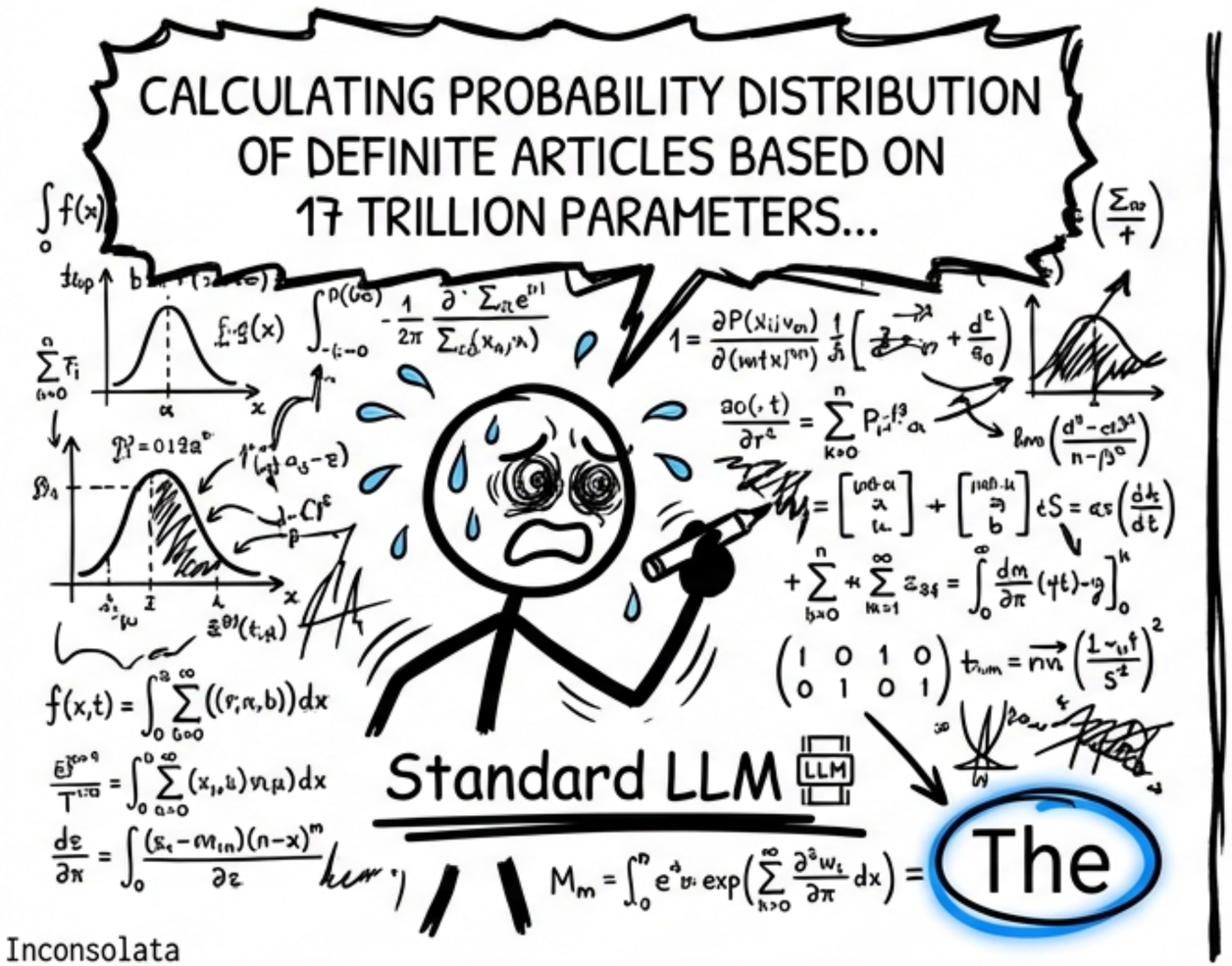
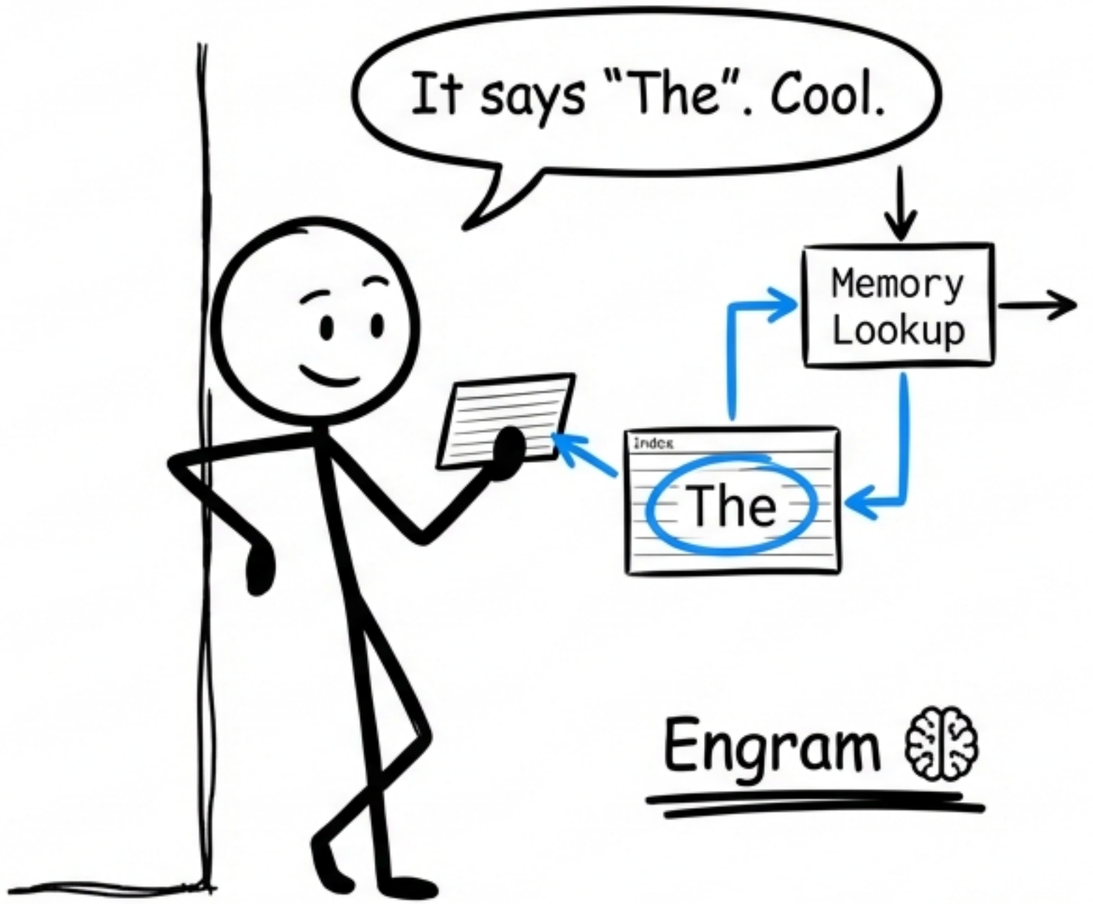
Stop Thinking, Start Remembering



Inconsolata

Left: The ‘Standard LLM’ approach: Brute force computation.



