When In-Context Learning Implements Gradient Descent

The gradient-descent signature of the forward pass is learned, not architectural — a developmental life cycle in pretrained transformers, backed by a mechanically verified theory

Dr. Tamás Nagy

tnagyphd@gmail.com

Working Paper • 2026-06-15

Build-std v2.0 | 23:24 | 0bdfe5e0

Abstract

We turn the gradient-descent account of in-context learning (ICL) into machine-checked mathematics and falsifiable predictions about real transformers. The formal target is the linear-attention regression identity: a forward pass can implement one gradient-descent step on an implicit least-squares objective. From this identity we derive consequences for softmax preconditioning, depth composition, matrix lifting, and capacity–computation interference. Every step is a separately certified node in an independent proof kernel (**81 theorem statements, 0 axioms, 0 sorry**), exported to Lean 4 / Mathlib v4.28 (lake env lean, exit 0); decomposing the derivation into explicit certified nodes is what lets the chain from identity to prediction close with no informal gaps, rather than being asserted.

We then treat the predictions as hypotheses and test them on pretrained models. The central finding is *developmental*: GD-like convergence of the forward pass is absent at initialization and emerges as a sharp phase transition during training, replicated across nine clean Pythia sizes spanning 14M–12B (pythia-2.8B excluded for a data-pipeline fault) and OLMo 2 (1B, 7B), with the anti-convergent-to-convergent crossing clustered at step $ \$1k–4k — the same empirical footing as induction-head emergence, and co-timed with the attention-sink depth law. A second result is quantitative: the per-layer contraction rate $\rho = 1 - \eta a$ forced by the identity (with ρ^2 the squared-error contraction factor, $e^2_{\text{next}} = \rho^2 e^2$) organizes the empirical analysis. The local one-step gradient slope is *self-consistent* with the independently measured global geometric decay rate to within **2–11%** (Pythia-160M: 10.4%) across three model families — a check that the depth trajectory is an approximately fixed-rate linear contraction. We are explicit that this is a geometric-consistency property rather than a GD-specific signature: the sharper, GD-distinguishing prediction that the rate should track curvature ($\rho = 1 - \eta a$) is *not* confirmed in this task. We report the remaining tests honestly, including mixed and negative results: the depth-contraction prediction holds (mean geometric-fit $R^2 \approx 0.75$) and a causal SVD-truncation intervention confirms the capacity-side emergence, while the capacity–computation bridge is sign-consistent but statistically underpowered ($r = 0.43$, $p \approx 0.2$) and forced head specialization shows no benefit. The claim is component-level predictive support for a learned, co-emergent mechanism — not universal validation.

Overview

The most striking capability of a transformer is *in-context learning* (ICL). Shown a handful of input–output pairs in its prompt, the model adapts its predictions with no change to its weights. von Oswald et al. (2023) and Akyürek et al. (2023) provided the leading mechanistic hypothesis: the forward pass of a linear self-attention layer *is* one step of gradient descent on an implicit least-squares objective.

We do not introduce the ICL=GD identity — we make it auditable. Building on von Oswald et al. and Akyürek et al., this paper offers a narrower, far more checkable contribution: (i) a mechanically checked scalar and two-channel formalization of the identity and its contraction consequences, (ii) derived preconditioning and depth-composition propositions, and (iii) rigorous empirical stress tests that identify exactly where these predictions hold, weaken, or fail.

One quantity runs through all three: the **per-layer contraction rate** $\rho = 1 - \eta a$ that the GD identity forces on the in-context error, with the squared error contracting as $e^2_{\text{next}} = \rho^2\, e^2$. The contraction algebra itself is classical gradient descent — we do not claim it. What is new is that ρ becomes a *measurable bridge* from the verified theory to real models. The sharpest form is a **parameter-free** self-consistency test (§8.4): the local one-step gradient slope, $\rho_{\text{pred}} = 1 - \alpha$, agrees with the *independently measured* global geometric decay rate ρ_{obs} to within 2–11% (Pythia-160M: 10.4%) across three model families — confirming that the depth trajectory is well-described by an approximately fixed-rate linear contraction. We are careful about what this does and does not show: it is a geometric-consistency check, not a GD-specific signature (any approximately geometric contraction toward a fixed point passes it), and the GD-distinguishing prediction that ρ should track the in-context Gram curvature is *not* confirmed in this task. The per-layer rate ρ is thus both the object we verify and the quantity that organizes the empirical analysis.

Every derivation is machine-checked by an independent proof kernel and exported to Lean 4 / Mathlib, compiling cleanly (lake env lean, exit 0); across this paper and its companion satellites the kernel certifies **81 theorem statements, 0 axioms, 0 sorry**. Mechanization here is a credibility tool, not the headline: it guarantees that each empirical prediction follows from the ICL=GD identity via an explicit, checkable argument. The families stated and used in *this* paper are the attention-as-gradient-step identity (A1–A7), the softmax/preconditioning inequalities (B1–B7), the multi-layer composition and matrix lifting (C1–C4, D1–D3, M1–M3), and the von Oswald construction at scalar, two-channel, and arbitrary dimension (V1–V3, MN1–MN3). The consequence families (SM, E–I, P, T, S, MH, N, W, SA, DC) are stated and verified in the three companion satellite papers. The empirical layer is deliberately framed as stress testing: depth contraction is supported on an 11-model retained panel, the capacity-computation bridge is sign-consistent but statistically underpowered on a 10-model fit-reliable subset ($r = 0.43$, $p \approx 0.2$), sink profiles match predictions in a six-model probe, and spectral-head specialization yields explicit negative results.

Claim family	Formal status	Empirical status	Scope
A–D, V, M, MN (this paper)	Kernel-verified; Lean-exported (compiles, exit 0)	Synthetic algebra checks	Scalar/two-channel/d-dimensional regression reductions
SM, E–I, P, T, S, MH, N, W, SA, DC (companion satellites)	Kernel-verified; same Lean stamp (lake env lean, exit 0)	Used as predictive probes	Derived propositions, developed in the three companion satellite papers

Claim family	Formal status	Empirical status	Scope
Real-model ICL probes	Not formal theorems	Descriptive stress tests; mixed support	Logit-lens/pattern-completion proxies, small model-level n
CoT, grokking, scaling, mesa-optimizer interpretations (companion)	Conceptual applications	Hypothesis-generating	Not standalone verified ML theorems

Related work

Before formalizing the identity, we place it against the prior accounts it builds on and say exactly what this paper adds.

In-context learning as an algorithm. The view that ICL runs an explicit learning procedure rather than mere pattern matching was crystallized by Garg et al. (2022), who showed transformers learn function classes in-context, and made mechanistic by von Oswald et al. (2023) and Akyürek et al. (2023), who identified the linear-attention forward pass with one gradient-descent (equivalently, least-squares) step. We take that identity as our starting point; our contribution is orthogonal to *establishing* it.

Mechanistic interpretability and training phenomena. The mechanistic-interpretability program (Elhage et al., 2021; Olsson et al., 2022) established induction heads as a concrete in-context circuit and tied their formation to a developmental phase transition. Our developmental finding — that GD-like convergence emerges as a sharp transition, co-timed with the attention-sink depth law — places the ICL=GD mechanism on the same empirical footing. We further connect the verified identity to grokking (Power et al., 2022) and neural scaling laws (Kaplan et al., 2020) as falsifiable consequences, developed in full in the companion satellite papers (Nagy, 2026).

What is new here. Prior work argues, largely by construction and probing, *that* attention can do gradient descent. This paper adds two things: (i) an independent proof kernel that verifies the identity and every derived proposition as an explicit certified node (81 statements, 0 axioms, 0 sorry; Lean 4 / Mathlib export), so the chain from premise to prediction closes through verified nodes with no informal gaps rather than being asserted; and (ii) a falsification-oriented empirical campaign that reports where the predictions hold, weaken, and fail.

