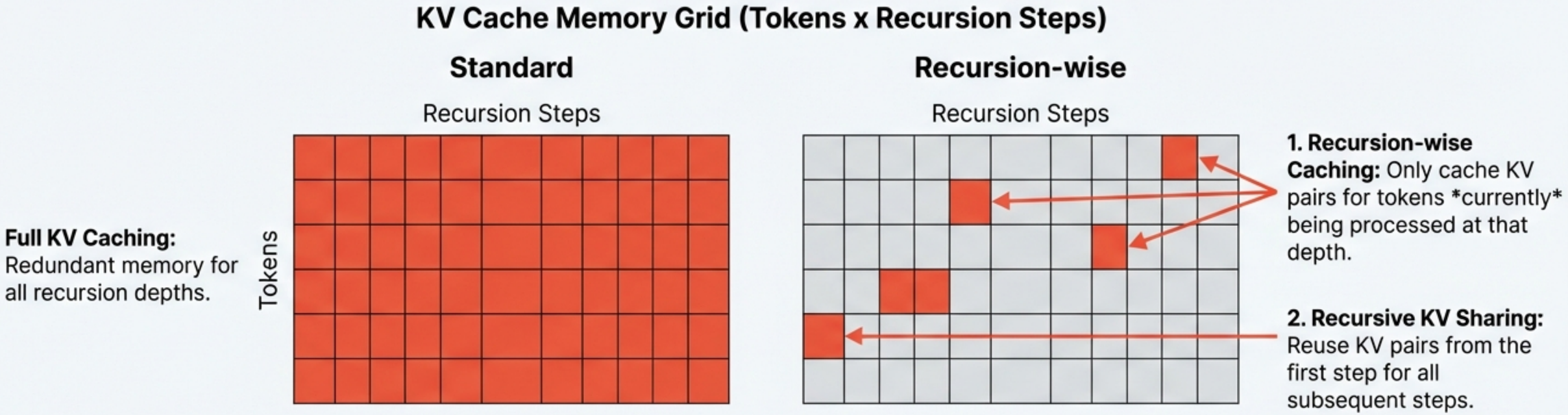
The Token decides its own path upfront.

- Preserves causality perfectly.
- Risks load imbalance.

Analysis: Semantic importance drives depth. Content-rich words like 'Drugs' trigger deeper recursion (3 steps). Function words like 'and' exit early (1 step).