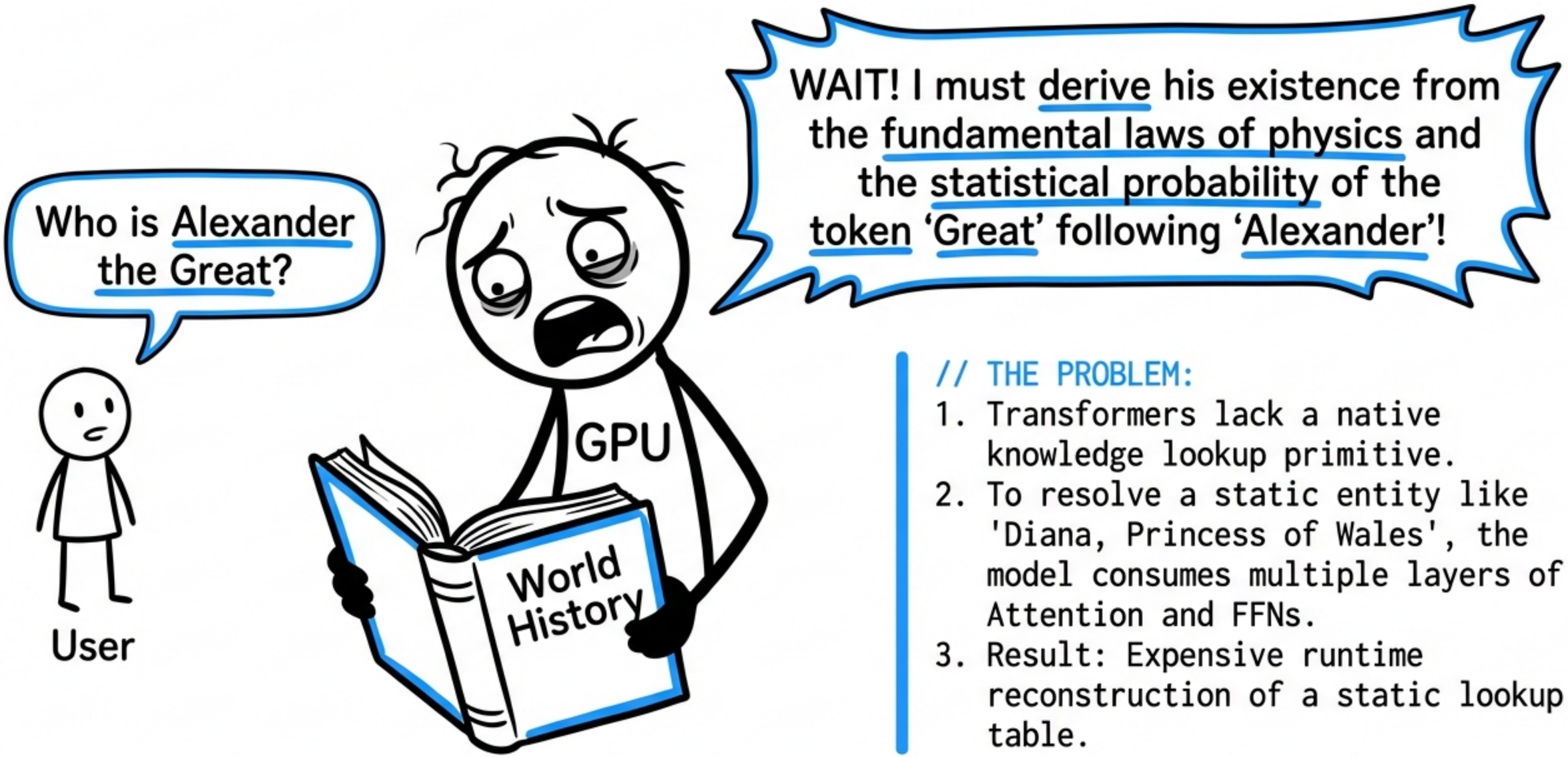
Inconsolata

Right: The ‘Engram’ approach: Instant retrieval.

The Engram Architecture: A Stick Figure’s Guide to Cheating at Intelligence

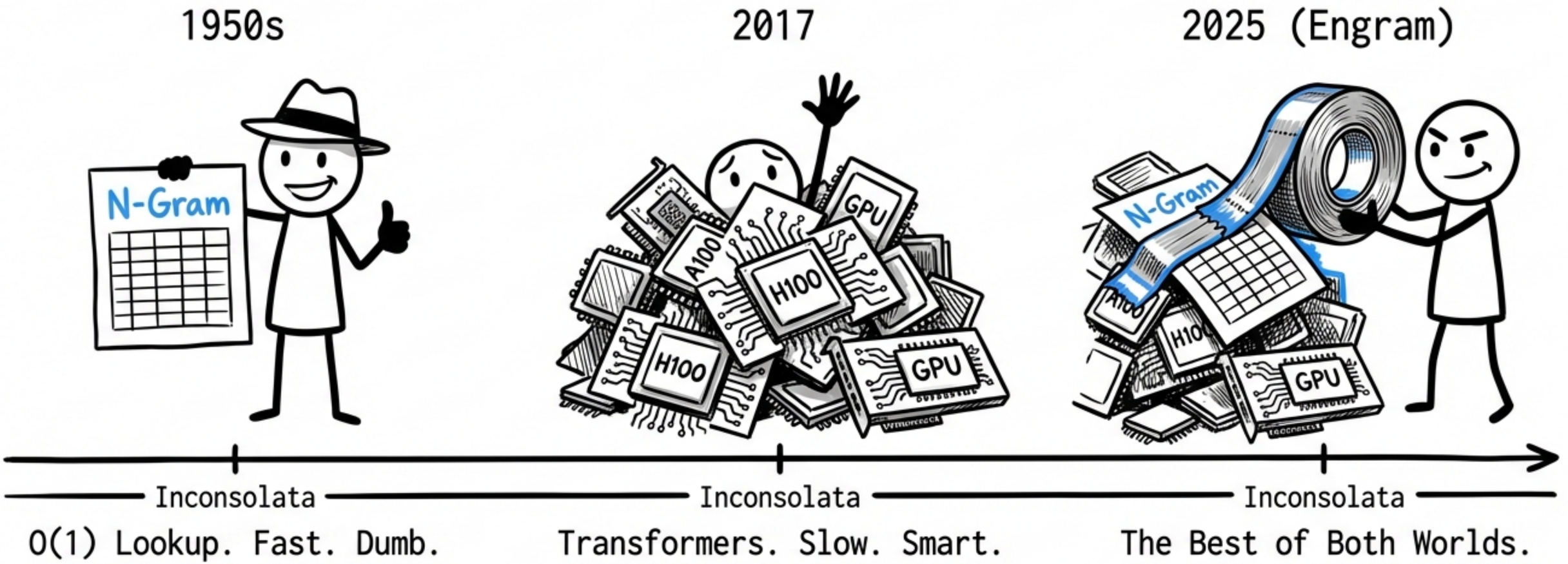
Inconsolata: Based on “Conditional Memory via Scalable Lookup” by Cheng et al. (2025). Warning: Contains hand-drawn charts and abuses of the te

The Status Quo: Simulation via Computation



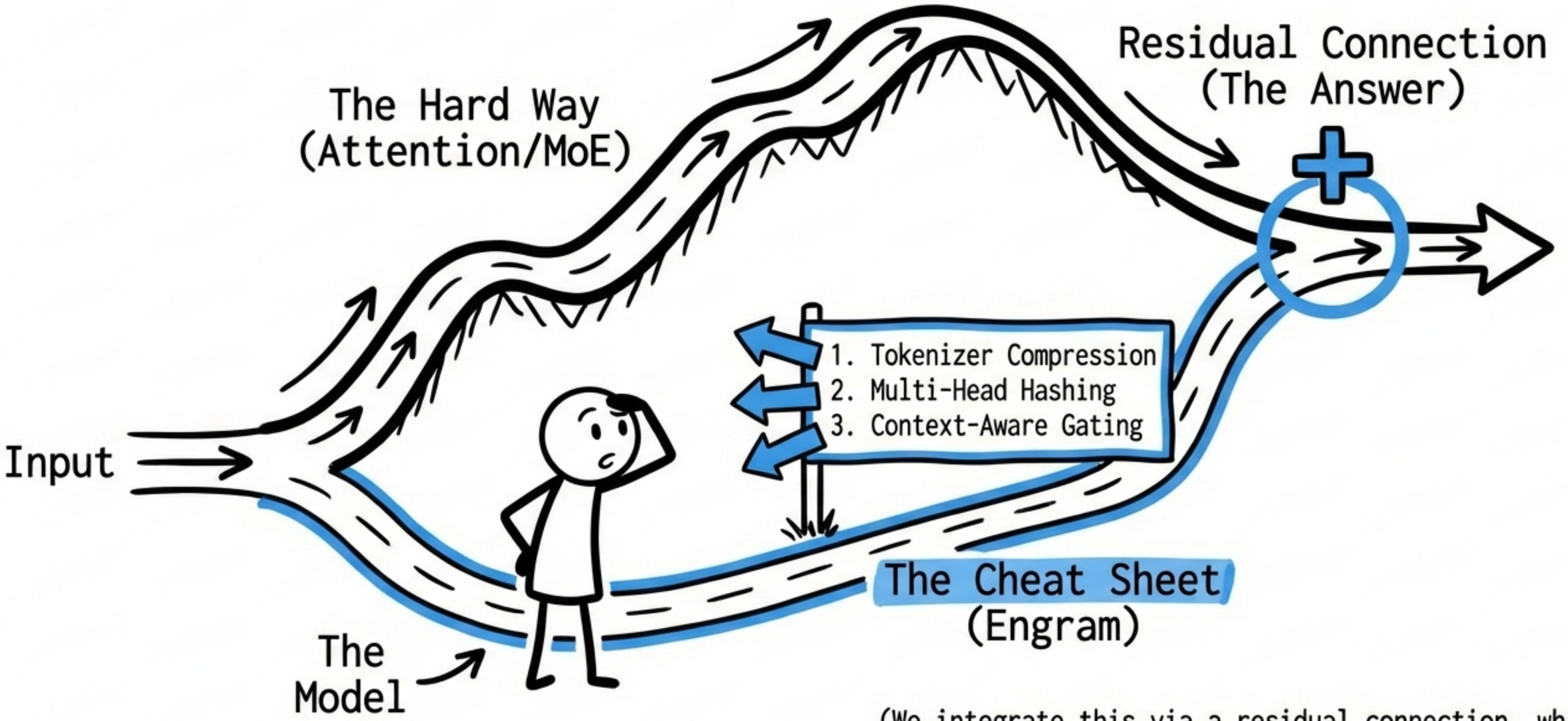

```
// THE PROBLEM:
1. Transformers lack a native
   knowledge lookup primitive.
2. To resolve a static entity like
   'Diana, Princess of Wales', the
   model consumes multiple layers of
   Attention and FFNs.
3. Result: Expensive runtime
   reconstruction of a static lookup
   table.
```

The ‘New’ Idea: Stapling a 1990s Search Engine to a H100



The Insight: Use ‘Conditional Memory’ as a new axis of sparsity.
Why compute what you can just find in a filing cabinet?