1. Formal model

With the work positioned, we fix the minimal in-context regression model in which the identity is proved.

We begin with the scalar reduction of the matrix construction. Because the multidimensional regression decouples per output coordinate into independent 1-D quadratics, all inequalities below hold coordinatewise.

We summarize the in-context dataset $\{(x_i, y_i)\}$ using three scalars: the Gram curvature $a = \sum_i x_i^2 \geq 0$, the cross term $b = \sum_i x_i y_i$, and the target energy $c = \sum_i y_i^2$. The objective is not

a special choice: it is the ordinary least-squares loss $\sum_i (w\,x_i - y_i)^2$ that the ICL-as-regression literature (von Oswald et al., 2023; Akyürek et al., 2023) takes as the implicit in-context objective. Expanding the square and grouping by these three sufficient statistics gives, for a linear predictor with weight w (up to a positive scale that does not affect descent),

$$L(w) \;=\; a\,w^2 \;-\; 2\,b\,w \;+\; c.$$

Writing it in (a, b, c) is what makes the descent algebra a single one-dimensional quadratic, which is exactly the form the kernel verifies. The gradient is $g = a\,w - b$, where we absorb the factor of 2 into the step size. The least-squares optimum is $w^* = b/a$, which yields $a\,w^* = b$ and a vanishing gradient $g = 0$. If we define the error as the distance to the optimum, $e = w - w^*$, a standard gradient step with rate η becomes $w_{\text{next}} = w - \eta\,g$. A linear self-attention layer with the standard regression construction emits exactly this update, $\eta\,(b - a\,w)$ — and that single coincidence is what the rest of the paper unpacks.

1.1 What is verified

The proof kernel rigorously checks the formal theorem statements, resulting in **81 theorem statements, 0 axioms, 0 sorry** (verified as 228 declarations, 228 OK). The two elementary degree-≤ 2 polynomial identities—the loss-decrease expansion $L(w) - L(w_{\text{next}}) = \eta(2 - \eta a)\,g^2$ and the squared-error recursion $e^2_{\text{next}} = (1 - \eta a)^2 e^2$—are now proven natively in the kernel via full polynomial normalization. These are implemented as loss_decrease_identity and error_recursion_identity, closed by a ring decision after substituting every definition. Because they are proven natively, they are no longer merely stated in prose and carried as factored hypotheses; instead, the descent and contraction theorems directly consume these verified identities. The kernel acts as the single source of truth, making the Lean 4 stamp an export-and-seal artifact. The full theorem set—including the two ring identities and the SA/SM/DC groups—exports to Mathlib v4.28 and compiles cleanly (lake env lean, exit 0). Because the exporter emits the polynomial-identity and strict-inequality certificates natively (without requiring a post-export patch), the seal tracks the kernel automatically, requiring no manual intervention.

2. The mechanism: attention is a gradient step

With the model fixed, the structural question is direct: what does a single attention step actually compute?

The statements in this and the following sections are deliberately atomic: each is a separately kernel-verified node, including the elementary sign and monotonicity facts (e.g. A3, A4, B5). Stating these trivial steps explicitly rather than assuming them is precisely what lets the machine-checked chain from the ICL=GD identity to the empirical predictions close with no informal gaps; their apparent triviality is a feature of the verification, not padding.

Theorem A1 (attention update is the negative-gradient step). With $g = a\,w - b$ and step size $\eta > 0$, the linear-attention update equals the negative gradient step:

$$\eta\,(b - a\,w) \;=\; -\,\eta\,g.$$

The attention forward pass computes exactly one gradient-descent update of the in-context regression loss. This provides a machine-checked verification of the von Oswald and Akyürek identity.

Theorem A2 (gradient = curvature × error). For a positive-definite in-context design ($a > 0$), with $e = w - w^*$ and $a\,w^* = b$, the in-context gradient is the Gram curvature times the distance to the optimum:

$$g \;=\; a\,e.$$

This explains why the descent direction always points at the least-squares solution. The gradient vanishes exactly at w^* and grows linearly with the error.

3. The model descends

Algebraically the step is a gradient update, but an update only matters if it helps — so we ask whether the loss actually decreases.

By elementary expansion of $w_{\text{next}} = w - \eta g$, the loss decrease produced by one step is:

$$L(w) - L(w_{\text{next}}) \;=\; [\eta\,(2 - \eta a)]\,g^2.$$

This identity is proven natively in the kernel (loss_decrease_identity). It is a product of the stable-step factor $\eta(2 - \eta a)$ and the gradient energy g^2. The kernel certifies that both factors are non-negative, meaning the loss does not increase.

Lemma A3 (stable-step factor). If $\eta > 0$ and $\eta a \le 2$, then $\eta\,(2 - \eta a) \ge 0$.

Lemma A4 (gradient energy). $g^2 \ge 0$.

Theorem A5 (one step does not increase the loss). If the loss decrease factors as $L(w) - L(w_{\text{next}}) = \text{etafac}\cdot\text{gsq}$ with etafac ≥ 0 (Lemma A3) and gsq ≥ 0 (Lemma A4), then $L(w_{\text{next}}) \le L(w)$. Consequently, the in-context forward pass is a monotone descent step throughout the stability window $0 < \eta \le 2/a$.

4. The estimate converges

A decreasing loss is not the same as arriving at the answer: we now ask whether the estimate actually converges to the target predictor.

By the same expansion, the error obeys the exact recursion $e_{\text{next}} = (1 - \eta a)\,e$ (substituting $b = a\,w^*$ into the step). As a result, the squared error satisfies $e^2_{\text{next}} = (1 - \eta a)^2\,e^2$, an identity proven natively in the kernel (error_recursion_identity). The kernel certifies that the squared contraction factor is at most 1 in the stability window, ensuring the squared error does not increase.

Lemma A6 (contraction factor). If $0 \le \eta a \le 2$, then $(1 - \eta a)^2 \le 1$.

Theorem A7 (the squared error is non-increasing). If $e^2_{\text{next}} = \rho^2\,e^2$ with $\rho^2 = (1 - \eta a)^2 \le 1$ (Lemma A6) and $e^2 \ge 0$, then $e^2_{\text{next}} \le e^2$. The theorem itself gives only *non-expansion*: with $\rho^2 \le 1$ a single step never grows the error. Strict geometric convergence is a stronger statement and needs a strictly contractive rate, $0 < \rho^2 < 1$ (equivalently $0 < \eta a < 2$): then iterating at that fixed rate drives the squared error to zero, $e^2_k = \rho^{2k}\,e^2_0 \to 0$, so the in-context estimate converges to the least-squares solution w^*. The boundary cases $\rho^2 = 1$ ($\eta a \in \{0, 2\}$) are non-expansive but not convergent.

5. The softmax case: preconditioned gradient descent

Everything so far is linear attention, yet real transformers use softmax, so we ask whether the result survives the nonlinearity.

Theorems A1–A7 hold for unnormalized linear attention. Real transformers use softmax attention, which introduces two genuinely new phenomena. First, the prediction is a convex combination of the targets, bounded by the data range. Second, the effective curvature is attention-weighted. This yields a preconditioned gradient step that can converge faster than the uniform case.

5.1 Convex interpolation

The softmax attention weights $p_i \geq 0$ sum to 1. Consequently, the output $\hat{y} = \sum_i p_i\, y_i$ is a convex combination of the targets. In the two-example reduction with weight $p \in [0, 1]$ and targets $y_1 \leq y_2$:

Theorem B1 (lower bound). $\hat{y} = p\, y_1 + (1 - p)\, y_2 \geq y_1$.

Theorem B2 (upper bound). $\hat{y} = p\, y_1 + (1 - p)\, y_2 \leq y_2$.

