Scaling from 20M to 1.57B: What We Learned

Emergent Behaviors, Efficiency Breakpoints, and Scaling Dynamics
in the LOLM Architecture

Bryan Leonard & Brandyn Leonard
Qira LLC, Maricopa, AZ

Abstract

We report observations from training the Latent Order Language Model (LOLM) across four scales: 20.5M, 149M, 304M, and 1.57B parameters, spanning datasets from TinyStories to FineWeb-Edu. Across this range, we observe several emergent phenomena: a characteristic gate trajectory that reverses direction during training, regime codes that resist collapse at scale when trained with gradient isolation, a contrastive predictive coding (CPC) loss that transitions from random chance to near-perfect future-state prediction, and a dependency inversion between surface and latent pathways that intensifies with scale. We document these behaviors quantitatively and discuss what the scaling curves suggest about the architecture's trajectory at larger scales.

1. Introduction

Scaling laws for Transformer language models are well-characterized: loss decreases as a power law with model size, dataset size, and compute. But these laws describe a single architectural paradigm. When the architecture itself contains heterogeneous components—attention, recurrence, discrete codes, memory—the scaling dynamics become richer and less predictable. Components may activate at different scales, interact in unexpected ways, or exhibit phase transitions that pure Transformers do not.

LOLM provides a natural laboratory for studying these dynamics. With five distinct subsystems (surface decoder, latent SSM, regime layer, persistent memory, and manifestation gate), each trained by its own objective, the architecture offers multiple observable signals that reveal how the model's internal organization evolves across scale.

2. Training Configurations

We trained LOLM at four scales with progressively larger datasets and richer architectural configurations:

	20.5M	149M	304M	1.57B
Dataset	TinyStories	FineWeb-Edu	FineWeb-Edu	FineWeb-Edu
d_model	256	768	1024	2048
Decoder layers	6	12	16	24
SSM layers	2	2	4	6
Regime codes	16	24	32	64

Memory slots	32	64	128	256
Final train PPL	13.7	63.2	35.5	~34.5
Gate at convergence	0.27	0.55	0.83	0.71
Regimes alive	1/16	1/24	32/32	64/64

Table 1: LOLM configurations and key metrics across scales.

3. The Gate Trajectory: A Reversal in Two Acts

At every scale, the manifestation gate exhibits a characteristic two-phase trajectory. In the first phase, the gate moves toward 0 (latent-preferring): at 304M it drops to approximately 0.17 by step 1,800; at 1.57B it drops even further to 0.125 by step 2,000. During this phase, the model is leaning heavily on the latent SSM, and the learning rate is near its peak.

In the second phase, the gate reverses and climbs toward a surface-preferring equilibrium. At 304M it crosses 0.5 around step 3,300 and settles at 0.83 by step 5,000. At 1.57B it climbs more slowly and settles at 0.71 by step 20,000. The lower equilibrium at 1.57B means the model allocates more capacity to the latent path at larger scale—consistent with the latent SSM becoming more useful as model complexity increases.

The reversal is not a simple artifact of learning rate scheduling. It reflects a genuine change in how the model distributes computation: early in training, the SSM provides a useful inductive bias before the Transformer has learned meaningful attention patterns. As the Transformer matures, it reclaims the majority of the representation. But it never reclaims all of it—the gate never reaches 1.0—because the SSM captures structure that the Transformer cannot.

4. Regime Codes: Solving Collapse at Scale

Discrete codebooks in language models are notorious for collapsing. The token prediction loss, which is orders of magnitude larger than any auxiliary diversity loss, backpropagates into the codebook and drives all codes toward a single embedding. In LOLM's first training attempts, regime codes collapsed from 32 to 1 within 500–2,000 steps at every scale.

The solution was gradient isolation: detaching the regime embedding before it enters the fusion equation. The regime layer is then trained exclusively by its own objectives (changepoint alignment, load balancing, and entropy regularization). Combined with logit clamping at [-5, 5] and maintaining Gumbel-Softmax temperature above 0.5, this completely prevents collapse.

The scaling behavior is notable: at 20.5M and 149M (before gradient isolation was implemented), only 1 code survived. At 304M and 1.57B (with gradient isolation), all 32 and all 64 codes respectively remain active with near-maximum usage entropy throughout training. The fix is binary: without it, collapse is total; with it, diversity is complete.

5. CPC: From Random Chance to Precise Prediction

The contrastive predictive coding (CPC) objective trains the latent SSM to predict future decoder states from current latent states. The CPC loss was stuck at random chance (5.55 = ln(256)) for all training runs until the

introduction of SimCLR/CLIP-style projection heads.

The root cause was a dimensionality-temperature mismatch. In high-dimensional space (d=1024), random unit vectors have cosine similarity standard deviation of approximately 1/sqrt(d) = 0.031. With CPC temperature 0.1, the logit standard deviation was only 0.31—far too small for meaningful softmax discrimination over 256 positions. The fix: project both representations through separate 2-layer MLPs down to 128 dimensions, where cosine similarity std is 0.088. With temperature 0.07, logit std becomes 1.26—well within the learnable range.

After this fix, CPC loss dropped from 5.55 to 0.15 at 304M—a 37x reduction, confirming that the SSM encodes highly precise information about future decoder states. This is direct evidence that the latent path captures meaningful sequential structure, not noise.

6. Compute Efficiency: The 2x Result

In a controlled comparison at 1.57B parameters—same data, same tokenizer, same hardware, same training budget—LOLM reaches the baseline's step 20,000 perplexity (46.5) at approximately step 10,000. This represents roughly 2x compute efficiency: LOLM achieves equivalent modeling quality in half the training steps.

The advantage is largest early in training (42.8x at step 2,000) and narrows as both models converge, stabilizing at approximately 1.2–1.3x from step 13,000 onward. At step 20,000, LOLM achieves 38.7 perplexity versus the baseline's 46.5—a 16.8% advantage that shows no sign of closing. The gap between the models has been widening in absolute terms as training continues beyond step 20,000.

7. What the Scaling Curves Suggest

Several trends from 20M to 1.57B point in consistent directions. The gate equilibrium shifts toward more latent contribution at larger scale ($0.27 \rightarrow 0.55 \rightarrow 0.83 \rightarrow 0.71$). More regime codes activate and stay alive. The CPC loss decreases, indicating better future-state prediction. The dependency inversion between surface and latent paths intensifies.

If these trends extrapolate, we would expect a 7B LOLM to settle with an even lower gate value (more latent contribution), maintain all regime codes with higher usage entropy, and exhibit even stronger dependency on the latent pathway. The architecture appears to be entering a regime where the latent subsystem is not a supplement to the Transformer but a structural co-equal.

TPU-scale training at 7B and beyond will test whether these trends hold or saturate. These experiments are planned as the next phase of this research.

8. Conclusion

Training LOLM across four orders of magnitude reveals that hybrid architectures exhibit richer scaling dynamics than homogeneous Transformers. Components activate at different scales, gate trajectories undergo reversals, and the balance of power between surface and latent pathways shifts systematically with model size. These observations suggest that scaling hybrid architectures is not simply a matter of adding

parameters—it involves navigating an evolving landscape of inter-component dynamics that may unlock capabilities unavailable to any single paradigm.