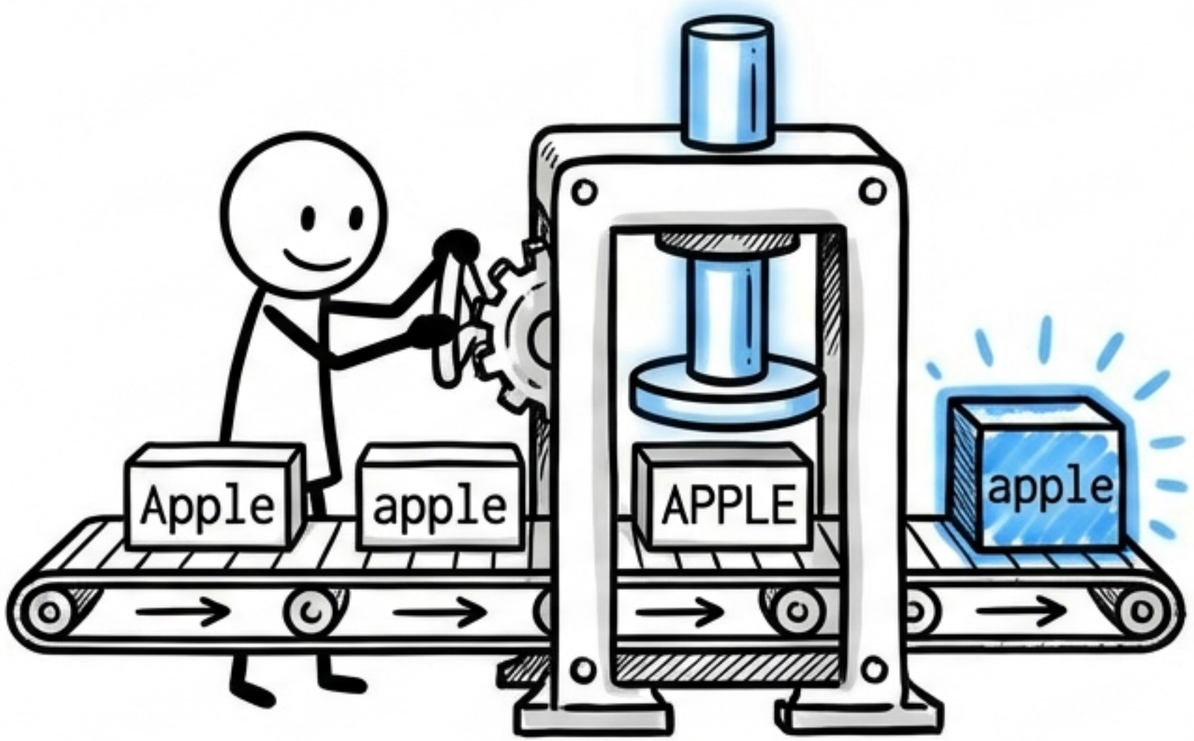
The Architecture: The Model's 'Cheat Sheet'



(We integrate this via a residual connection, which is fancy talk for 'adding the answer at the end'.)

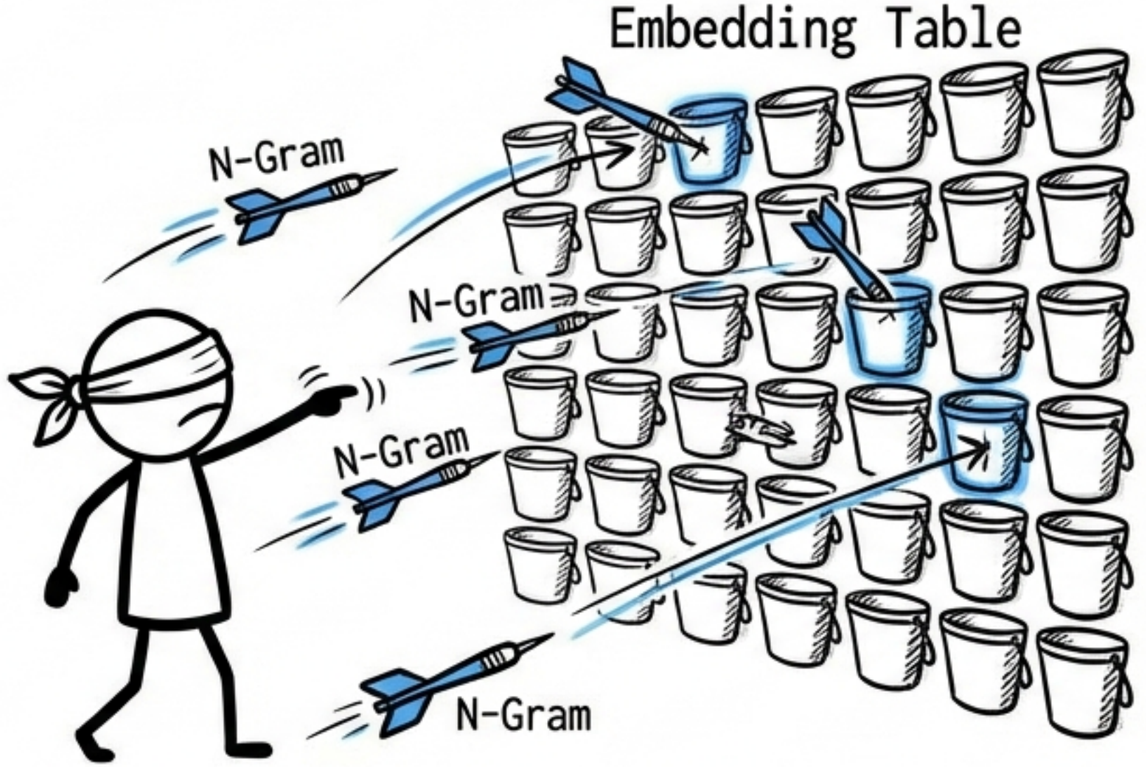
Architects Daughter: Step 1: Squashing and Hashing

