Academic Industrialism
High-Fidelity Engineering

Engineering Algorithmic Structure

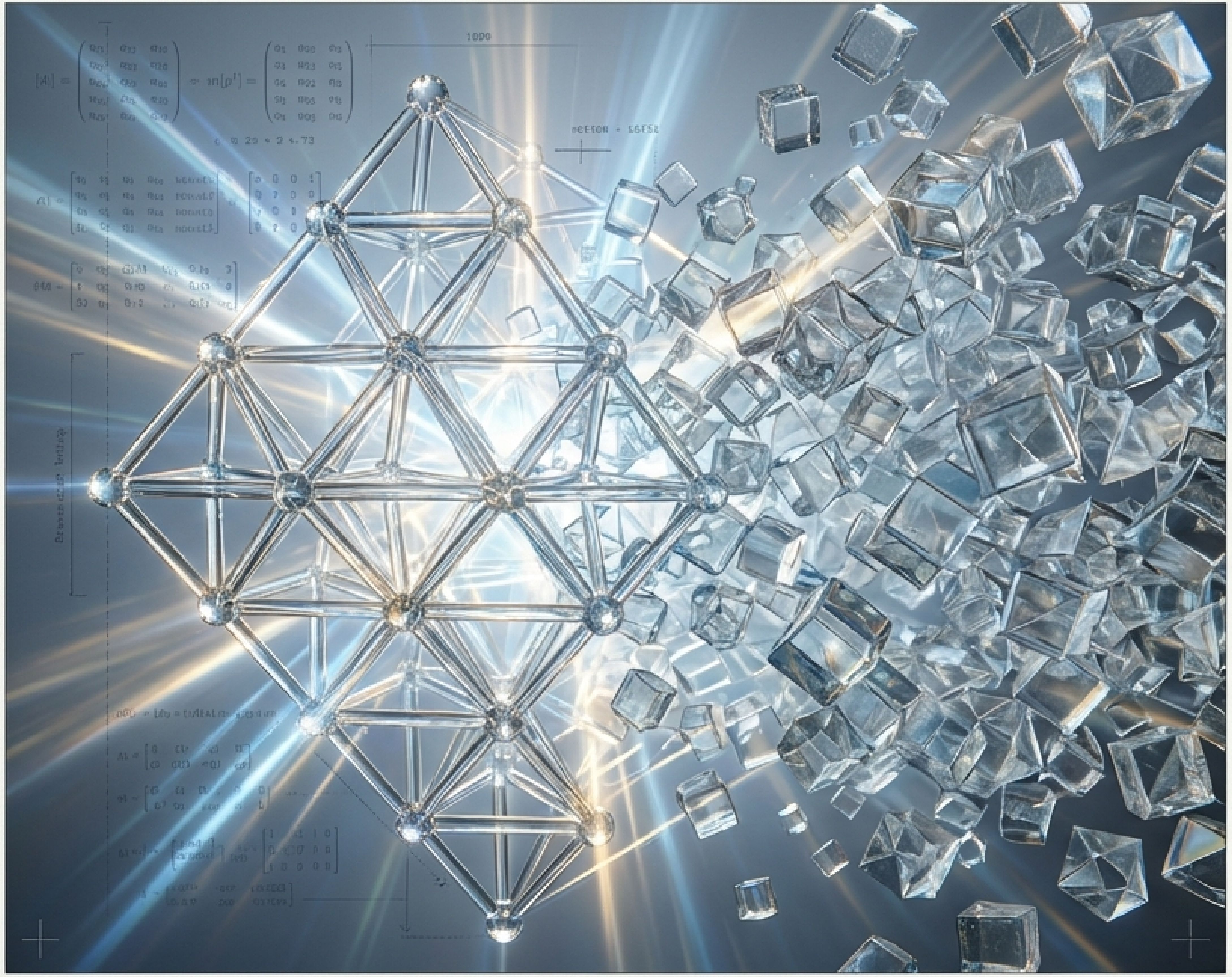
A Materials Science Perspective on Deep Learning

Author: grisun0

Context

Inducing and verifying Strassen matrix multiplication in neural networks.

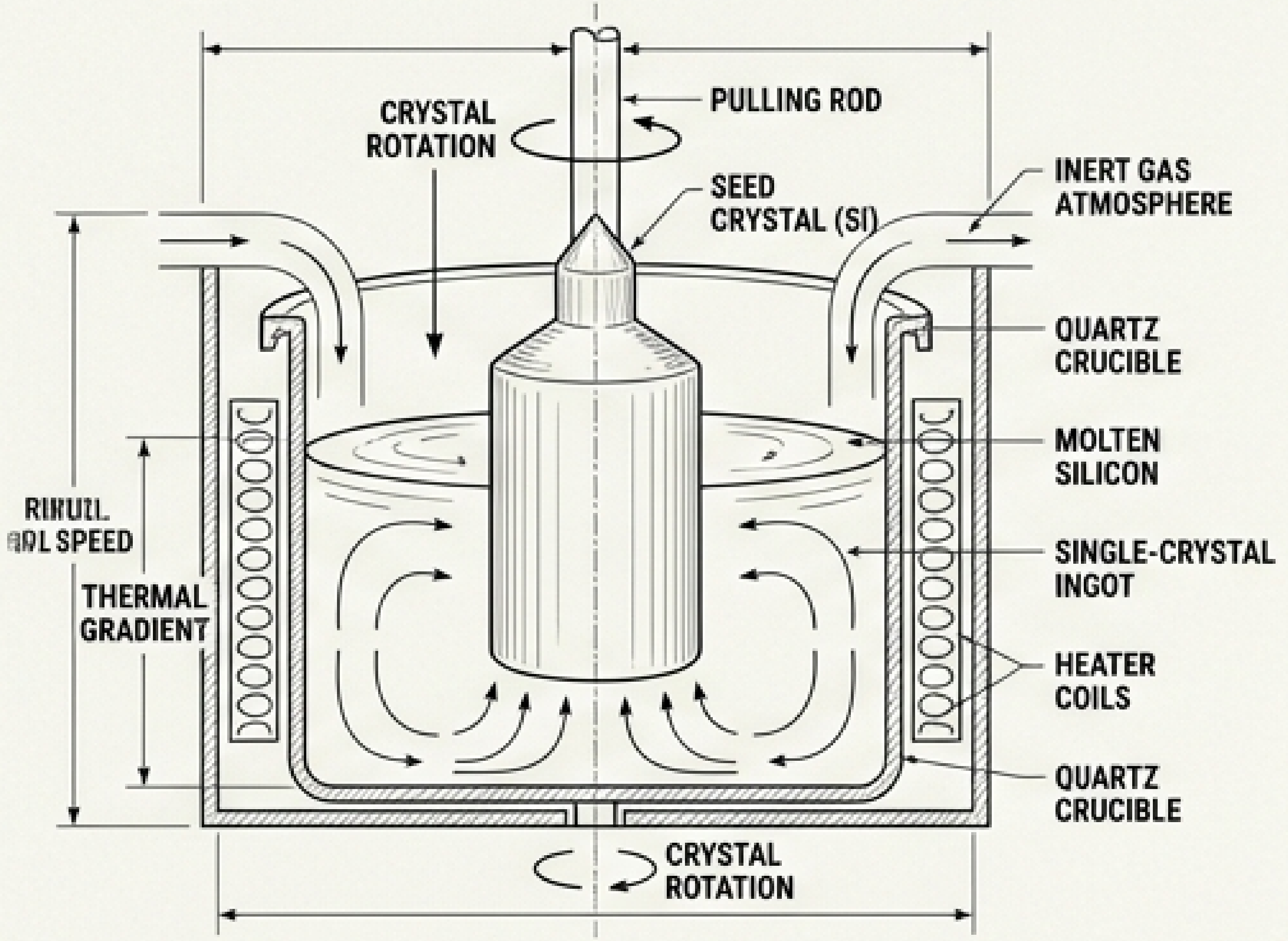
`paradigm_shift: active_construction vs passive_emergence`