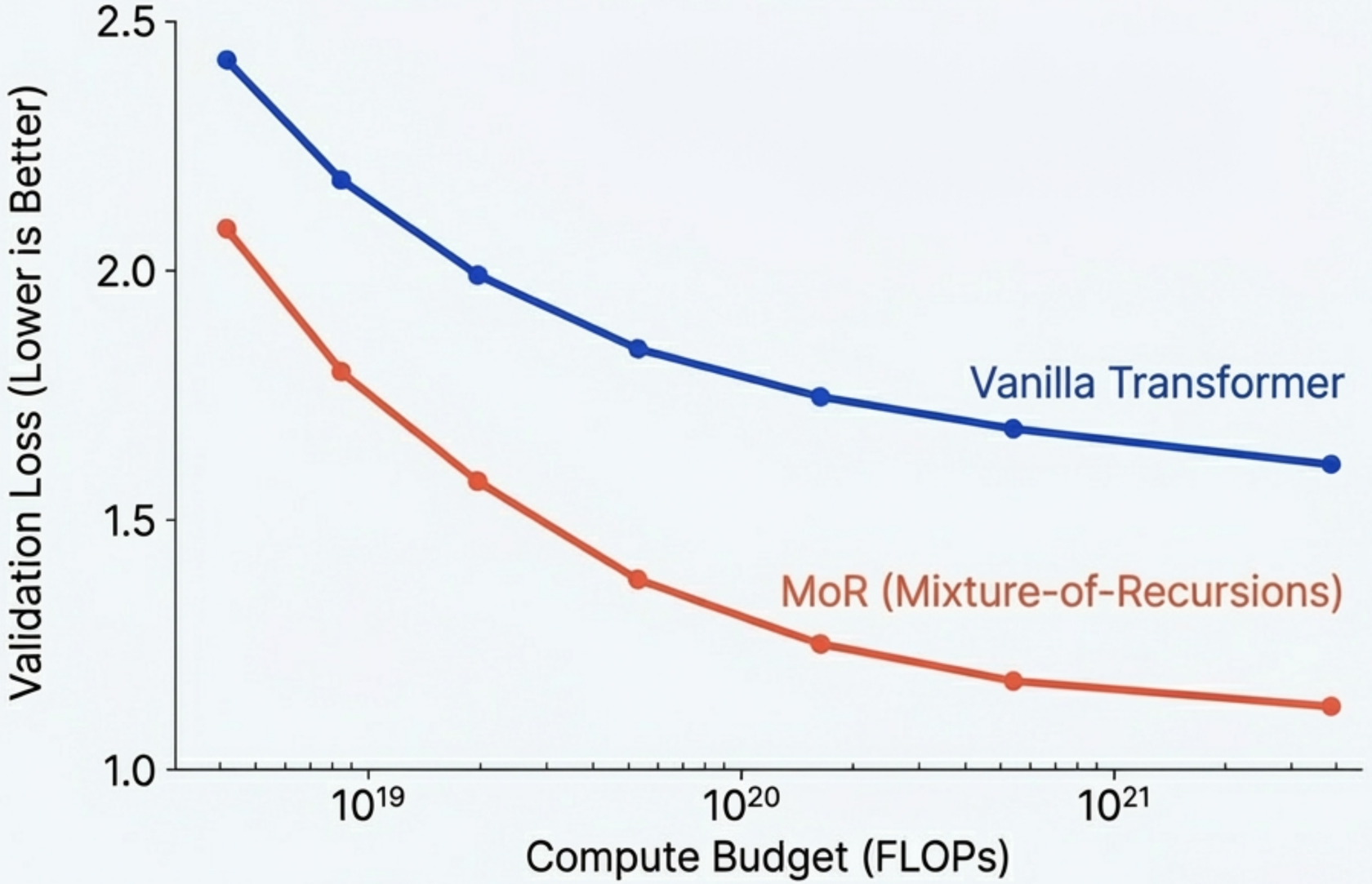
Efficiency Breakthrough: Recursion-wise KV Caching

Eliminating Redundant Memory Access



Up to 2.18x Higher Inference Throughput via continuous depth-wise batching.

MoR Establishes a New Pareto Frontier



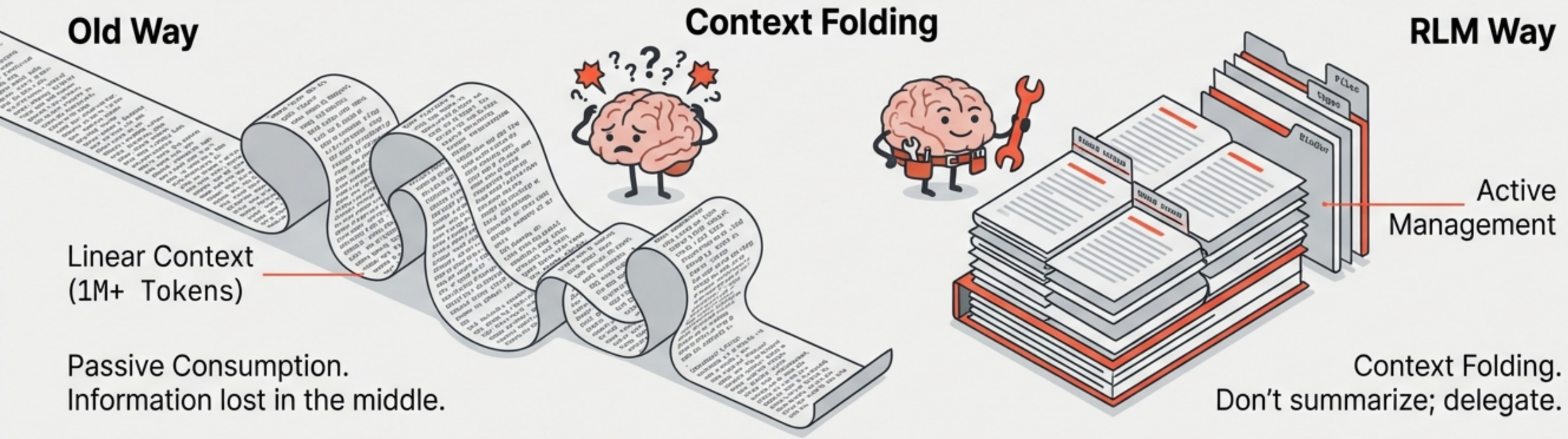
~50% Fewer Parameters
(due to weight tying)