A. Compression



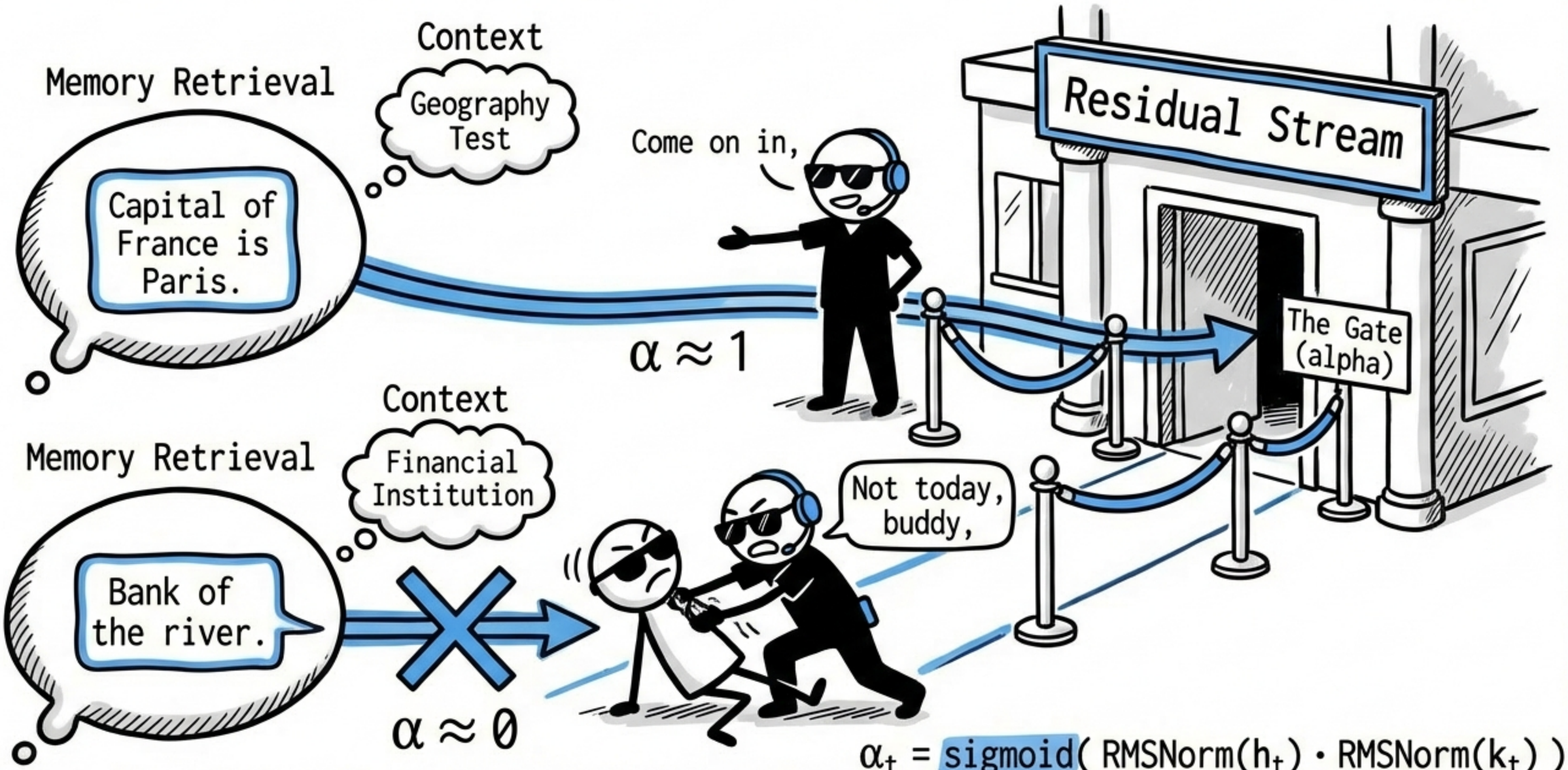
Tokenizer Compression: Surjective mapping reduces effective vocab size by ~23%.

B. Multi-Head Hashing



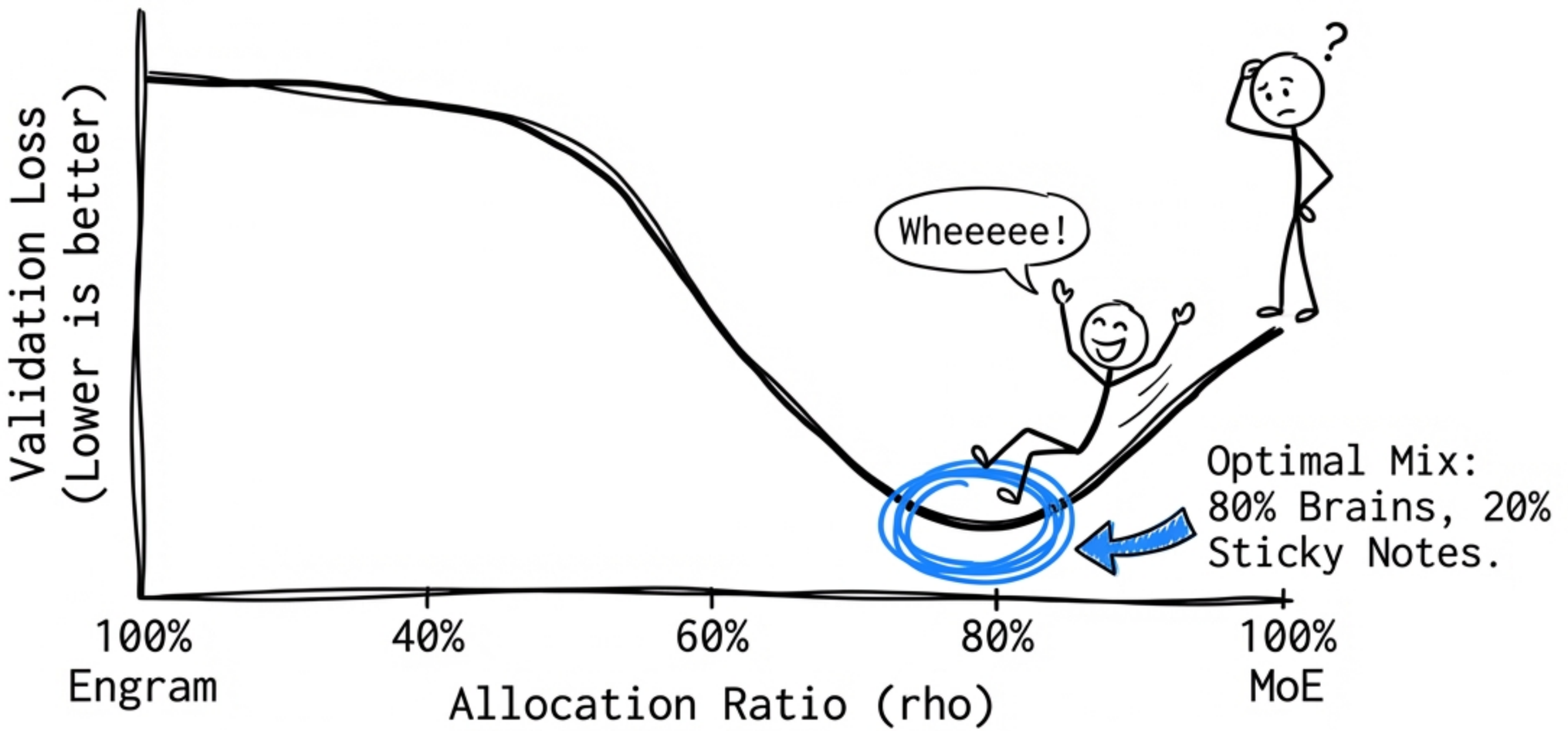
Multi-Head Hashing: We use K distinct hash heads. We rely on the magic of collisions not mattering as much as you think.

Architects Daughter: Step 2: (Context-Aware Gating)



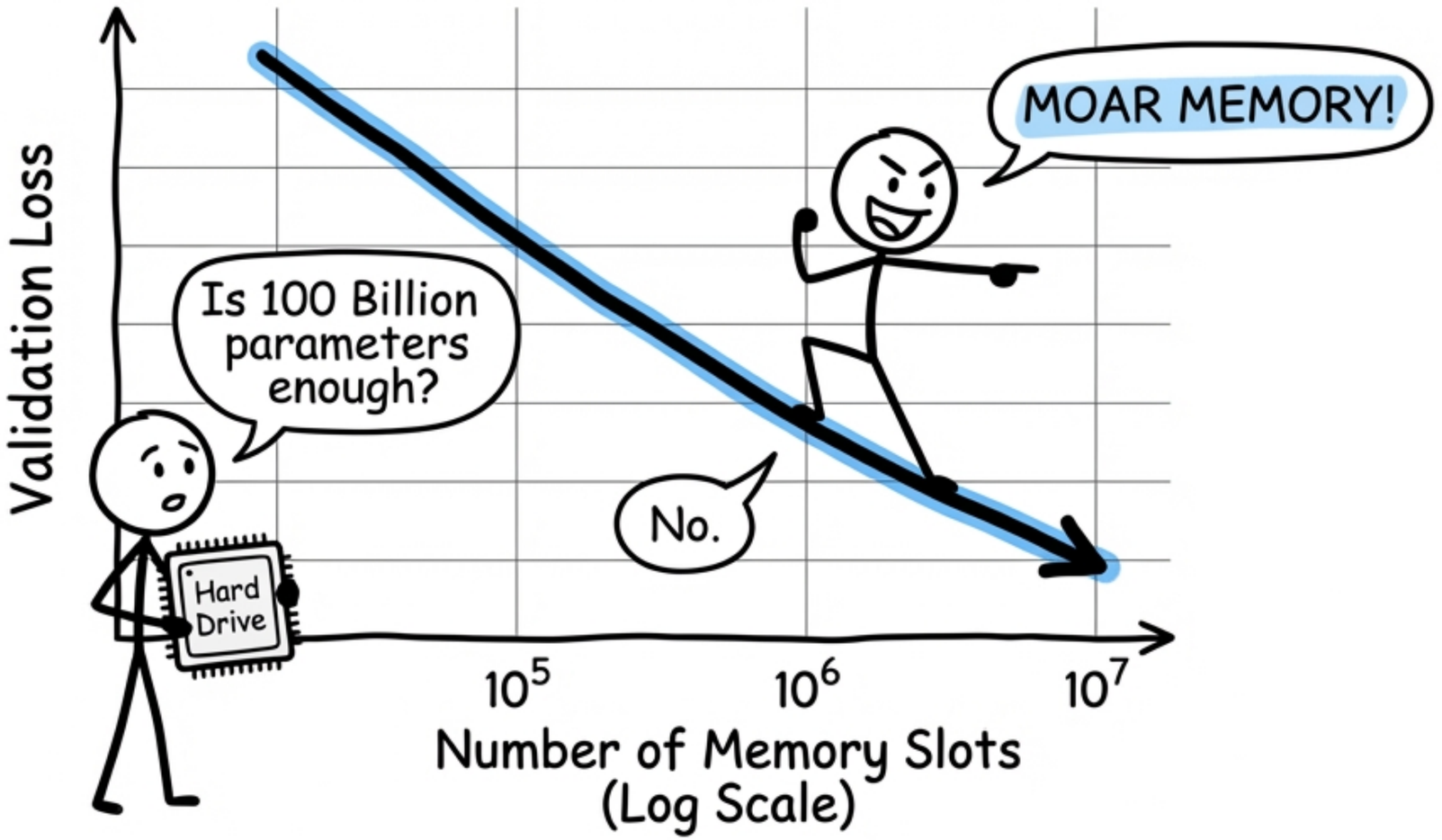
Architects Daughter

The Sparsity Allocation Problem



Pure MoE (rho=100%) is suboptimal. A mix of computation and memory yields better performance.