Growing Crystals vs. Understanding Electrons

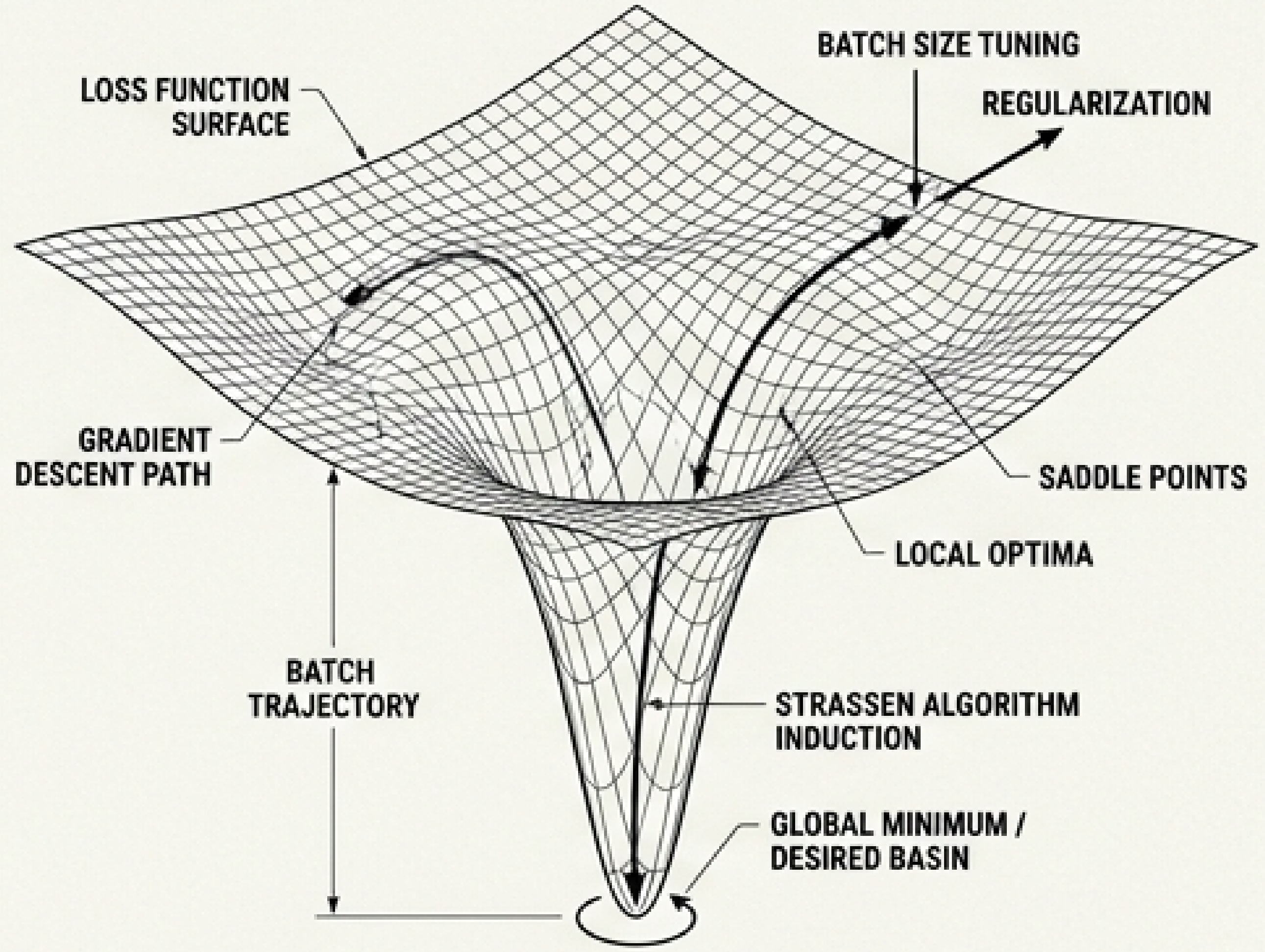
Moving from theoretical discovery to manufacturing recipes.

The Analogy: Semiconductor Mfg.



We possess exact recipes to grow silicon crystals (Czochralski process) without fully mapping electron mobility at the quantum level. We rely on the "recipe" for yield.

The Application: Deep Learning

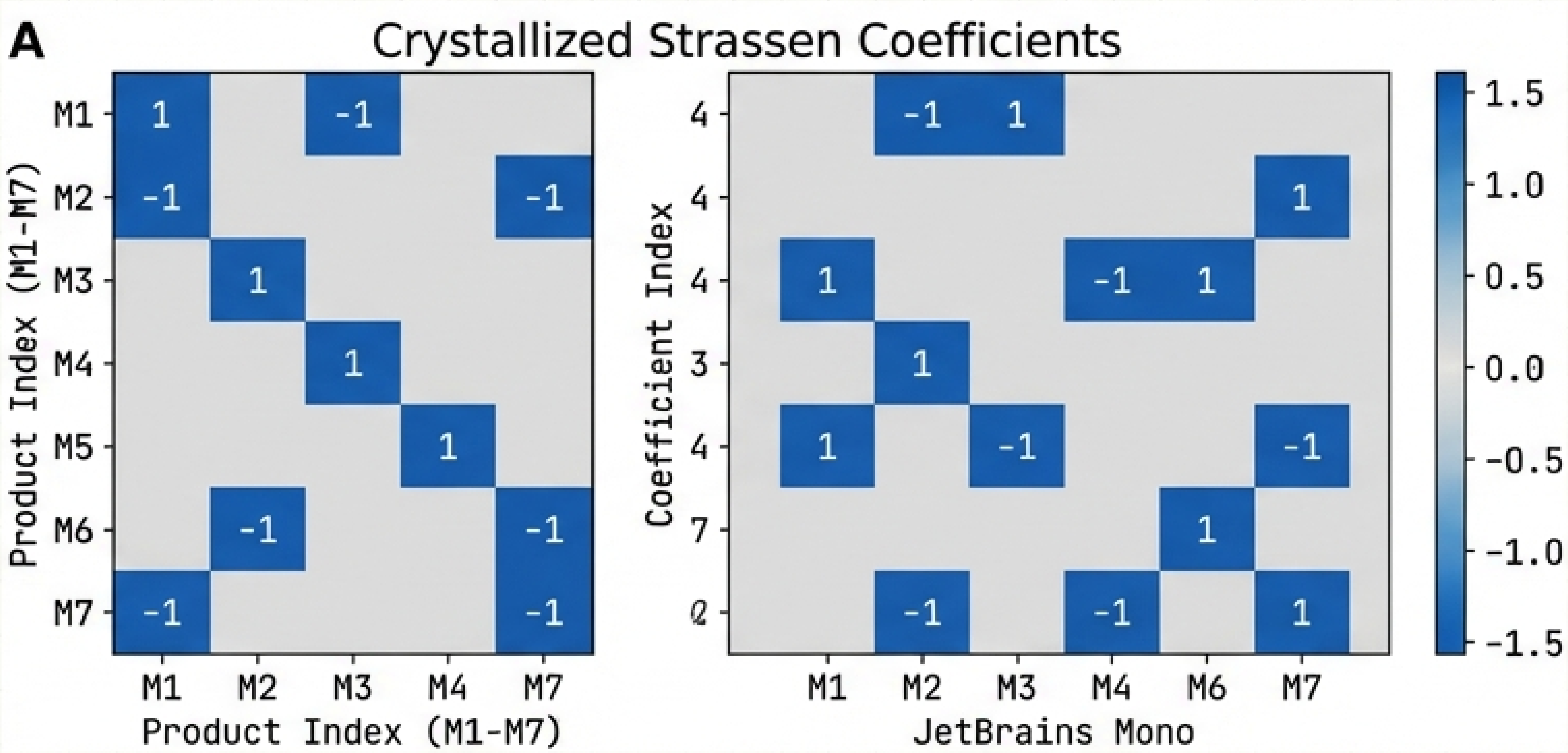


We can reliably induce Strassen algorithms (the "crystal") by tuning batch size and regularization, even if the theoretical mechanism (κ) remains elusive.

Insight: The goal is **Active Construction** of material properties, not passive observation.

The Target Material: Strassen Matrix Multiplication

Objective: Learn exact 2x2 matrix multiplication using only 7 products (Rank-7) instead of the standard 8. This provides a rigorous test case because the structure is discrete $\{-1, 0, 1\}$ and mathematically verifiable.