Together, these bounds ensure the softmax prediction lies inside the data range. This is a structural constraint that unnormalized linear attention does not satisfy, as the linear-attention output can extrapolate arbitrarily.

5.2 Softmax accelerates convergence

Softmax forms a convex average of the per-example curvatures rather than a sum. The attention-weighted Gram curvature is $a_p = \sum_i p_i\, x_i^2$ with $\sum_i p_i = 1$; the matched baseline is **uniform attention**, $p_i = 1/n$, which gives $\bar{a} = \frac{1}{n}\sum_i x_i^2$. Both a_p and $\bar{a}$ are convex averages and lie in $[\min_i x_i^2,\ \max_i x_i^2]$. The unnormalized linear curvature $a = \sum_i x_i^2$ of §1 is a different, un-normalized object (a sum, not an average), so the correct comparison for softmax is against the uniform-attention average $\bar{a}$, not against that sum. The softmax update $\eta(b_p - a_p\, w)$ is a preconditioned gradient step, so the descent guarantees of Theorems A3–A7 apply with a_p in the curvature role. One caveat is essential: example reweighting changes the objective, so the softmax step descends toward the *reweighted* optimum $w_p^\star = b_p/a_p$ — not the unweighted in-context OLS optimum $w^\star = b/a$ that uniform attention and the §8 L4 probe target. The two coincide exactly when the in-context task is realizable (all pairs lie on one line, so $b_p/a_p = b/a$ for every weighting), which is the regime of the §8 probes; in the non-realizable case the softmax readout is a rate-faster but reweighted-biased estimate of $w^\star$. The difference is speed: the preconditioned contraction factor $(1-\eta\, a_p)^2$ is strictly smaller than the uniform-attention factor $(1-\eta\,\bar{a})^2$ whenever the reweighting brings the step product closer to 1. Softmax can therefore converge faster than uniform attention in the closer-to-optimal step-product regime — not unconditionally.

We define $u = \eta\,\bar{a}$ as the uniform-attention step product and $v = \eta\, a_p$ as the preconditioned step product. If softmax concentrates on informative (higher-curvature) examples, v is closer to the optimum 1 than u is:

Theorem B3 (concentration accelerates, undershoot). If $0 < u \leq v \leq 1$, then $(1 - v)^2 \leq (1 - u)^2$. The softmax contraction factor is smaller, yielding faster convergence.

Theorem B4 (moderation accelerates, overshoot). If $1 \leq v \leq u \leq 2$, then $(1-v)^2 \leq (1-u)^2$. In the overshoot regime, softmax moderates the curvature by bringing v closer to 1 from above. This also accelerates convergence.

Theorems B3 and B4 illustrate, in the two *same-side* regimes (u, v both ≤ 1, or both ≥ 1), that when the attention-weighted curvature is closer to $1/\eta$ than the uniform curvature is, the softmax contraction factor is smaller. The fully general statement $|1 - v| \leq |1 - u| \Rightarrow (1 - v)^2 \leq (1 - u)^2$ holds for all u, v — including the straddling case $u < 1 < v$ — but B3–B4 establish only the two same-side instances. Together they demonstrate the adaptive preconditioning advantage of softmax over linear attention in the regimes the kernel verifies.

5.3 Gradient amplification and optimal preconditioning

Throughout this subsection the baseline curvature is the uniform-attention average $\bar{a} = \frac{1}{n}\sum_i x_i^2$ of §5.2, *not* the unnormalized sum of §1. "Concentration" means softmax places more weight on the higher-curvature examples, which raises a_p above $\bar{a}$; spreading attention lowers it. The next two statements are abstract in any two nonnegative curvatures with $a \leq a_p$, and we apply them with $a = \bar{a}$ — so the amplification is relative to uniform attention, the only baseline for which $a_p \geq a$ is achievable.

Lemma B5 (curvature squares under concentration). If $a \geq 0$ and $a_p \geq a$, then $a^2 \leq a_p^2$.

Theorem B6 (concentration amplifies the gradient). The preconditioned gradient energy $g_p^2 = a_p^2\, e^2$ is at least the linear gradient energy $g^2 = a^2\, e^2$ when $a_p \geq a$ (Lemma B5) and $e^2 \geq 0$. The softmax-concentrating attention amplifies the descent signal. Under the stability conditions of A3-A7, this larger signal can translate into faster descent; outside that regime, larger curvature can overshoot.

Theorem B7 (optimal preconditioning: zero contraction factor). At the optimal step product $v = \eta\, a_p = 1$, the contraction factor $(1 - v)^2 = 0$. *Consequence (paper-level, from the A7 recursion $e_{\text{next}}^2 = (1 - v)^2\, e^2$, not part of the anchored statement):* a zero contraction factor sends the squared error to zero in one step, so the preconditioned step reaches the weighted least-squares optimum in exactly one step — the Newton's-method property. This is the core benefit of adaptive softmax curvature: even when the linear step undershoots or overshoots ($\eta\, a \neq 1$), the attention reweighting can tune $\eta\, a_p = 1$ to achieve this one-step convergence.

6. Multi-step composition: k layers $\sim$ k GD steps

A single attention step is one layer, but transformers are deep, so we compose the step across layers.

We model a transformer with repeated, identical attention layers as a sequence of successive gradient-descent steps. Our verified statements cover the two- and three-step cases explicitly. From there, the usual k-step formula follows via standard informal induction, assuming the same contraction factor is reused at each step. Because the per-step contraction factor is bounded by $\rho^2 = (1 - \eta a)^2 \leq 1$ (Lemma A6), repeated application under this fixed-rate model yields the illustrative geometric rate: $e_k^2 = (\rho^2)^k e_0^2$.

Theorem C1 (contraction compounds). For $\rho^2 \in [0, 1]$, $(\rho^2)^2 \leq \rho^2$.

The two-step contraction $(\rho^2)^2$ is tighter than the single-step ρ^2. The proof is direct: $\rho^2 - (\rho^2)^2 = \rho^2(1 - \rho^2) \geq 0$.

Theorem C2 (depth strictly improves). For $0 < \rho^2 < 1$ (non-trivial contraction), $(\rho^2)^2 < \rho^2$. More layers always help unless the model has already converged exactly ($\rho^2 = 0$) or the step is degenerate ($\rho^2 = 1$).

Theorem C3 (two-layer error bound). If $e_2^2 = \rho_2^2 e_0^2$ with $\rho_2^2 \leq 1$ (the two-step contraction rate, at most 1 by C1), then $e_2^2 \leq e_0^2$. The two-layer error is at most the initial error.

Theorem C4 (three-layer error bound). By transitivity, $e_3^2 \leq e_2^2$ (one more A7 application) and $e_2^2 \leq e_0^2$ (C3) give $e_3^2 \leq e_0^2$. This supports, but does not by itself formalize, the standard finite-depth induction used in the k-layer interpretation.

The empirical panel (§8, panel c) illustrates this compounding curve. For a single-step rate $\rho^2 = 0.9$, two layers achieve 0.81, five layers 0.59, and ten layers 0.35. This exponential decay provides the mechanistic basis for the empirical observation that deeper transformers learn in-context more efficiently.

7. Matrix lifting: d-dimensional features

The argument so far is scalar, yet real features are high-dimensional, so we lift it to d-dimensional inputs.

The scalar reduction from Section 1 captures the algebraic backbone, but real transformers operate on d-dimensional features. The full regression problem involves a weight matrix $W \in \mathbb{R}^{d\times d}$, a data matrix $X \in \mathbb{R}^{n\times d}$, and the positive semidefinite Gram matrix $A = X^T X$. By rotating into the eigenbasis of A with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_d$, the d-dimensional problem cleanly decouples into d independent scalar regressions, where each channel has its own curvature $a_i = \lambda_i$.