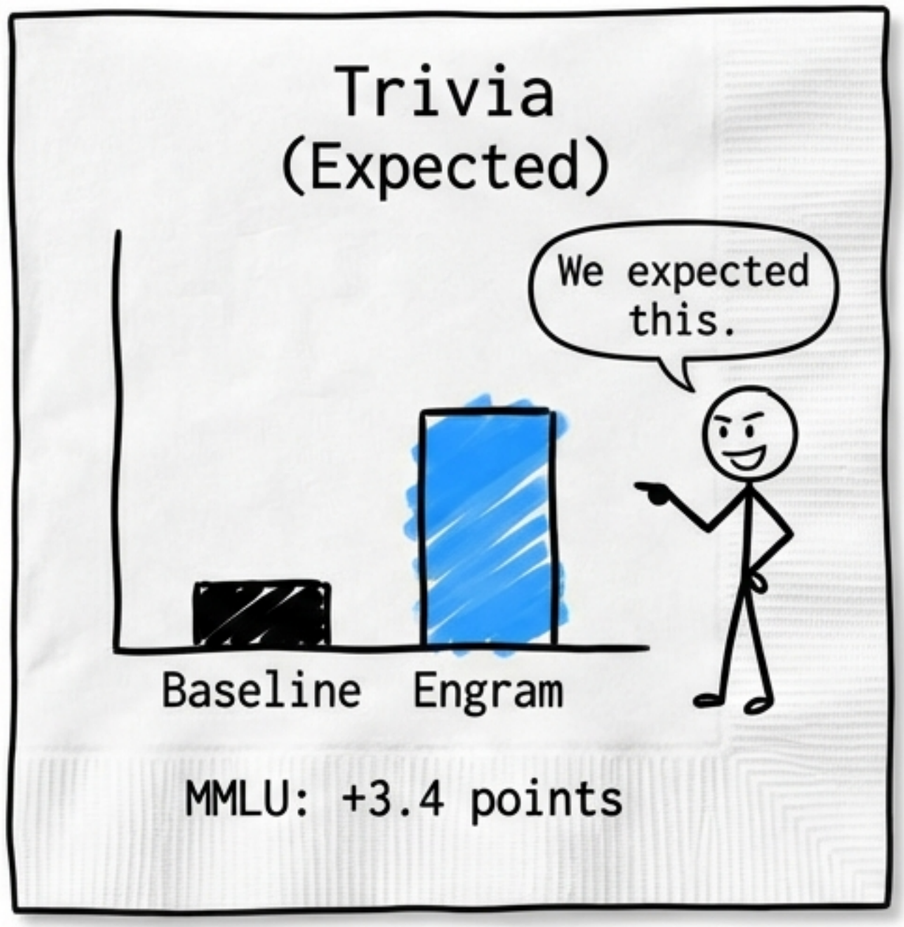
The “Infinite Memory” Regime

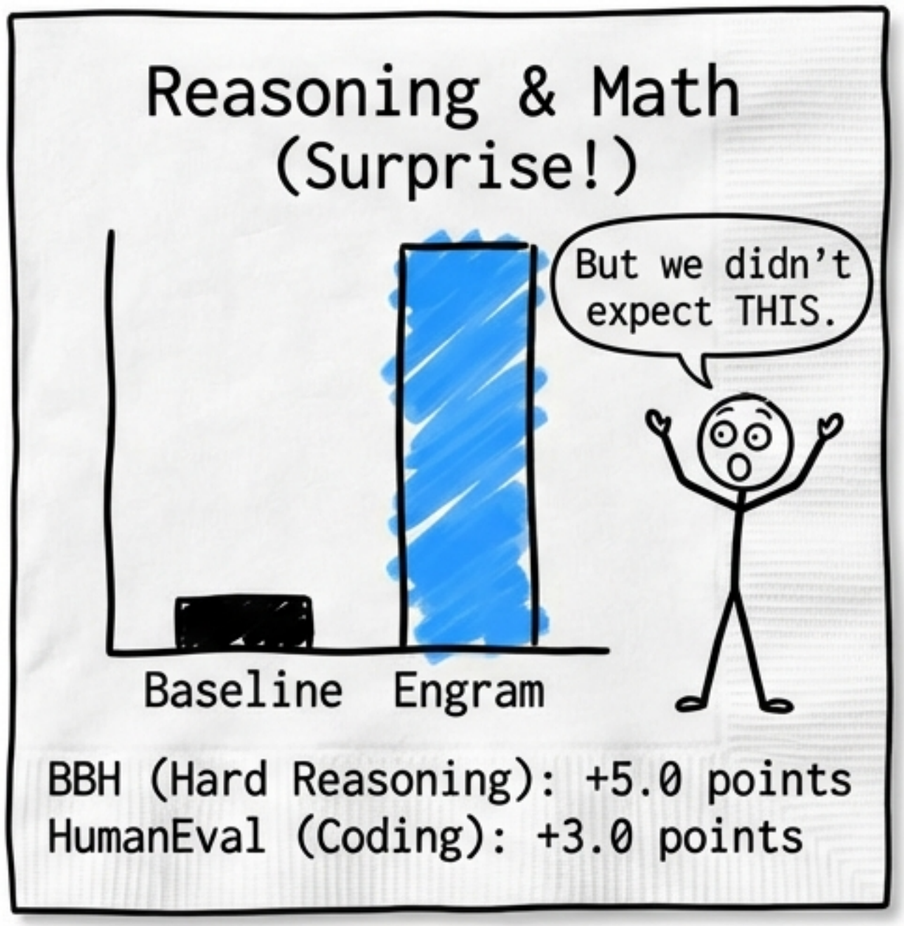


Fixed MoE Backbone (3B). Memory adds capacity without increasing compute cost.

Architects Daughter

The Plot Twist: It’s Not Just for Trivia

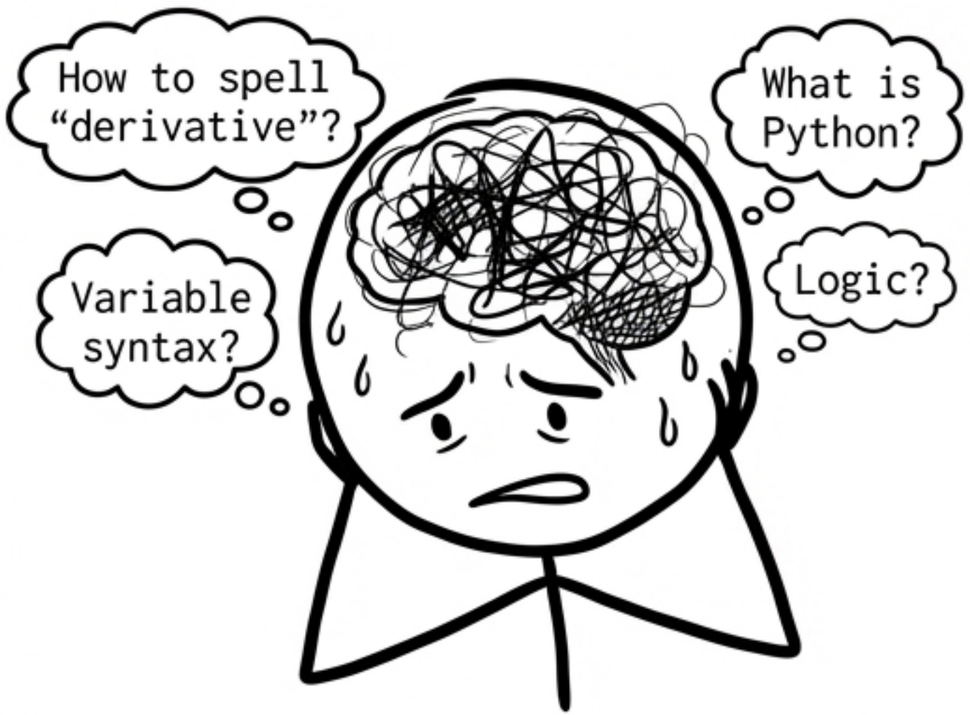




We built a cheat sheet for history class, but it helped us pass Calculus. Weird.