Higher Few-Shot Accuracy

2.06x Speedup
(MoR-4 vs Vanilla)

Deep Dive II: Recursive Language Models (RLM)

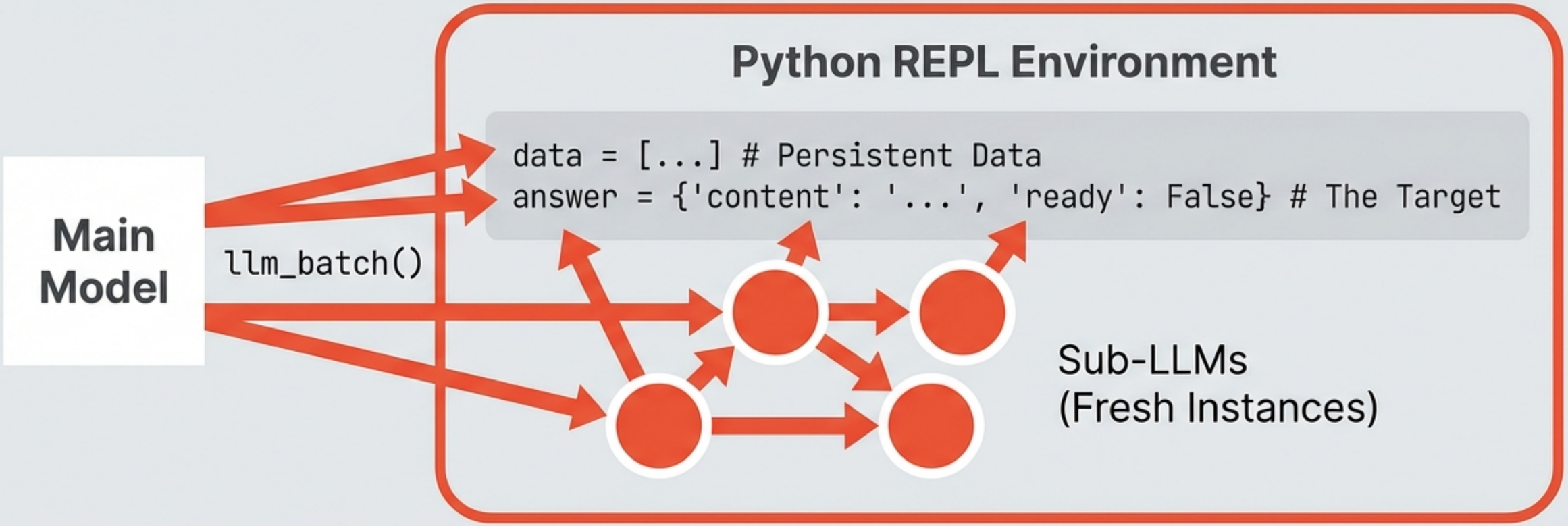
Solving Context Rot via Scaffolding