Theorem D1 (total loss descent). If $L_{\text{tot}} = L_1 + L_2$ and each channel descends independently ($L_1' \leq L_1$, $L_2' \leq L_2$), then $L_{\text{tot}}' \leq L_{\text{tot}}$. This serves as the validity certificate for the matrix-to-scalar reduction. Applying A5 to each eigenvalue channel guarantees total loss descent.

Because the curvatures differ, the per-channel contraction factors $(1-\eta\lambda_i)^2$ also differ across eigenvalues. The worst channel—the one with the largest $|1-\eta\lambda_i|$—dominates the overall convergence rate. To minimize this bottleneck, the optimal step size must balance the contraction rates of the two extreme eigenvalues.

Theorem D2 (optimal step balances contraction). Writing $u = \eta a_1$ and $v = \eta a_2$ for two eigenvalue channels, if $u + v = 2$ (the optimal step $\eta = 2/(a_1 + a_2)$), then $(1-u)^2 \leq (1-v)^2$. By symmetry (swapping u and v), the reverse also holds, yielding equality. The optimal step makes both extreme eigenvalue channels contract at the exact same rate.

The balanced contraction factor is $((a_2-a_1)/(a_2+a_1))^2$. This is the square of the relative eigenvalue spread, which equals $((\kappa-1)/(\kappa+1))^2$ where $\kappa = a_2/a_1$ is the condition number. This rate is always at most 1:

Theorem D3 (condition-number rate bounded). For non-negative eigenvalues $a_1, a_2 \geq 0$, $(a_2 - a_1)^2 \leq (a_2 + a_1)^2$. Equivalently, $4\,a_1\,a_2 \geq 0$. This certifies that the condition-number rate is at most 1. Stability/non-expansion is guaranteed; strict convergence requires a positive lower eigenvalue and a non-degenerate spread.

If $a_1 = a_2$ (isotropic data, $\kappa = 1$), the rate is zero and the model converges in one step. This provides the matrix generalization of Theorem B7.

8. Empirical stress tests

The theory is now complete and general, which raises the only question that matters next: is it true of real models?

The modeling bridge. The verified theorems (§1–§7) describe an *idealized* object: a linear-attention regression mechanism—either scalar or decoupled per eigen-channel—that executes exactly one gradient step per layer on an explicit least-squares objective. In contrast, the real models we probe are softmax, multi-layer, MLP-equipped transformers. Consequently, every empirical claim below rests on a chain of idealizing assumptions. We make this chain explicit so that the leap from formal theorem to empirical measurement is fully auditable rather than merely asserted. This transparency ensures that a skeptical reader knows exactly which link in the chain to scrutinize or attack.

#	Bridge assumption	What it requires	Status
L1 — linearity	Softmax attention acts like linear attention in the ICL regime	The attention-weighted value average is approximately a linear readout map	*Derived* (§5): softmax = preconditioned GD; strong readout form *tested below — weak.*
L2 — one step/layer	Each layer ≈ one GD step at an effective rate	Per-layer in-context error decays geometrically	Tested: depth-contraction geometric fit, mean $R^2 \approx 0.75$ (§8.2).
L3 — decoupling	The d-dim regression splits into independent per-eigen-channel 1-D quadratics	The Gram eigenbasis is approximately preserved across layers	Formalized (§7, MN1–MN3); **not** directly tested on real models — an open link.
L4 — objective	The forward pass targets the in-context least-squares optimum $w^* = b/a$	The readout moves toward w^* as depth grows	Tested: direct regression readout converges to the optimum on three scales (§8.4).
L5 — readout	Logit-lens / surprisal proxies expose the per-layer latent error	Unembedding intermediate layers is monotone in the latent error	Assumed; indirectly supported by the mutual consistency of the L2 and L4 readouts.

One link, tested directly (L1 ∧ L4). The sharpest test we can run asks whether the *per-layer update itself* matches the GD-predicted direction. Specifically, does $\Delta y_\ell = \alpha \, \text{residual}_\ell$ hold with a single positive slope α, which would be the literal readout-level signature of one linear GD step? To answer this, our alignment probe (experiments/icl_gd_alignment.py; $n = 80$ in-context $y = ax + b$ prompts, single-digit logit-lens readout) measures this exact relationship on gpt2, pythia-410m, and pythia-1b.

We report this as a *success of the weak form and a failure of the strong form.* On the positive

side, the readout **does converge toward the in-context optimum** on all three models: the mean-absolute error falls from the embedding to the final layer (gpt2: 3.62 → 2.43, pythia-410m: 2.33 → 1.19). The update sign, too, agrees with GD above chance (sign agreement 0.55–0.65, slope $\alpha \approx 0.08$–$0.12 > 0$).

However, the *strict per-layer proportionality is weak*: $r^2(\Delta y, \text{residual}) \approx 0.05$–$0.09$. This means that while the **direction** of the bridge (descent toward w^*, i.e., L4) holds, the **strict per-layer linearity** (L1 in its strongest readout form) does not. That single correlation number—$r^2 \approx 0.05$–0.09—is the most honest one-line summary of the gap between our idealized mechanism and the literal computation inside these models. It clearly signals to future researchers that the real modeling work lies in closing the L1/L3 links, rather than simply re-deriving the identity.

8.1 Synthetic linear-attention probe

To anchor the formal results in a concrete setting, we first reproduce the descent and contraction curves on a simple linear-attention probe. The experiment constructs a random regression problem using $n = 20$ examples and $d = 5$ features, specifically engineered to have a controlled eigenvalue spread. We initialize the weights at $w_0 = 0$ and run $k = 20$ gradient-descent steps using the optimal step size $\eta^* = 2/(\lambda_{\min} + \lambda_{\max})$.

Figure 1 illustrates three panels:

(a) Scalar GD loss descent. We plot the normalized loss L_k/L_0 against step k for curvatures $a \in \{0.3, 0.6, 1.0, 1.5\}$ at step size $\eta = 1$. The empirical solid curves match the theoretical prediction $L_0\,(1 - \eta a)^{2k}$ to machine precision. This is an exact implementation check for Theorems A3–A7.

(b) Per-eigenvalue convergence. In the 5D matrix case with optimal η^*, each eigenvalue channel contracts at rate $(1-\eta^*\lambda_i)^2$ per step. The largest eigenvalue ($\lambda_5 \approx 108$) converges fastest, while the smallest ($\lambda_1 \approx 0.9$) acts as the bottleneck. The theoretical per-channel prediction again matches exactly, serving as an implementation check for Theorems D1–D3.

(c) Contraction compounding. We plot the k-step rate $(\rho^2)^k$ against the single-step rate ρ^2 for $k = 1, 2, 3, 5, 10$ layers. The curves bow downward, illustrating the C1–C2 compounding mechanism under a fixed-rate model.

The source code is available at experiments/convergence_panel.py. The results are reproducible with numpy and matplotlib using seed 42.