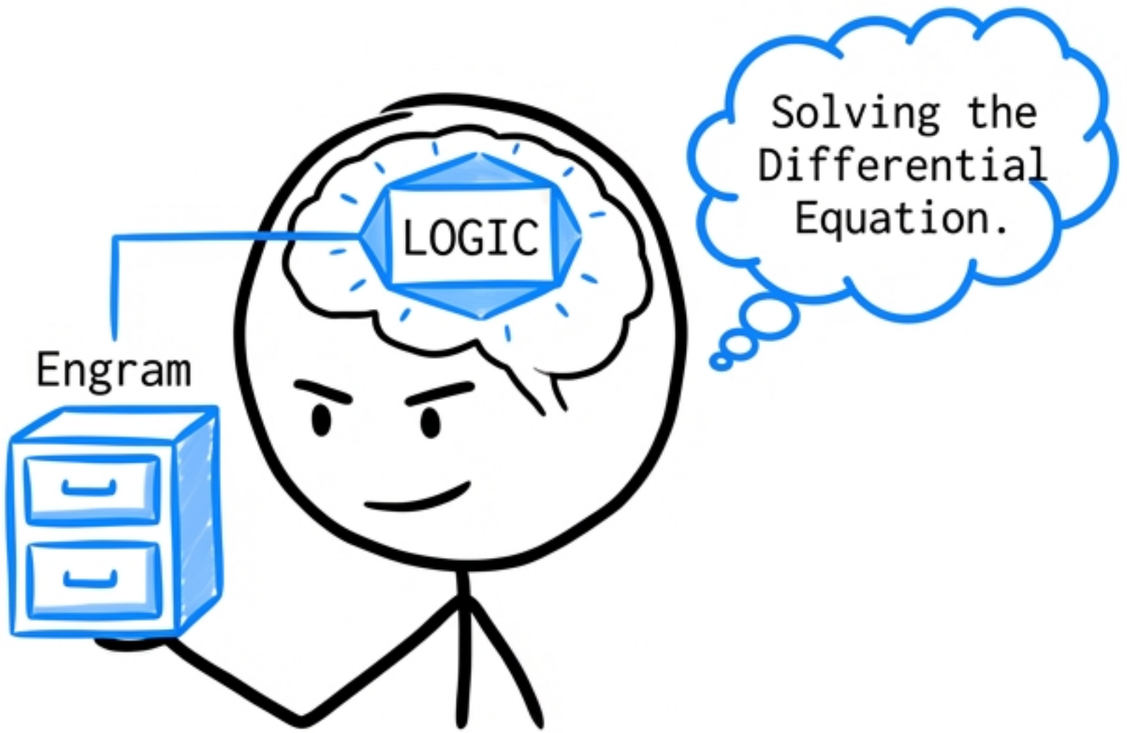
Why does Memory help Reasoning?

Standard Model



Brain clogged with static patterns.

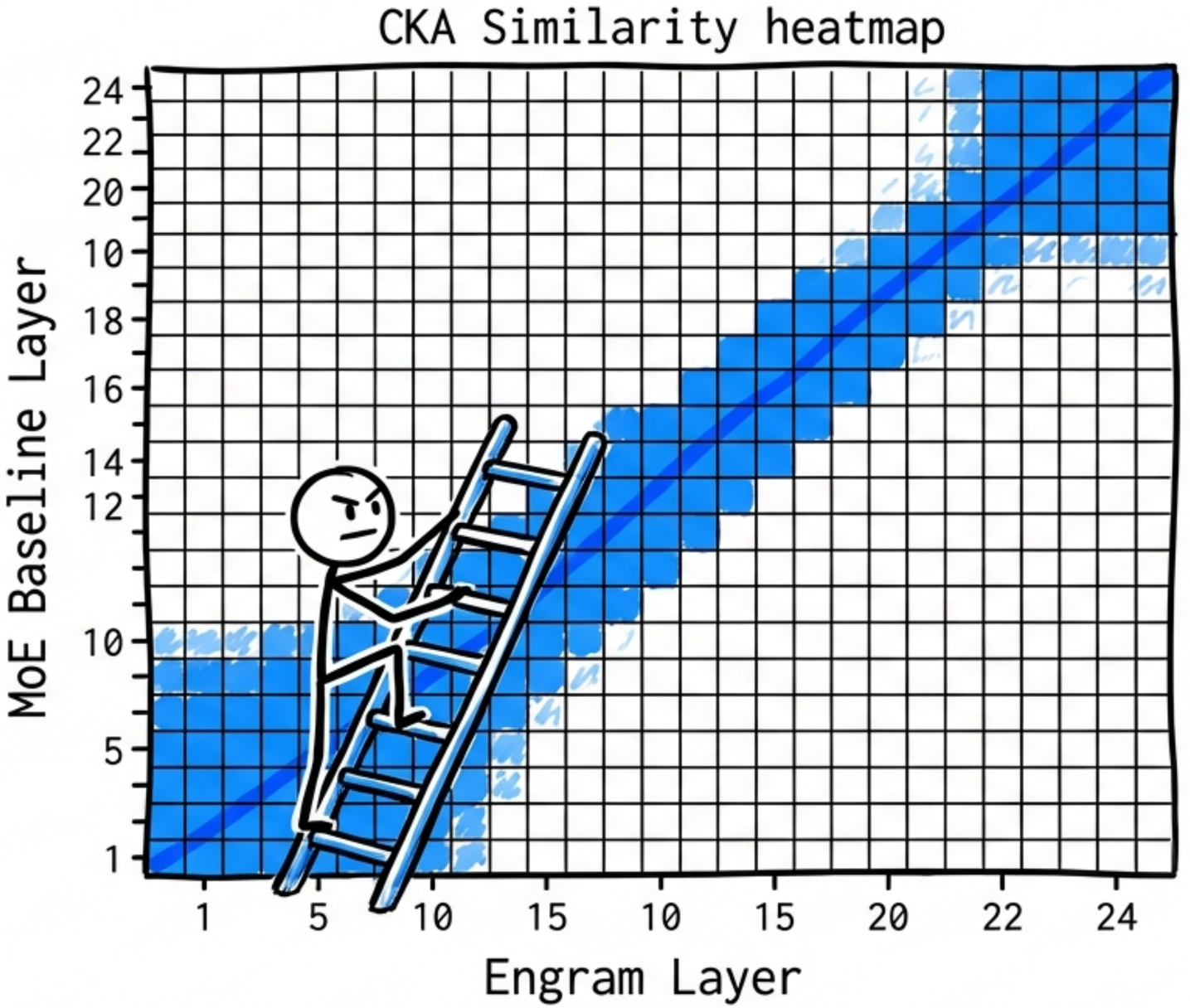
Engram Model



Static patterns offloaded to lookup. Brain free to think.

"Don't sweat the small stuff. (Lookup the small stuff)."