"RLM moves from optimizing weights
to optimizing **workflow."**

How RLM Works: The Persistent REPL

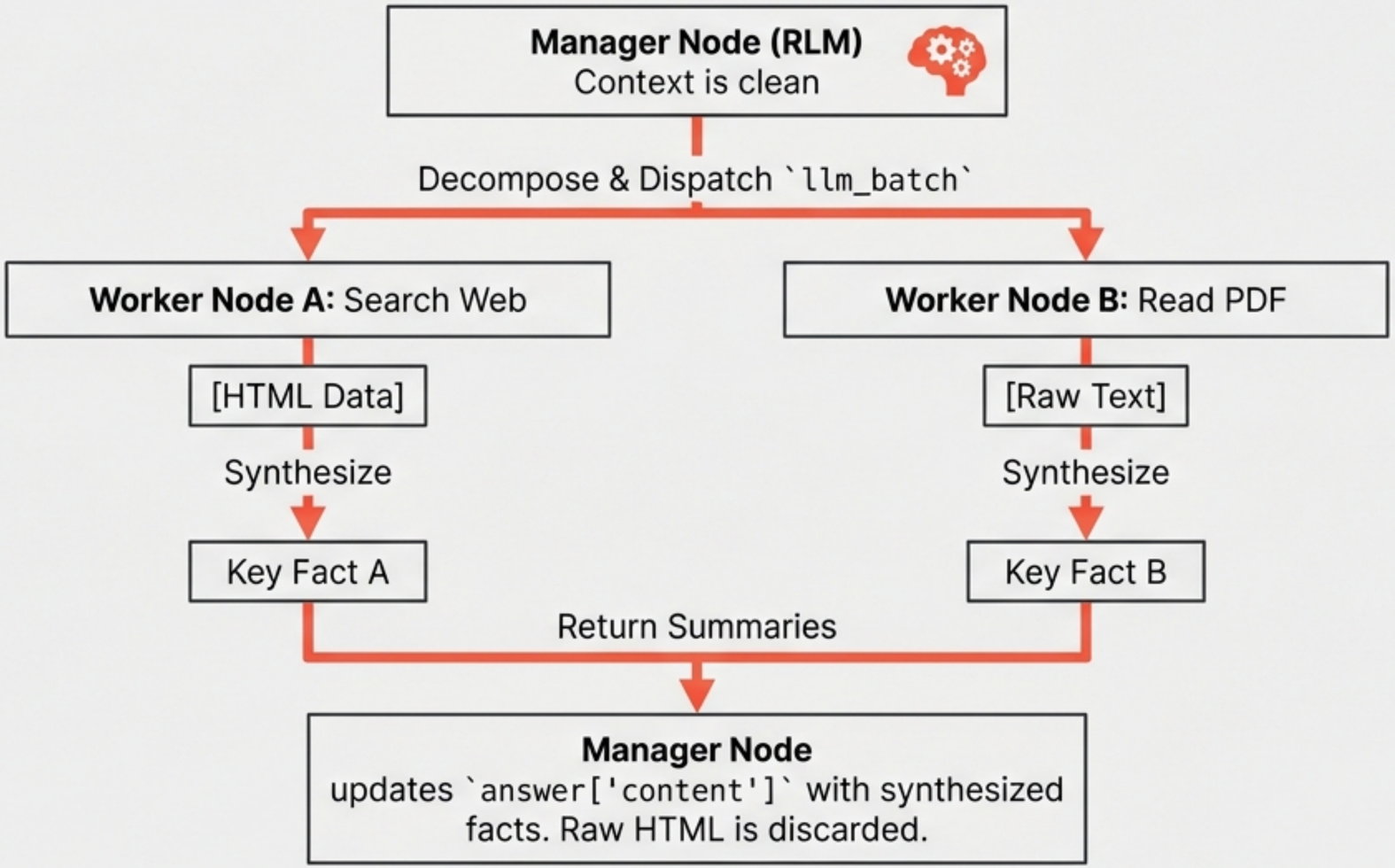
The model manages its own memory through code.