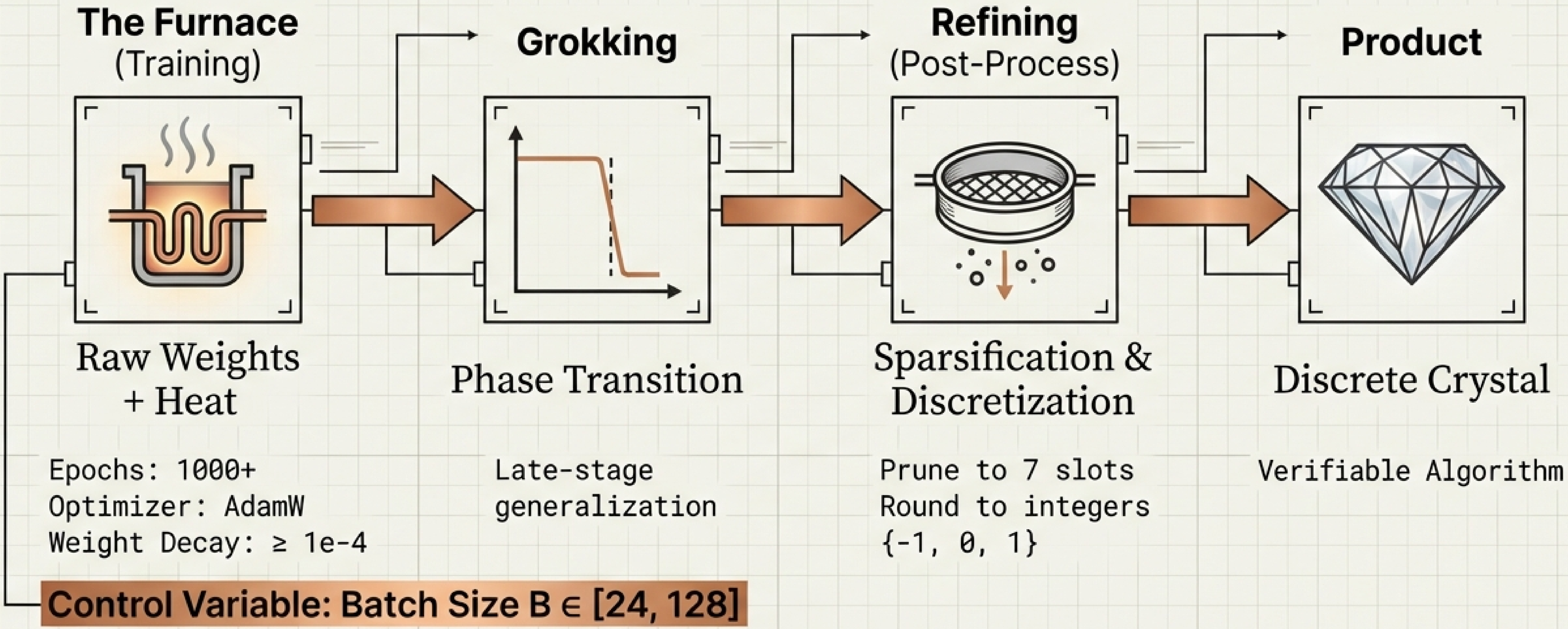


The “Crystal”: A rigid, discrete structure discovered within the neural weights.