Proof: The Shortcut Mechanism



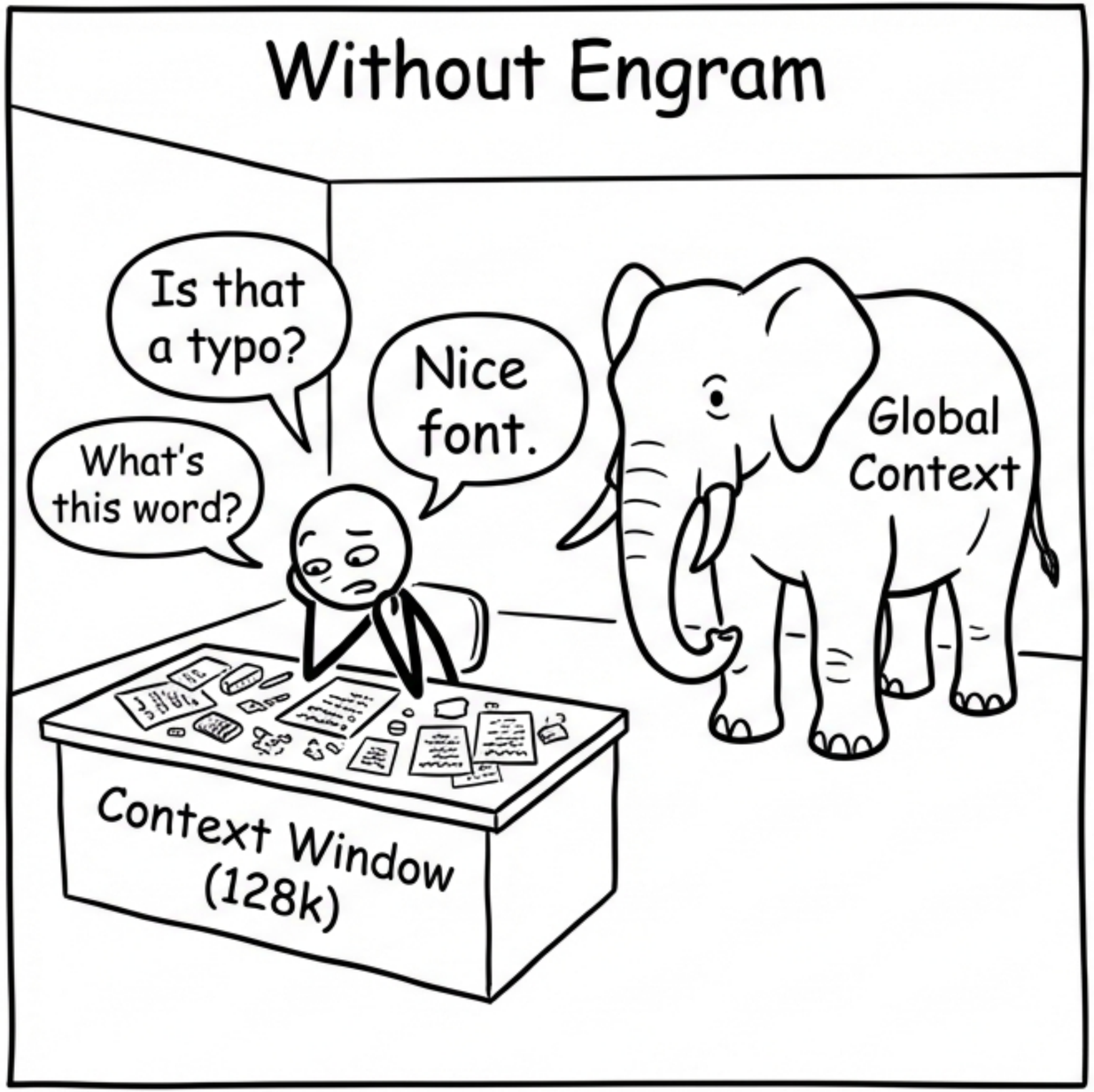
CKA Analysis & LogitLens: Architects Daughter

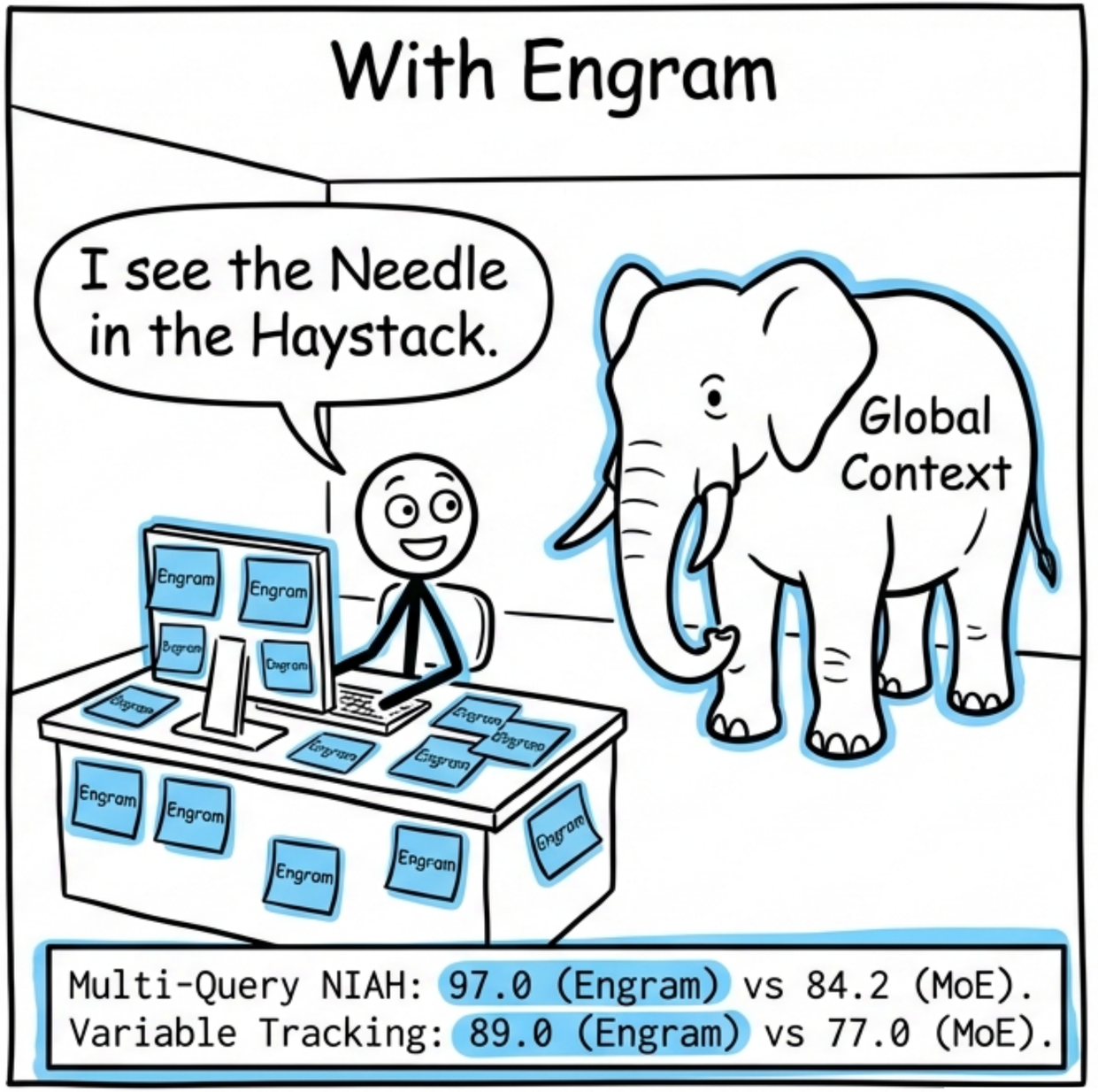
1. Engram's shallow layers are functionally equivalent to the the MoE baseline's deeper layers.

2. The model skips the "Who am I?" phase and gets straight to work.

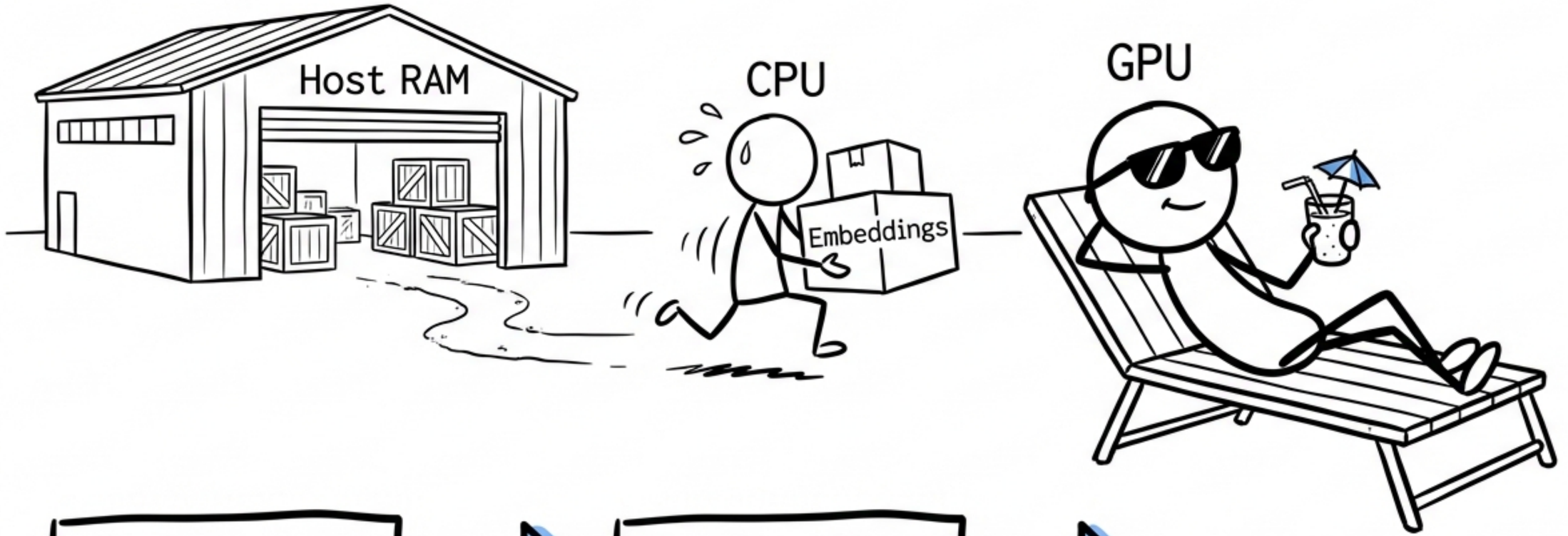
3. Effective Depth > Actual Depth.