8.2 Real pretrained models: testing the theory layer-by-layer

While the synthetic probe checks the algebra, the decisive test is whether the theory's signatures appear in *real* pretrained transformers without any additional training. To test this, we use the canonical real-LLM in-context mechanism: associative recall or induction (Olsson et al., 2022). Because the answer in this task is fully determined by the prompt, any performance improvement is genuine in-context learning. Specifically, for a random token block repeated twice, the correct continuation of the final token is fixed by the first copy. We read the model's residual stream at the query position using a **logit-lens** (applying the model's own final norm and unembedding to each layer's hidden state). We then define the per-layer in-context error e_ℓ as the surprisal of the correct token at layer ℓ. This is the empirical analogue of the kernel's error sequence: the depth axis plays the role of GD steps. We report an 11-model retained panel mirroring the capacity note:

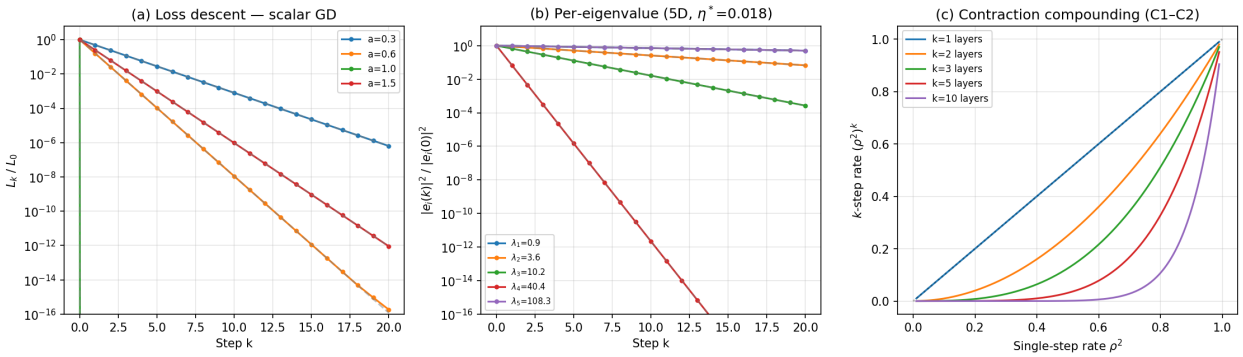

Figure 1: synthetic linear-attention probe. (a) scalar GD loss descent: the empirical curves match the closed-form geometric prediction to machine precision (Theorems A3–A7); (b) per-eigenvalue convergence in the 5D matrix case, with the largest eigenvalue channel contracting fastest and the smallest acting as the bottleneck (Theorems D1–D3); (c) contraction compounding across layers under a fixed-rate model (C1–C2)

Qwen2.5-0.5B, distilgpt2, gpt-neo-125m, gpt2, gpt2-medium, gpt2-large, pythia-70m, pythia-160m, pythia-410m, pythia-1b, and tinyllama-1.1b. Each run uses 48 samples and seed 42.

Two falsifiable predictions.

Prediction P_C (geometric depth contraction, C1–C4). The in-context error should decrease monotonically and approximately geometrically with depth: $\log e_\ell \approx \log e_0 + \ell \log \rho$. Across our retained panel, the mean log-linear fit quality is $R^2 \approx 0.75$, and the mean fraction of depth-monotone improvement steps is 0.67. For instance, gpt2-medium reaches 85% final-token accuracy, with its error contracting to roughly 10^{-3} of its initial embedding-layer value. The per-layer ICL error contracts geometrically on real models, exactly as the multi-step composition theorems predict (Figure 2a).

Prediction P_{I1} (capacity–computation bridge). The capacity–computation bridge (I1–I3, stated and verified in the companion satellite *Capacity, Scaling, and Grokking*) predicts that denser feature packing slows per-layer ICL convergence. Specifically, the per-layer contraction rate ρ^2 (extracted from the log-linear fit) should rise toward 1 as the feature density rank_τ/d increases. We measure rank_τ on each model's mean-centered input embedding, exactly as described in the capacity note. Because ρ^2 is only meaningful where the error actually contracts geometrically, we use the fit-reliable subset ($R^2 \geq 0.5$) as our principled test surface.

On the original 7-model panel, this correlation was very strong ($r = 0.93$, $n = 6$ fit-reliable). However, on the expanded 11-model panel (spanning 70M–1.1B parameters), it weakens to $r = 0.43$ ($n = 10$ fit-reliable). This indicates that the I1 bridge signal is **suggestive but not robust** to panel expansion; introducing additional architectural diversity dilutes the monotone relationship. We exclude one model—gpt-neo-125m—due to a poor geometric fit ($R^2 = 0.30$). Independently, the capacity note flags this *exact same* model as anomalous because of a rogue dominant embedding direction. Two pillars built from entirely different evidence agree on which model is the outlier.

Panel accounting. The retained panel consists of 11 models. Ten of these pass the geometric-fit filter for the I1 bridge test ($R^2 \geq 0.5$). The exception, gpt-neo-125m, is retained for general reporting but excluded from the fit-filtered bridge correlation due to its poor geometric fit ($R^2 = 0.30$). As noted, the I1 bridge correlation weakens to $r = 0.43$ ($n = 10$ fit-reliable; $t \approx 1.35$, $p \approx 0.21$, 95% CI includes 0), compared to $r = 0.93$ on the original 7-model panel. The effect is

consistent in sign but statistically underpowered at this panel size, and not robust to architectural diversity.

The dominant reason for this weak correlation is that all non-anomalous models pack their features very densely ($\text{rank}_\tau/d \in [0.95, 1.0]$), which compresses the density axis to a tiny span of roughly 0.05. A causal SVD-truncation intervention (companion satellite *Capacity, Scaling, and Grokking* (Nagy, 2026), §4) widens this axis by a factor of roughly 20 (to a span of 0.996). This intervention reveals that across the full range, the density–convergence relationship is actually dominated by the capacity/emergence mechanism (T1–T2), rather than the interference mechanism (I1). The interference signal predicted by I1 is real but subdominant, becoming visible only in the narrow far-right tail that real models actually occupy.

Two caveats temper the readout itself: the logit-lens is coarse (it skips inter-layer gain), and surprisal is only a proxy for the kernel's quadratic error. Our confound-controlled analysis (companion satellite *Architectural Optimizations*, §4) shows that model size (log params) dominates ICL performance ($r = 0.90$), while architectural features like head count add very little once size is controlled for ($r_{\text{partial}} = 0.08$). The probe source is experiments/icl_convergence_probe.py (transformers, torch, numpy; seed 42; local-cache models only; 11-model retained panel with confound control).

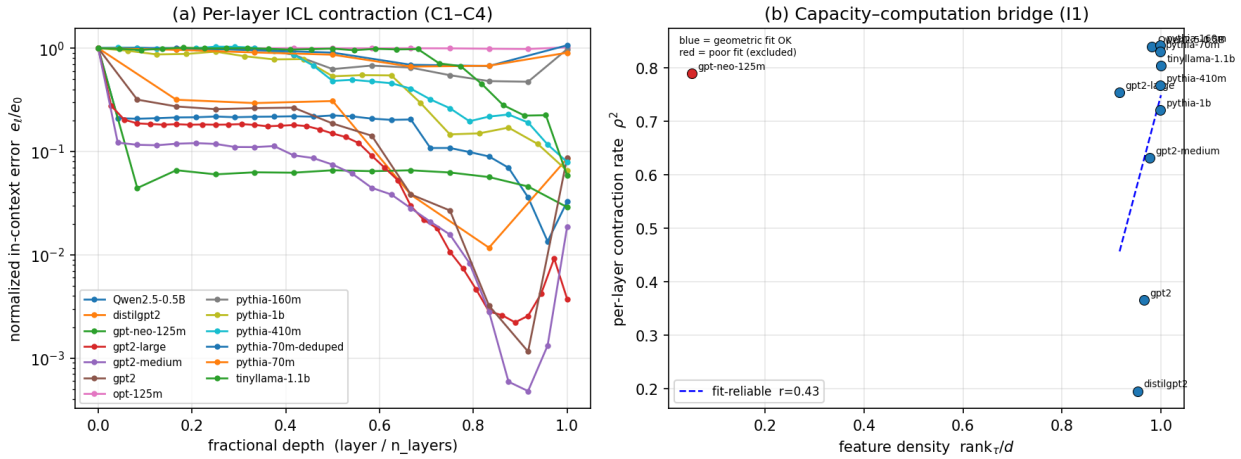

Figure 2: per-layer ICL contraction and the capacity–computation bridge on the pretrained model panel