Technical Paper

A Recipe for Crystallization

Protocol Yield: 68% Verified Structure

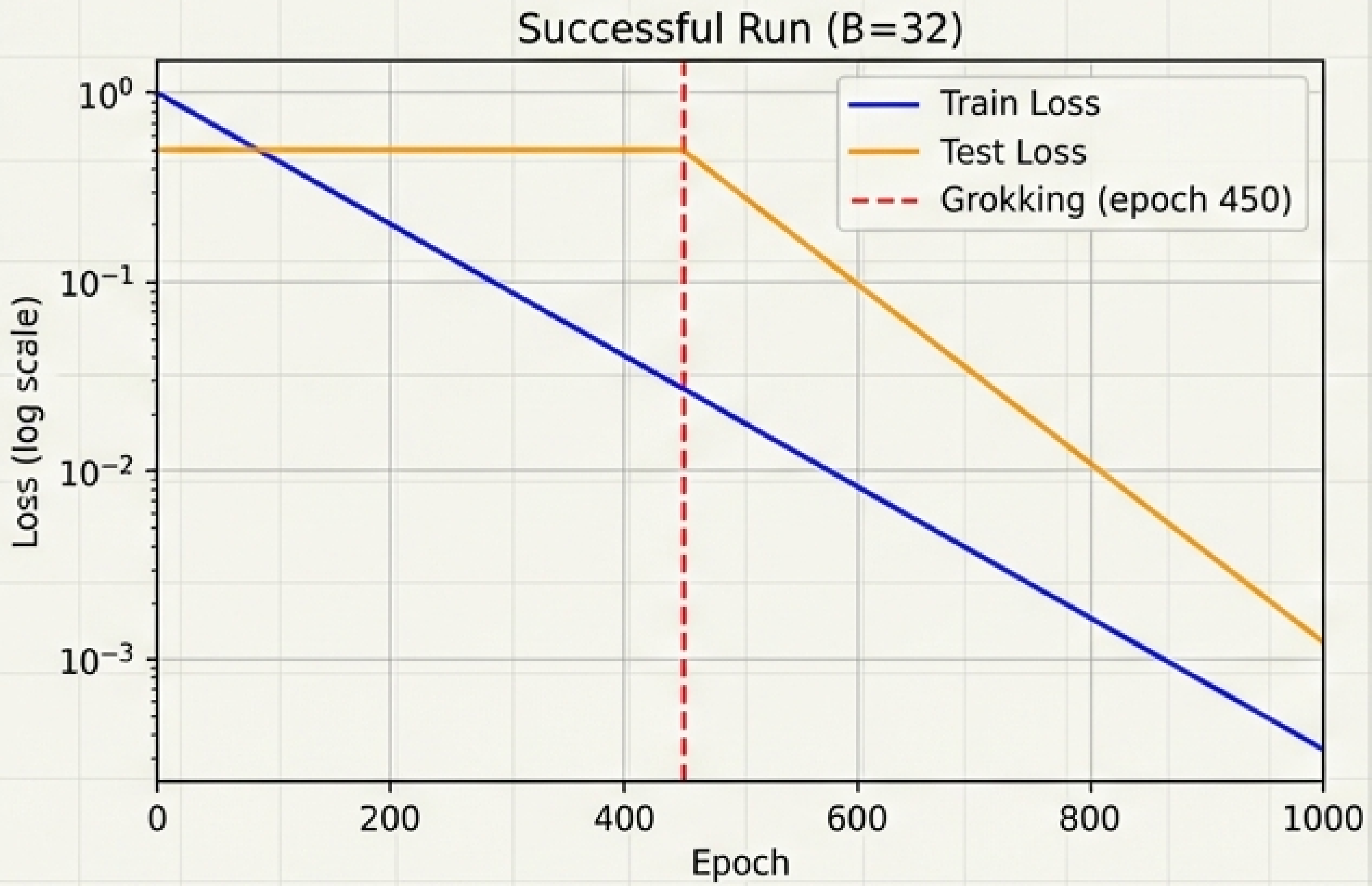


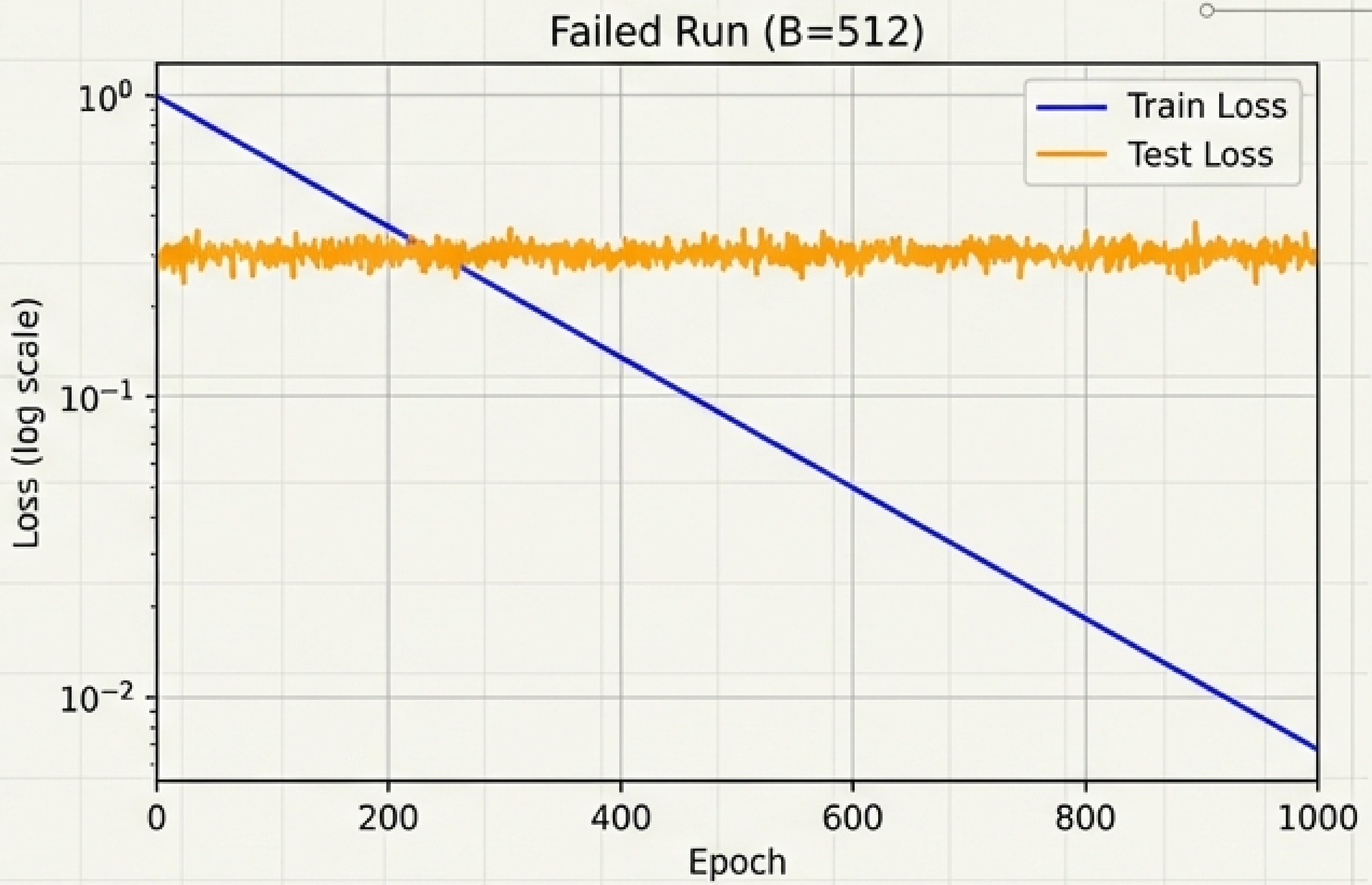
Technical Paper

Technical Paper

Delayed Generalization as a Phase Transition

The 'Grokking' Phenomenon





The 'crystal' does not form immediately. It requires extended 'baking' time. The successful run ($B = 32$) exhibits a sudden phase transition (Grokking), while the large batch run memorizes data but never learns the algorithm.

The Critical Control Parameter: Batch Size

The Finding

Success is statistically clustered in batch sizes 24–128 ($F=15.34$, $p<0.0001$).

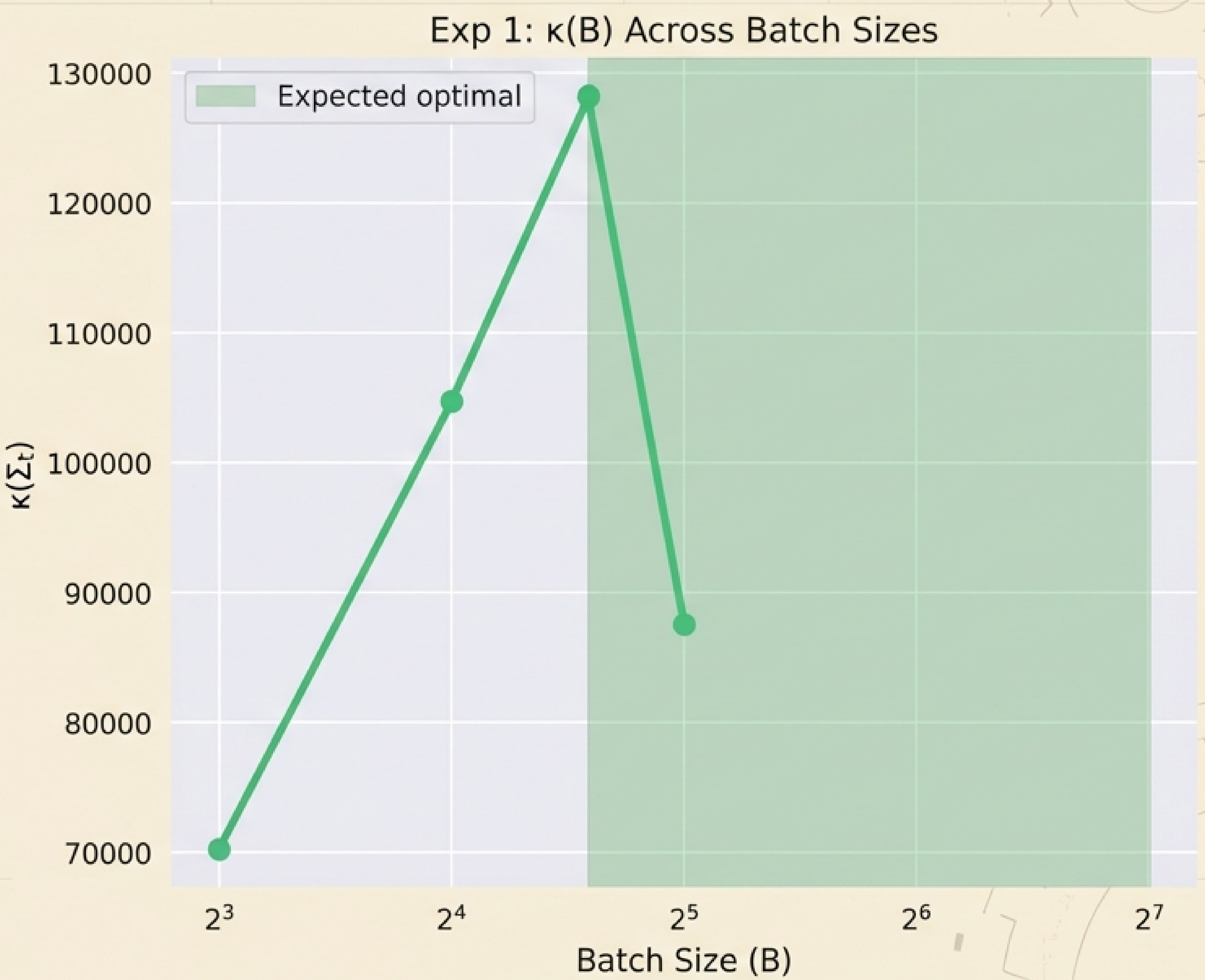
The False Lead (Hardware Cache)

Hypothesis: Do small batches fit in L3 cache?

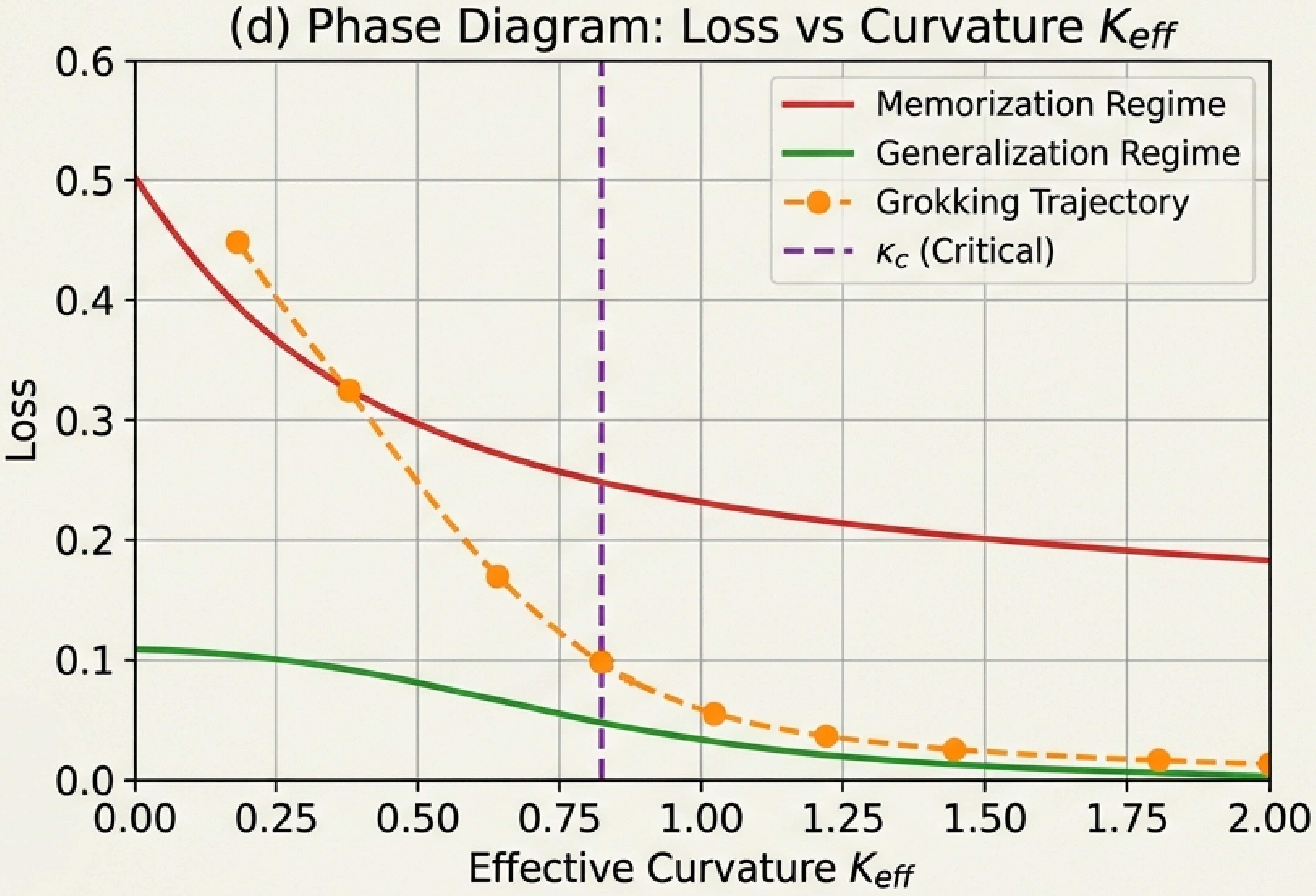
Correction: Even B=1024 takes only ~321 KB. Modern L3 caches are >1MB. This is *not* a hardware constraint.