Attention is Precious (Don't Waste It)





The Lazy GPU Strategy (System Efficiency)



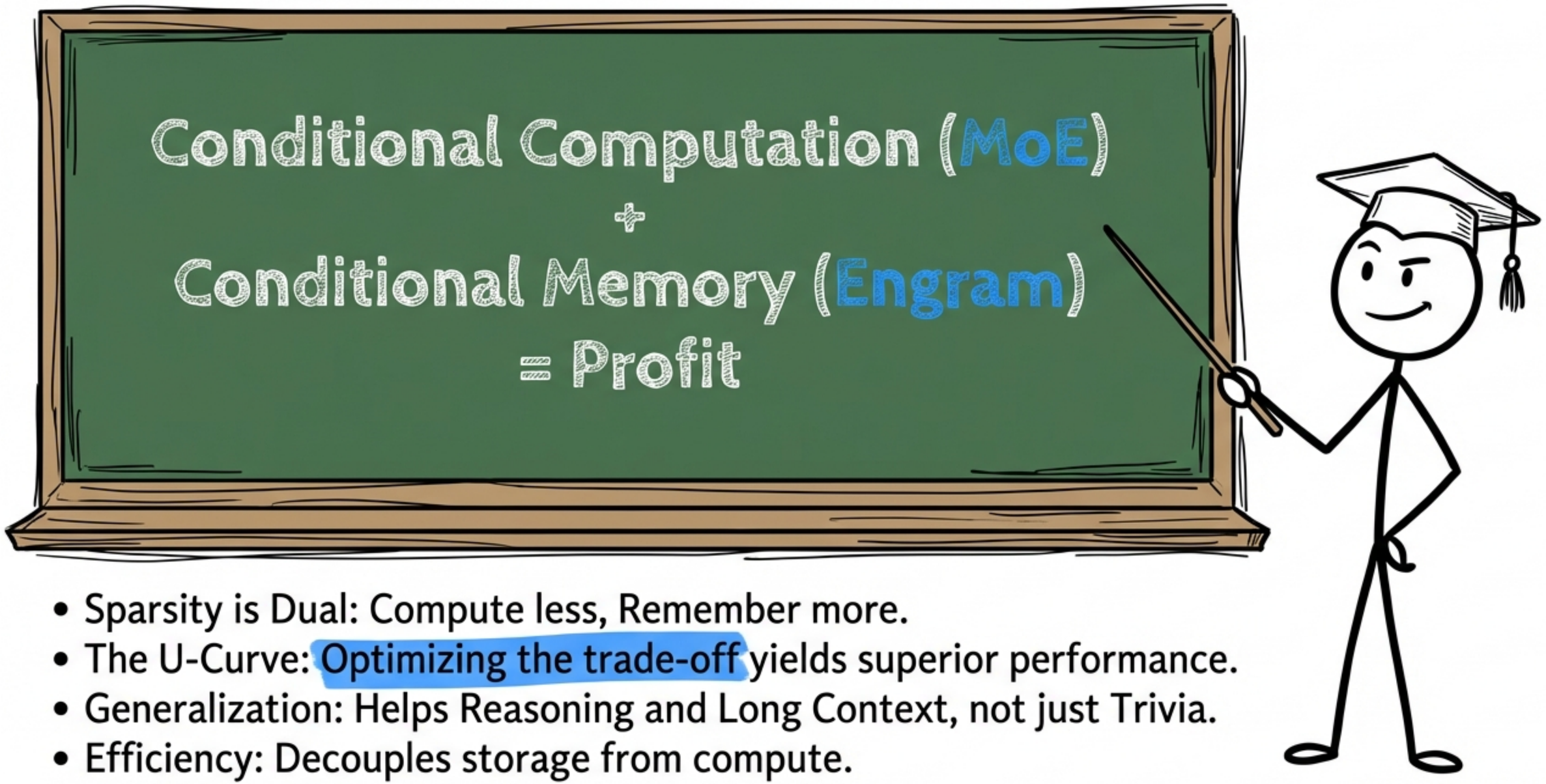
Determinism
Determinism

Prefetching
Prefetching

GPU stays happy

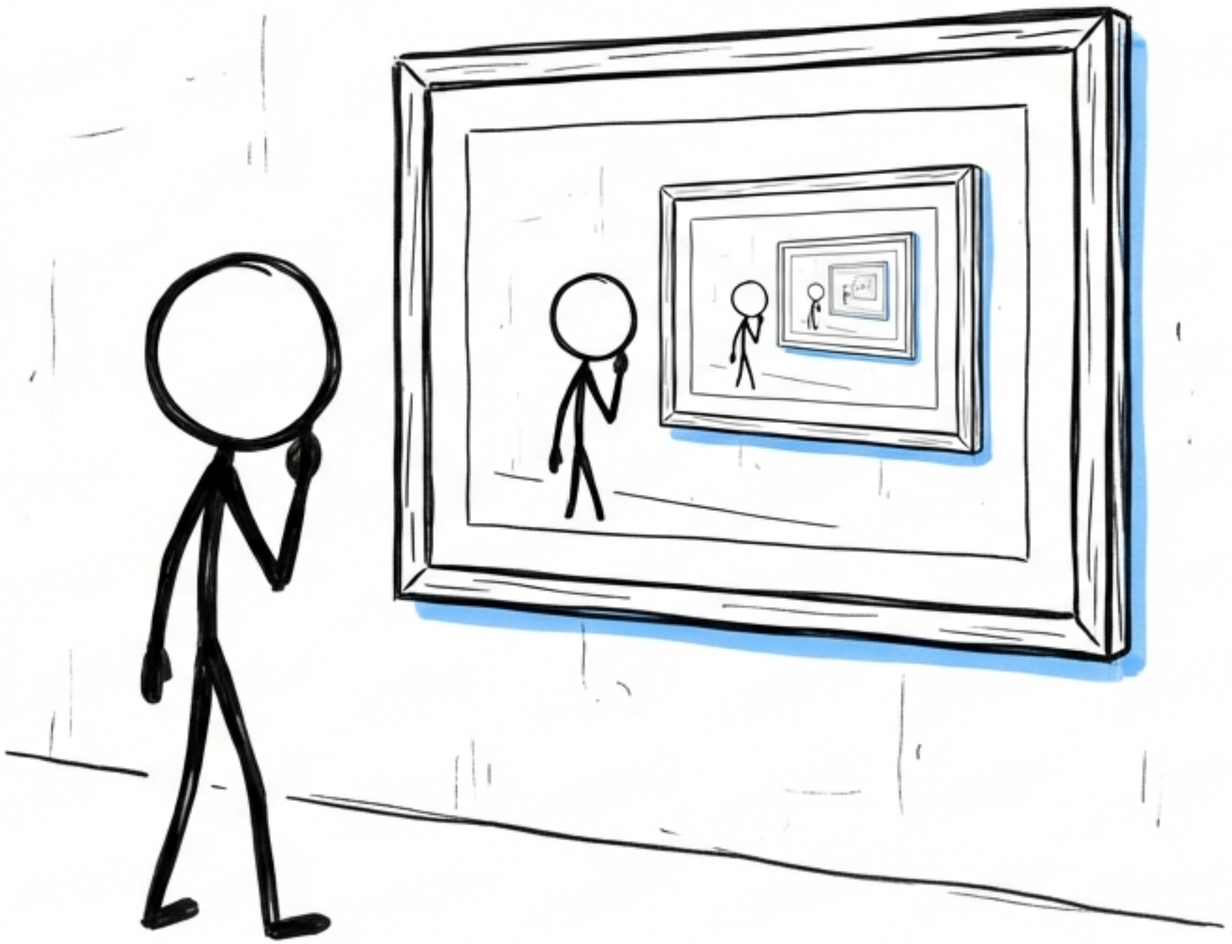
Overhead < 3% even with 100B-parameter table offloaded to CPU RAM.
Because addressing is deterministic, we know what the GPU needs before it asks for it.

The Takeaway



- Sparsity is Dual: Compute less, Remember more.
- The U-Curve: Optimizing the trade-off yields superior performance.
- Generalization: Helps Reasoning and Long Context, not just Trivia.
- Efficiency: Decouples storage from compute.

Future Work



We envision conditional memory as an indispensable primitive for next-gen models.

We also envision eventually forgetting how to multiply because the Engram module does it for us.

Code available at: https://github.com/deepseek-ai/Engram