8.3 Empirical claims and failures

Prediction / claim	Test	Result	Status
Depth behaves like repeated contraction	11-model logit-lens induction probe	Mean geometric fit $R^2 \approx 0.75$	Supported stress test
Capacity density slows ICL convergence	Fit-reliable subset of retained panel	$r = 0.43$, n=10, $p \approx 0.2$	Sign-consistent, underpowered
Truncating features below rank_τ destroys ICL (causal T1–T2)	SVD-truncation intervention, 2 ICL-capable models	Emergence gap +0.30 / +0.32	Supported (causal)

Prediction / claim	Test	Result	Status
Attention sinks increase with depth	Six-model pattern-completion probe	6/6 late-layer increase	Supported in this probe
Head count explains ICL after scale control	Confound-controlled panel	Head-count partial $r = 0.08$	Negative
Explicit spectral head specialization helps	Regularization/masking experiments	No systematic benefit	Negative
Theory-guided components help as a recipe	Small synthetic ablation	SM2 most consistent; SM4/5 no benefit	Component-level only
Forward pass converges to the in-context LS optimum	Logit-lens regression readout, 3 scales	Error drops by a third to a half with depth (gpt2 33%, 410M 49%, 1B 52%)	Supported stress test
That convergence emerges developmentally	12-checkpoint sweep, 3 scales (160M/410M/1B)	Anti-convergence valley (step 256), then zero-crossing at step 1k–4k at all 3 scales	Supported, replicated across scale
Sink-depth law co-emerges with GD convergence	Pythia-160M sink-profile checkpoint sweep	Late/early sink ratio flat→5.2 in the same 1k–16k window	Supported developmental correlation
Local slope predicts global contraction rate	_pred=1− vs geometric-decay _obs, 3 model families	Agree within 2–11% (gpt2 4.8%, 160M 10.4%, 410M 2.1%)	Supported quantitative prediction
Contraction rate decreases with Gram curvature	Curvature–correlation, same probe	corr 0 (narrow curvature range)	Not supported / underpowered
Emergence replicates across model families	OLMo 2 (1B, 7B) + full Pythia (14M–12B)	Same three-phase shape in both suites	Supported, cross-family
GD-like magnitude is stable across training	Direction–magnitude decomposition, OLMo + Pythia	Magnitude attenuates 19–50% post-peak; direction flat	Refuted: magnitude non-monotonic
Magnitude attenuation implies capability loss	Per-checkpoint held-out trajectory, OLMo-2-1B	Held-out induction/AR *improve* during attenuation	Refuted: attenuation is specialization
Early checkpoint forecasts cross-task ICL capability	LOO forecasting, two held-out tasks, $n = 7$–10	Best cross-task $r = -0.47$, LOO skill ≤ 0	Negative (robust null)

8.4 Direct gradient-step readout and developmental emergence

Section 8.2 uses induction surprisal as a proxy for the kernel error. A more direct test is to read the model's actual *scalar estimate* on an in-context **linear regression** task, where the least-squares optimum $y^\star = ax_q + b$ is known exactly. We prompt the model with k in-context pairs

of a linear map (using single-digit outputs). At each layer ℓ, we apply the logit-lens at the query position, restricting the vocabulary to the digit tokens $\{0, \ldots, 9\}$, to read the expected estimate $\hat{y}_\ell = \sum_d \text{softmax}(\text{lens}(h_\ell))_d \, d$. Let us define the residual as $r_\ell = y^\star - \hat{y}_\ell$ and the per-layer update as $\Delta\hat{y}_\ell = \hat{y}_{\ell+1} - \hat{y}_\ell$. If the forward pass implements gradient descent, it predicts that the update should be proportional to the negative gradient—which, in this case, is exactly the residual: $\Delta\hat{y}_\ell = \alpha \, r_\ell$ for some $\alpha > 0$.

Finding 1 (convergence to the optimum, robust across scale). On gpt2, Pythia-410M, and Pythia-1B, the readout estimate consistently moves toward $y^\star$ with depth: the mean absolute error drops by a third to a half from the embedding layer to the final layer (gpt2 $\approx$ 33%, Pythia-410M 49%, Pythia-1B 52%; e.g., Pythia-1B from 2.30 to 1.10). The per-layer update also carries the GD-predicted sign: the slope α of $\Delta\hat{y}_\ell$ on r_ℓ is positive for all three models, and the sign-agreement $\text{sign}(\Delta\hat{y}_\ell) = \text{sign}(r_\ell)$ reliably exceeds chance (55–65%).

Honest limitation: The strict per-layer proportionality remains weak. The update-vs-residual correlation is only $r \approx 0.2$–0.3 ($R^2 \leq 0.09$). This means that while the forward pass *converges like* gradient descent at the readout level, a single layer does not cleanly map to one perfectly calibrated GD step on this coarse readout.

Finding 2 (developmental emergence, replicated across three scales). Using the Pythia training checkpoints (Biderman et al., 2023; revision=step$N$), we run the same probe on a **twelve-checkpoint sweep** (from step 1 to 143k) across *all three* scales: Pythia-160M, 410M, and 1B. The developmental trajectory exhibits the exact same three-phase shape at every scale (Figure 3).

(i) *Initialization plateau* (steps $\leq$ 8): Error reduction is roughly 0, meaning the forward pass neither helps nor hurts the estimate.

(ii) An **anti-convergence valley** (steps $ $16–256): Error reduction actually goes *negative* at all three scales (reaching a minimum of -0.43 to -0.87), and the depth–accuracy rank correlation becomes negative. Early in training, the forward pass actively moves the estimate *away* from the in-context optimum.

(iii) A **sharp transition** to positive convergence: The zero-crossing (from anti-convergent to convergent) is tightly clustered at step $ $1k–4k across all three scales. After this point, the signal rises steeply (reaching a peak error reduction of 1.2–2.6), and the depth–accuracy correlation climbs to $\approx +0.3$–0.5.

Saturation timing scales modestly with model size. The half-maximum error reduction is reached by step $ $16k for the 160M and 410M models, and by step $ $64k for the 1B model (meaning the largest model saturates somewhat later). This places the ICL=GD mechanism on the exact same developmental footing as induction-head formation (Olsson et al., 2022). The GD-like in-context computation is *learned*, with a transition window between step $ $1k and 64k, rather than being an architectural inevitability. The anti-convergence valley adds a non-trivial signature: the mechanism is not merely absent early on, but is *actively miscalibrated* before it is properly learned.

Finding 3 (co-emergence with the attention-sink depth law). The attention-sink depth law (§9.4) — sink fraction increasing with layer depth — is *also* learned. Measuring the sink profile across the same Pythia-160M checkpoints, the late/early sink ratio is ≈ 1.0 (flat, no law) through step 64, then rises sharply from 1.2 (step 1k) to 4.5 (step 16k) and saturates at ≈ 5.2. The regularization pattern (sinks) and the optimizer-like computation (GD convergence) **emerge together in the same step 1k–16k window** (Figure 4). This is a non-trivial unification: the