The Reality

Batch size dictates the **trajectory geometry** through the loss landscape. It is an algorithmic constraint.



The Phase Diagram of Algorithmic Learning



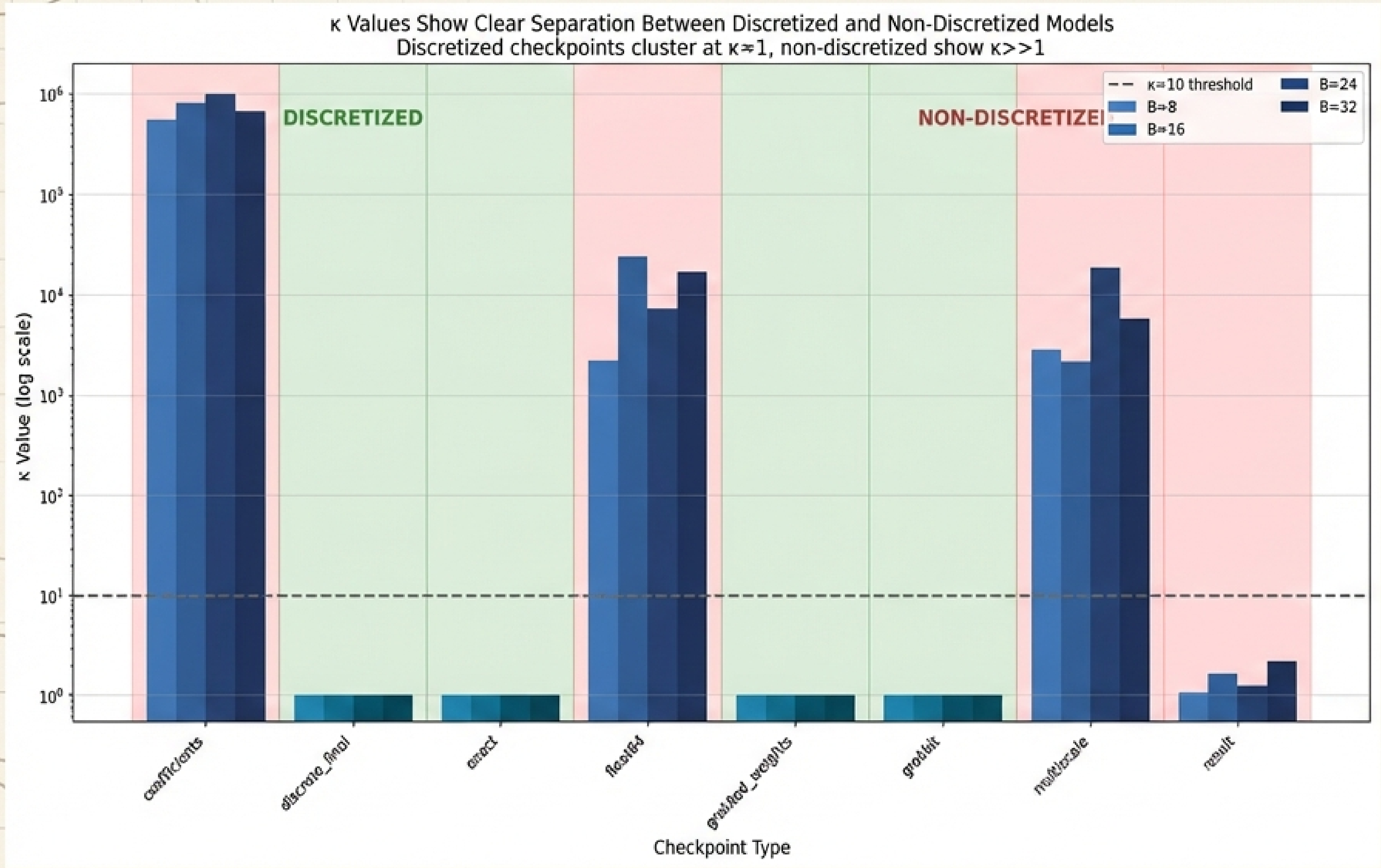
The Sweet Spot

Just as water requires specific temperature and pressure to freeze, Strassen structure requires:

1. Batch Size $B \in [24, 128]$
2. Epochs ≥ 1000
3. Weight Decay $\geq$ 1e-4

Technical Paper

Correlation is Not Causation: The κ Metric



The Hypothesis:
That gradient covariance (κ) predicts success.

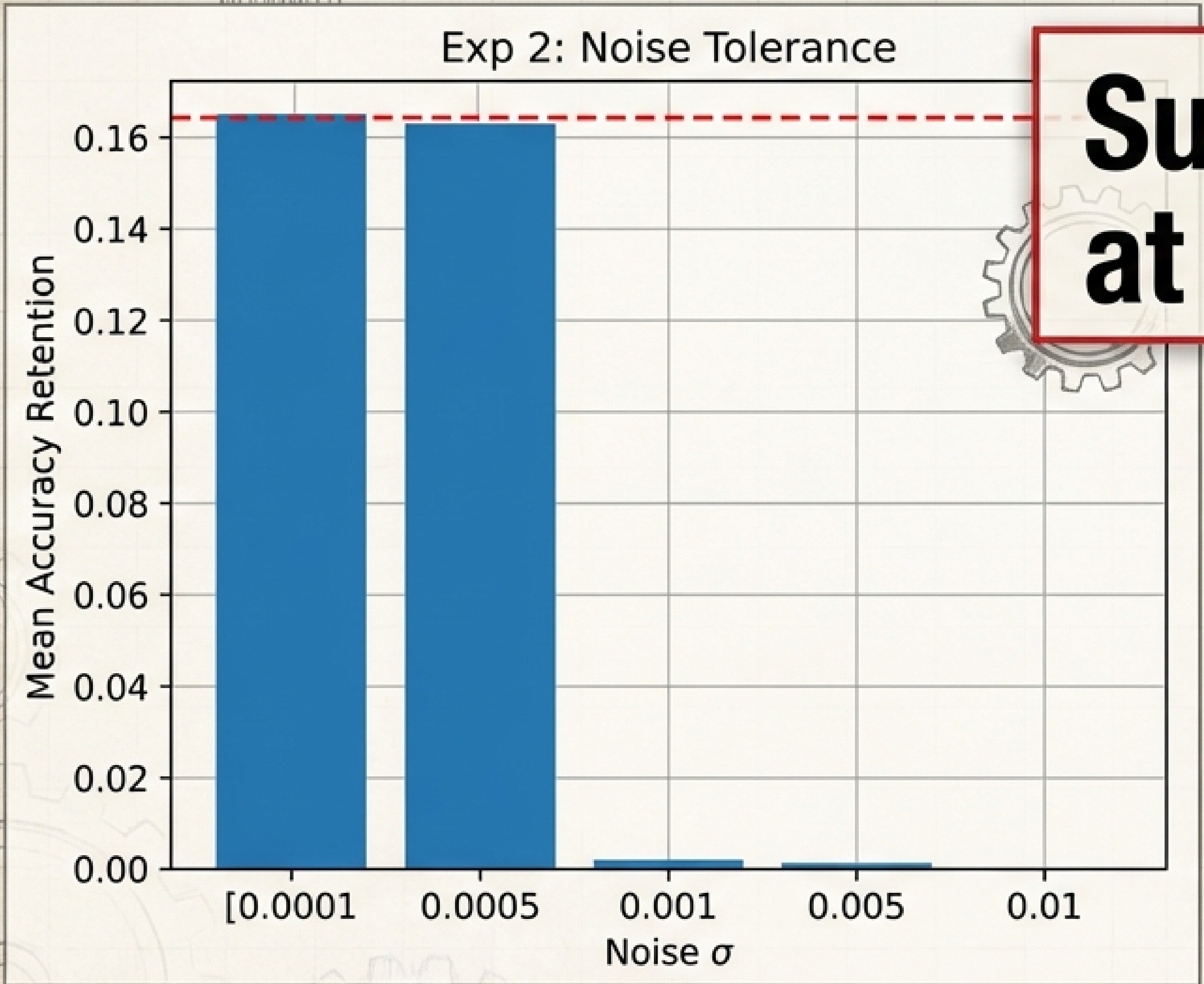
The Evidence:

- *Post-Hoc*: κ ≈ 1 correlates perfectly with successful models.
- *Prospective*: Early measurement yields ~58% accuracy (coin flip).

Conclusion: κ is a **diagnostic signature** of the crystal, not the cause of its growth. We can identify the crystal once formed, but we cannot predict its formation early on.

Technical Paper

The Extreme Fragility of Algorithmic Solutions



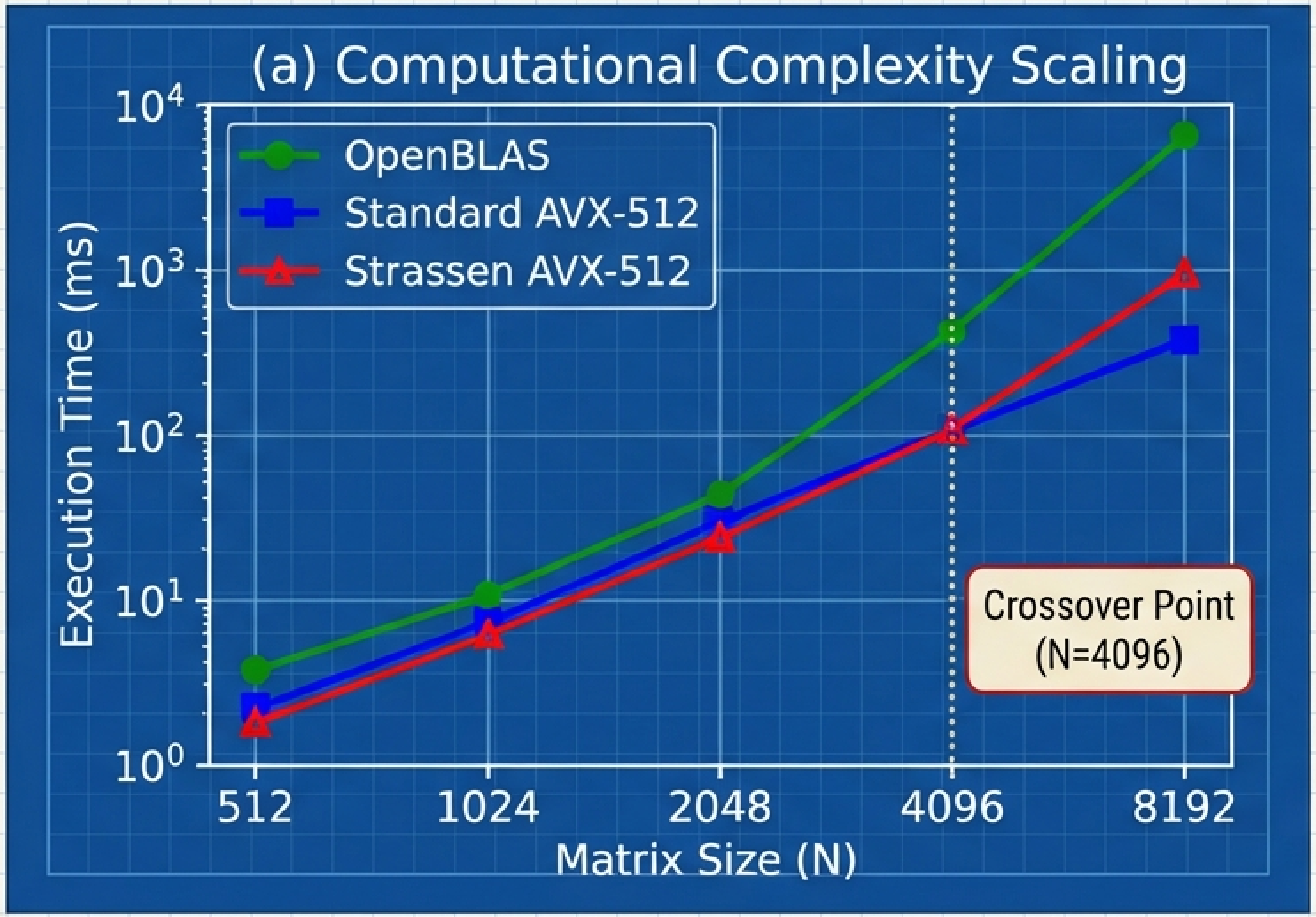
Success Rate at $\sigma \geq 0.001$: 0%

The Test: Adding Gaussian noise to weights before discretization.

The Implication: Algorithmic solutions live in incredibly **“narrow basins”** of attraction. You cannot stumble into them; the training dynamics must steer the trajectory with microscopic precision.