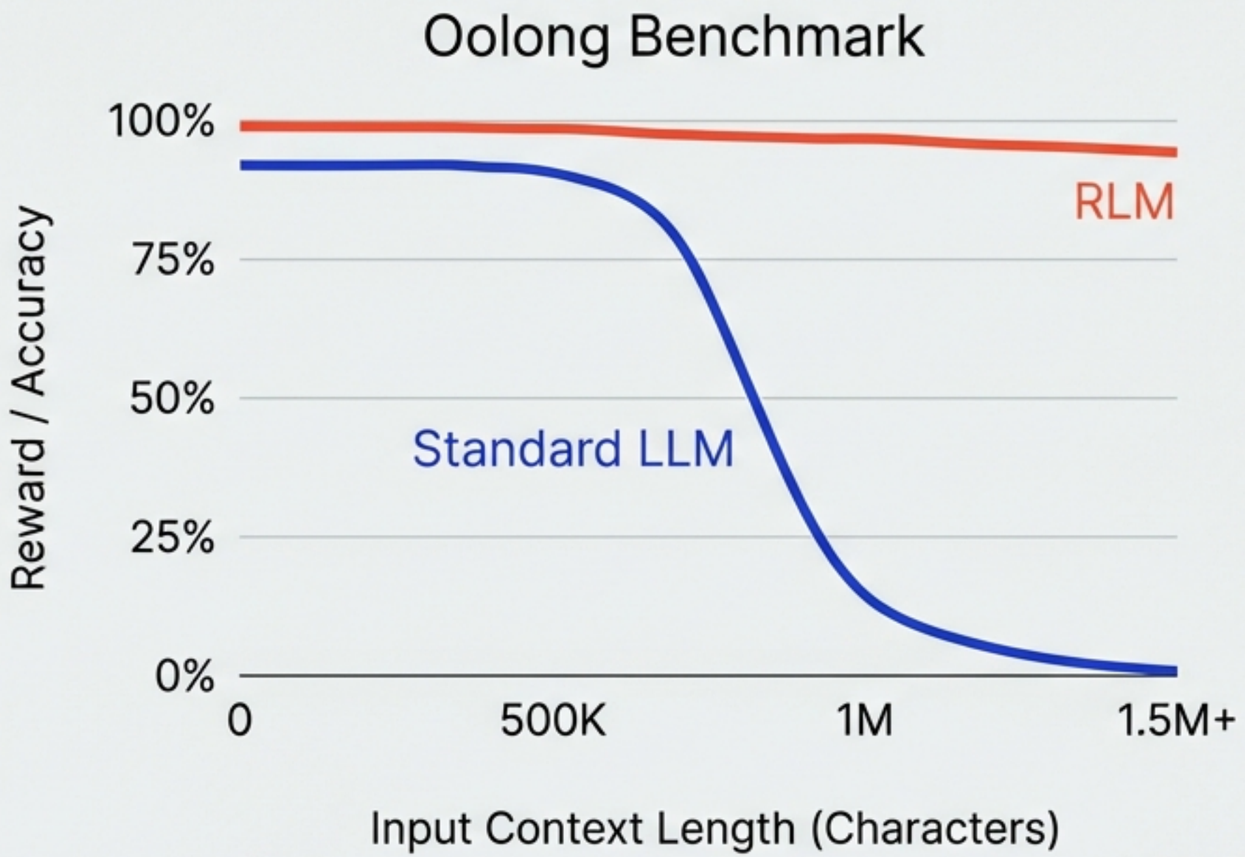
The model iterates on the 'answer' variable via code until 'ready' is True.
Tools are available only to Sub-LLMs to keep main context clean.