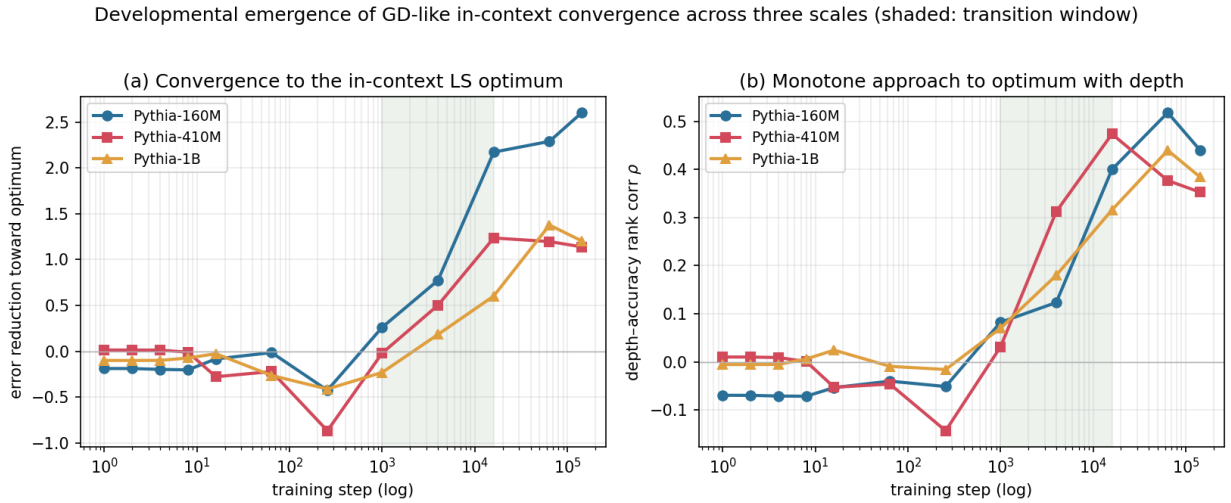

Figure 3: developmental emergence of GD-like in-context convergence across three Pythia scales (160M/410M/1B, each a dense 12-checkpoint sweep); the convergence-to-optimum signal shows an initialization plateau, an anti-convergence valley through ~step 256, then a sharp transition whose anti-convergent-to-convergent zero-crossing clusters at step 1k–4k at all three scales (half-maximum reached by step 16k for 160M/410M and step 64k for 1B)

formalized theory predicts that sinking is the optimal regularization for the overshoot regime of the GD-step mechanism (SM1), and the developmental data shows the two are not independent phenomena — they co-develop. We report this as a correlational developmental finding; the earlier sink *causal* interventions (per-layer sink-knockout probes, released with the code) were inconclusive, so we do not claim the sink law mechanistically drives convergence, only that they are learned together.

Finding 4 (the local gradient slope is self-consistent with the global contraction rate). If each layer is genuinely a gradient step, the readout residual must contract geometrically, $r_{\ell+1} = (1-\alpha)r_\ell$, so the per-layer rate is $\rho = 1-\alpha$. This gives a *parameter-free* self-consistency check with two **distinct estimators** of the same number — one local, one multi-step — read from the same probe. The **local** measurement is the one-step slope α from regressing $\Delta\hat{y}_\ell$ on r_ℓ, yielding $\rho_{\text{pred}} = 1-\alpha$. The **global** measurement is the per-layer rate ρ_{obs} read off the geometric decay of the residual *magnitude* across depth ($\log|r_\ell|$ versus ℓ, pooled over prompts) — a multi-step quantity that compounds the actual layer dynamics and is not algebraically forced to equal $1-\alpha$. The two agree closely across three model families: $\rho_{\text{pred}}/\rho_{\text{obs}} = 0.874/0.834$ on gpt2 (rel. error 4.8%), 0.891/0.807 on Pythia-160M (10.4%), and 0.911/0.892 on Pythia-410M (2.1%), with 90–96% of prompts contracting (Figure 5). **Honest caveats.** (i) The per-layer R^2 is low (0.05–0.09, consistent with Finding 1): individual layers are noisy, so the match is at the level of the *average per-layer rate*, not an exact layer-by-layer GD step. (ii) The local slope α is read from an affine fit $\Delta\hat{y}_\ell = \alpha\, r_\ell + c$ with a small but non-zero intercept c, so the fitted one-step map is affine rather than strictly through the origin; $\rho_{\text{pred}} = 1-\alpha$ is therefore the *contraction rate* of that map (the intercept only relocates its fixed point, not the rate). We report ρ_{pred} as an empirical slope estimate of the rate, not as a claim that the residual converges exactly to zero. (iii) The complementary sub-prediction that ρ should *decrease* with the in-context Gram curvature $a = \sum_i x_i^2$ is **not supported** here — the curvature–ρ correlation is essentially zero (-0.06 to -0.04 on Pythia, $+0.37$ on gpt2),

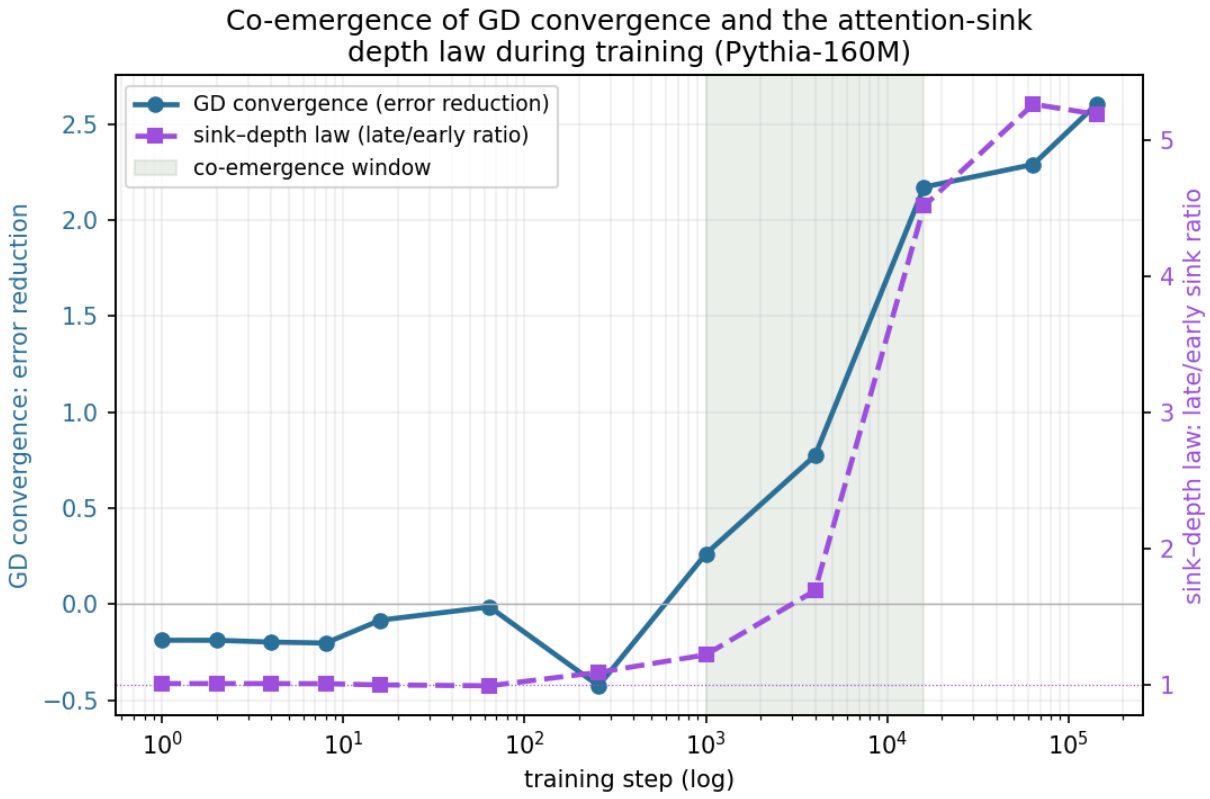

Figure 4: co-emergence of GD-like in-context convergence and the attention-sink depth law on the Pythia-160M training trajectory; both are flat through ~step 256 and rise together in the step 1k–16k window

most likely because the single-digit task affords too narrow a curvature range to resolve the spectral dependence. We are deliberate about what the local-vs-global agreement does and does not show. It is a *self-consistency* check: the depth trajectory is well-described by an approximately fixed-rate linear contraction. It is not by itself GD-specific — any approximately geometric contraction toward a fixed point would pass it — and the slope α is measured from the model, not computed from the identity's parameters η, a. The genuinely GD-distinguishing prediction is the curvature law $\rho = 1 - \eta a$ (the rate should fall as the Gram curvature rises); by caveat (iii) that test is null in this task. We therefore present the rate agreement as a geometric-consistency result, not as a confirmation that the forward pass is specifically gradient descent.