Verification: Zero-Shot Transfer

From 2×2 to 64×64

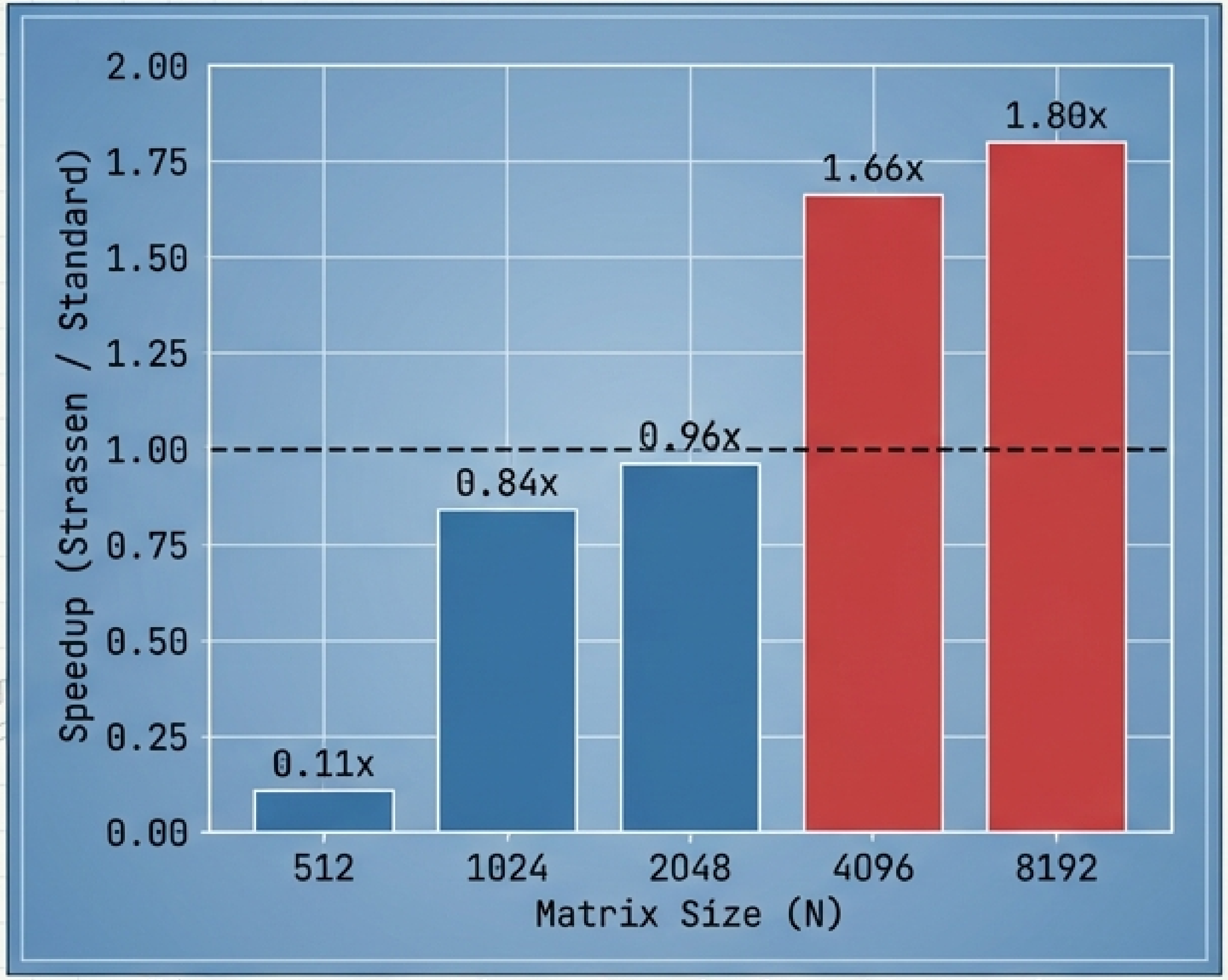


The Win: Structure learned on small 2×2 matrices works perfectly on large 64×64 matrices without retraining.

The Metric: Relative error $< 10^{-6}$ across all scales.

Proof: The network learned the mathematical **operator** T, not just the training data distribution.

Performance: Executable, Not Just Theoretical



(b) Strassen Speedup vs Standard Implementation

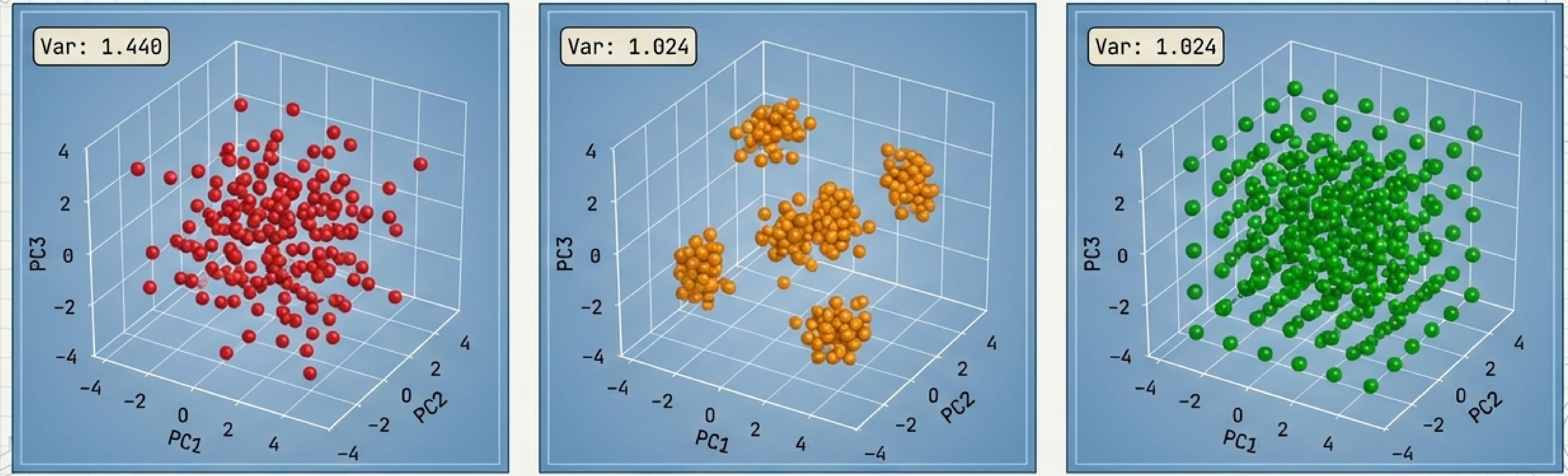
Analysis:

Comparison: Induced Strassen vs. OpenBLAS (Single-Thread).
Result: 1.95x speedup at $N = 8192$.
Nuance: While it beats single-threaded implementations, it is currently slower (0.52x) than highly optimized multi-threaded kernels.

Takeaway:

The goal was structural induction. The result is a fully functional, efficient algorithm.

Geometry of Learning: Visualizing the Trajectory



(a) Random Initialization (Diffuse Point Cloud)

(b) Memorization Phase (Cluster Formation)

(c) Grokking Phase (Crystallization)

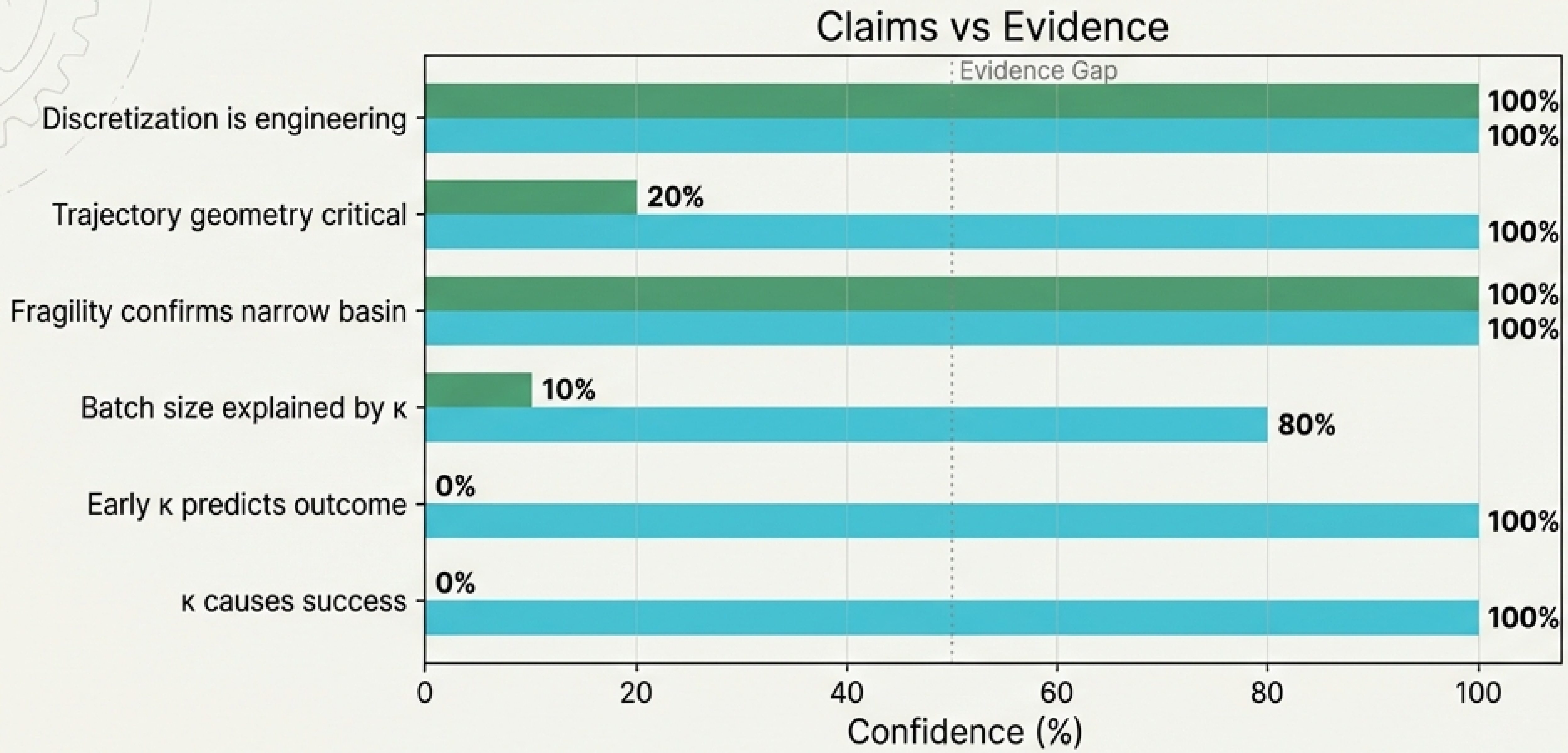
This visualization confirms the **'Basin of Attraction'** concept. The weights must evolve from a random cloud, find specific clusters, and finally snap into a rigid integer lattice.