RLM in Action: Deep Research

Case Study: DeepDive Workflow



RLM Performance: Beating the Context Curve



- Outperforms on ‘Real’ Data (Unstructured, Complex)
- Maintains high reward while keeping Main Model context short.
- Impact of Tips: Environment-specific hints double the reward in complex tasks.

Synthesis: MoR vs. RLM

Two Sides of the Recursive Coin

Mixture-of-Recursions (MoR)	Recursive Language Models (RLM)
• Type: Internal / Architectural • Mechanism: Reusing Weights & Adaptive Depth • Optimization Target: **Processing Efficiency** (FLOPs & Parameters)	• Type: External / Scaffolding • Mechanism: Reusing Context & Sub-Agents • Optimization Target: **Memory Efficiency** (Infinite Context)

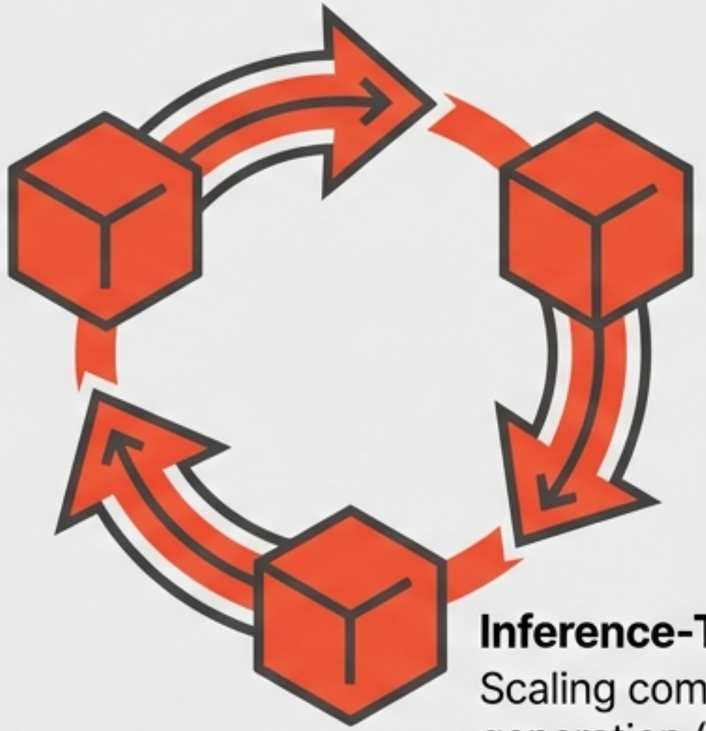
The Synergy: A true Recursive AI uses MoR architecture *inside* an RLM agent structure.

Computational Swiss Style

The Horizon: Recursive Self-Improvement (RSI)

From Thought Experiments to Deployed Loops (ICLR 2026 Vision)

Self-Rewriting Code
Agents that modify their own scaffolding.



Synthetic Data Pipelines
Loops generating data to fine-tune the generator.

Inference-Time Scaling
Scaling compute during generation (MoR).

The Goal: Loops that actually get better. Moving from "hand-waving" to measurable, reliable self-improvement.

Computational Swiss Style

The Paradigm of 2026: The Thinking Loop

Strategic Predictions

Training for Recursion

Models will be pre-trained specifically to use RLM scaffolds. RL will be used to teach models context management, not just next-token prediction.

Unified Architectures

MoR concepts (layer tying, adaptive depth) will become standard in large-scale foundation models to manage compute costs.

Agentic Workflows

‘Context’ is no longer a text window. It is a dynamic database (REPL) managed by the model via sub-agent dispatch.

Computational Swiss Style

The Future is Not Linear.

It is Recursive.

1. **MoR**: Thinking **deeper** on hard problems.
2. **RLM**: Working **longer** on hard tasks.

By reusing weights and actively managing memory, we unlock the next order of magnitude in AI capability—deeper, faster, and infinitely more capable.