Technical Paper
JetBrains Mono

#F5F5F0
JetBrains Mono

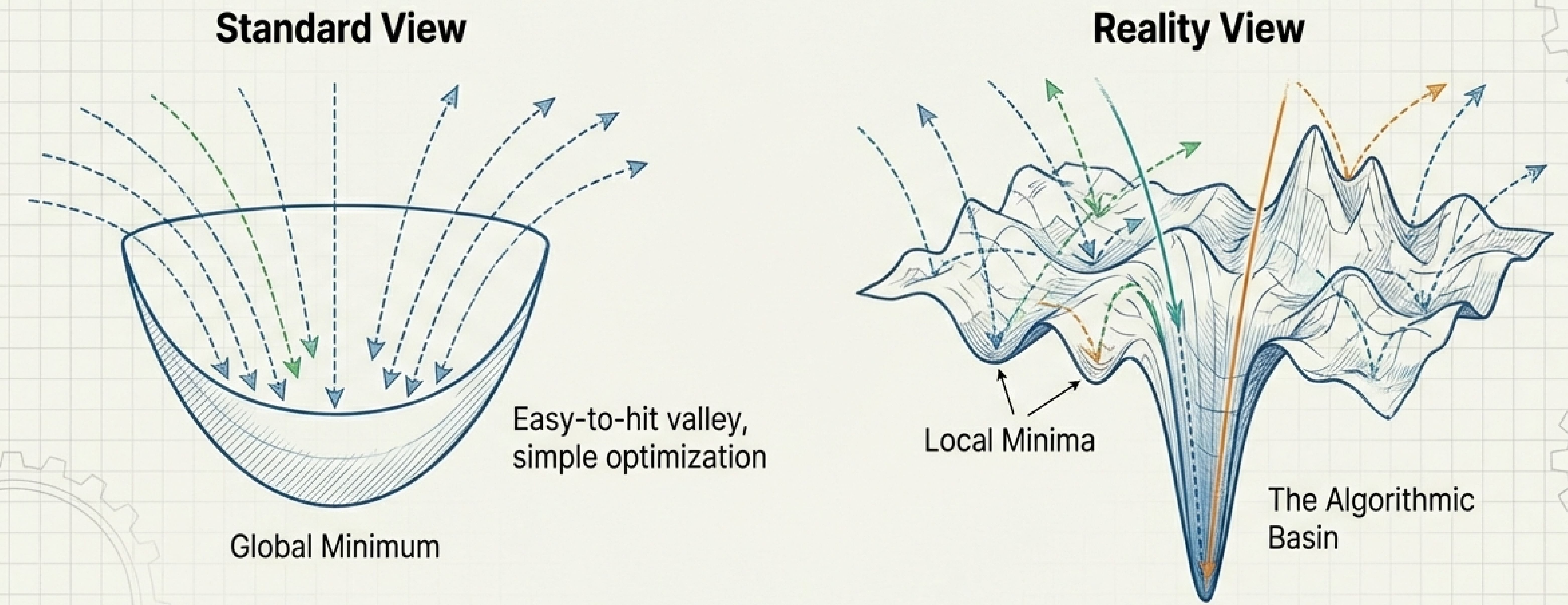
Summary of Claims vs. Evidence

Intellectual Honesty & Rigor



Demonstrated: We possess a working recipe (Engineering Protocol) and have proven the solution's geometric fragility.
Refuted: The hypothesis that gradient covariance (κ) *causes* success was incorrect. It is a correlate, not a cause.
Lesson: We know *that* it works (the recipe), even if the theoretical *why* remains partially open.

The Reproducibility Crisis & Trajectory Engineering



The Problem: If a formal algorithm like Strassen requires such precise conditions ($B = 24-128$) to emerge, implicit tasks likely have similar const.

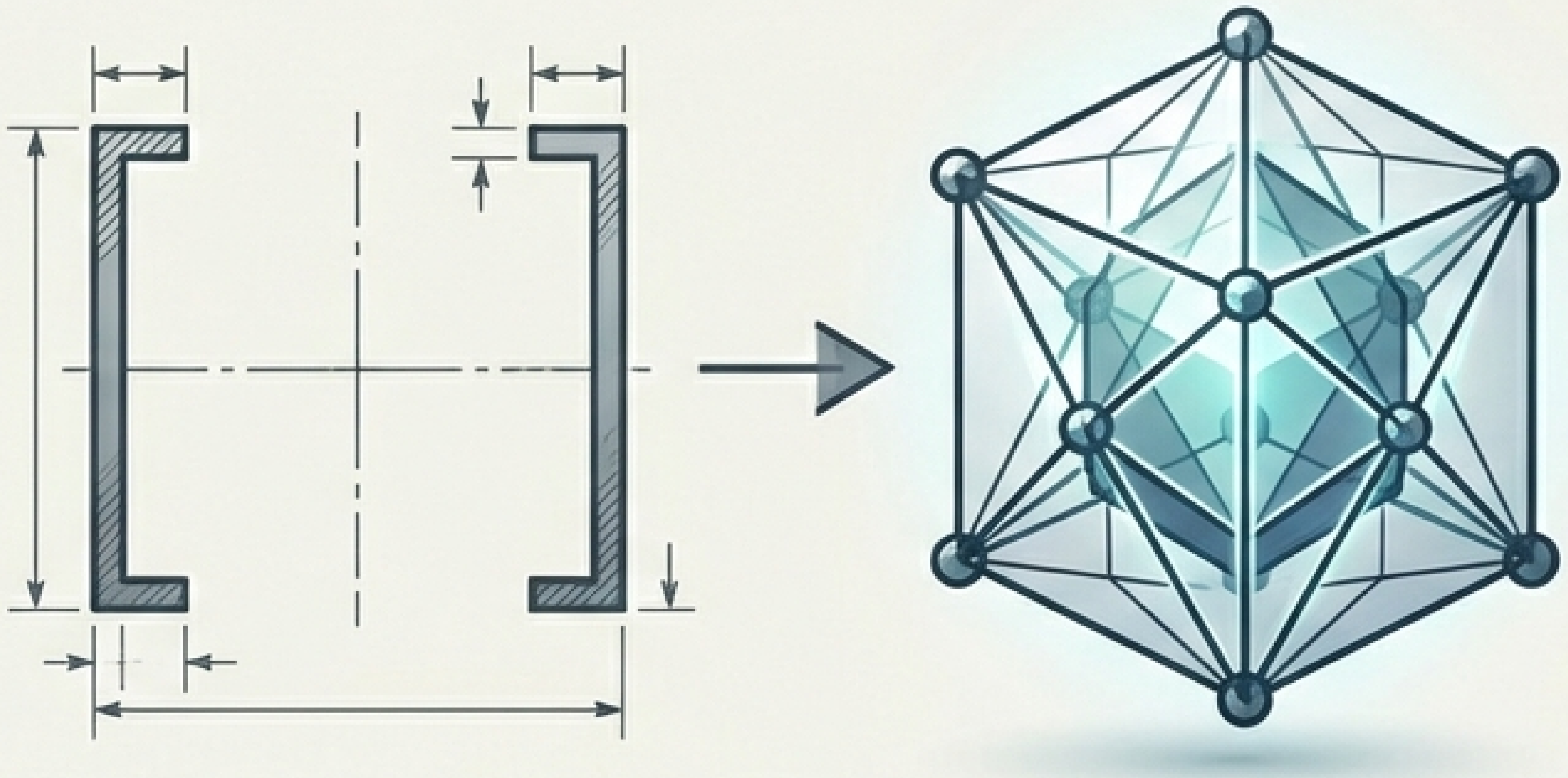
The Insight: Reproducibility isn't just about code sharing; it's about **Trajectory Engineering.**

Conclusion: Many "failed" experiments in Deep Learning might simply be trajectories that missed the narrow basin of the optimal solution.

Technical Paper
JetBrains Mono

From Alchemy to Materials Science

#F5F5F0
JetBrains Mono



Final Takeaways

- **The Recipe:** Protocol established with **68% yield** for inducing Strassen structure.
- **The Mechanism:** Batch size effect is empirically robust ($p<0.0001$) but theoretically open.
- **The Future:** Interpretable AI lies in **actively constructing** these "crystalline" structures, not waiting for them to emerge from noise.

Code & Data: github.com/grisuno/strass_strassen

DOI: 10.5281/zenodo.18263654