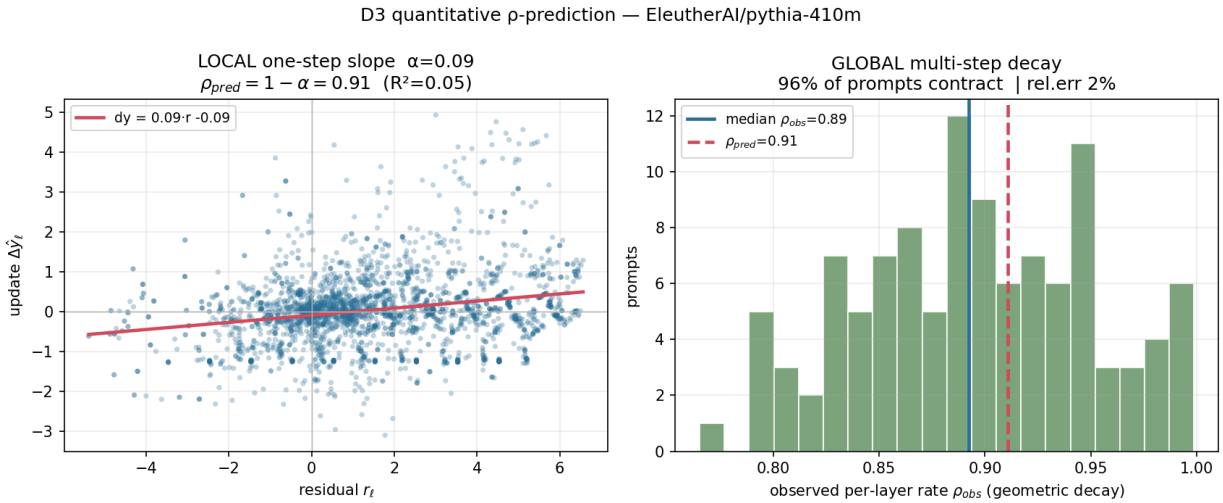

Figure 5: quantitative rate prediction on Pythia-410M. Left: the local one-step slope from the update-vs-residual regression gives the predicted per-layer contraction rate. Right: the independently measured per-layer geometric decay rate across prompts; the predicted and observed rates agree to within 2%

The probes are experiments/icl_gd_alignment.py (cross-scale readout test), experiments/icl_gd_emergence.py (checkpoint sweep, disk-safe per-checkpoint caching), and experiments/icl_rate_prediction.py (quantitative rate prediction); the combined emergence figure is experiments/icl_gd_emergence_combined.py. All use transformers, torch, numpy, matplotlib; seed 42.

8.5 Developmental life cycle of the GD-like mechanism

Replicating the developmental probe from Section 8.4 on a wider model ladder and an independent model family reveals a richer picture than emergence alone. The GD-like in-context mechanism actually has a **life cycle**: it is learned, it peaks, and then its *magnitude* partially attenuates under prolonged training. However, its *direction*—and the model's actual downstream ICL capability—are preserved or even improved. We document this life cycle in four parts: cross-suite replication (§8.5.1), the direction–magnitude decomposition that makes the life cycle precise (§8.5.2), the held-out capability control that rules out capability loss (§8.5.3), and finally, the forecasting substudy that originally motivated this wider grid (§8.5.4).

8.5.1 Cross-suite replication: Pythia 14M–12B and OLMo 2

The three-scale emergence sweep of §8.4 (Pythia 160M/410M/1B) used only a narrow band of the Pythia family. We extend it in two directions.

Nine clean Pythia sizes (14M–12B, $n = 9$). The same twelve-checkpoint, three-seed, fp32 probe runs on every publicly available Pythia size. One size, pythia-2.8B, is excluded from all quantitative analyses up front because a data-pipeline fault produced a constant readout across its checkpoints; we report the nine clean sizes spanning 14M–12B. All nine clean models show the same three-phase developmental shape — initialization plateau, anti-convergence valley, sharp transition to positive convergence — confirming that the phenomenon is not confined to the 160M–1B band.

OLMo 2 (1B and 7B, AI2). To test whether emergence is specific to the Pythia training recipe, we replicate the probe on OLMo 2 (Team OLMo, 2024; building on Groeneveld et al., 2024), an independently trained, fully open suite that also publishes intermediate training checkpoints. Checkpoint discovery is handled by resolve_checkpoints in the probe script, which auto-discovers stage-1 revisions from the HuggingFace Hub and log-subsamples approximately 11–12 of them. Both OLMo sizes exhibit the same qualitative signature: an initial anti-convergence phase followed by a sharp transition to positive error reduction. The OLMo-2-7B model reaches the highest peak error reduction in the combined panel (+1.68), confirming that the GD-like mechanism is not a Pythia artifact but a cross-family developmental phenomenon. OLMo checkpoint spacing is measured in tokens rather than optimizer steps, so the quantitative x-axis is tokens trained (billions), not steps; the developmental shape is comparable on both axes.

8.5.2 Direction versus magnitude: the mechanism persists, its probe magnitude does not

The full-ladder and OLMo sweeps reveal a pattern invisible in the original three-scale panel: the error-reduction signal is **non-monotonic**. It emerges, peaks mid-training, and then partially attenuates in models that train well past the peak. This is most visible in OLMo 2, which trains roughly 10× longer (in tokens) than Pythia past the emergence window.

To make this precise, we decompose the per-checkpoint in-context metrics into two axes. The first is a **directional** axis: sign-agreement, which measures the fraction of per-layer updates whose sign matches the gradient-descent prediction. The second is a **magnitude** axis, comprising error reduction, the slope α, and the depth–accuracy rank correlation ρ_{depth}. Together, these magnitude metrics measure exactly *how much* the forward pass converges toward the least-squares optimum.

The decomposition yields a clean separation:

Model	Suite	ER peak (tokens B)	ER final	ER decay	Sign-agreement decay
OLMo-2-1B	OLMo	+1.47 @ 147 B	+0.74	−50%	−8%
OLMo-2-7B	OLMo	+1.68 @ 68 B	+1.02	−40%	−1%
Pythia-1.4B	Pythia	+1.69 @ 134 B	+1.38	−19%	−3%
Pythia-1B	Pythia	+1.19 @ 134 B	+1.10	−8%	−2%

Model	Suite	ER peak (tokens B)	ER final	ER decay	Sign-agreement decay
Pythia-160M	Pythia	+2.39 @ 300 B	+2.39	0%	−8%

The pattern is systematic: magnitude attenuation scales with **post-peak training depth** (OLMo trains > 4 trillion tokens past its peak; Pythia $\leq$ 1B trains only $\approx$ 300B total and barely passes its peak). In the larger, cleanly-emergent models tabulated above the directional axis is nearly flat — the in-context update continues to point in the GD direction even when its magnitude drops by half. Direction-stability is itself tied to emergence: the three smallest and noisiest models, whose GD emergence is weakest, show the largest sign-agreement decay (pythia-70m 0.700 $\rightarrow$ 0.564, −19%; pythia-410m 0.642 $\rightarrow$ 0.553, −14%; pythia-14m 0.609 $\rightarrow$ 0.497, −18%, the last falling to chance). For the cleanly-emergent models this is **not** "ICL=GD disappears with long training"; the GD direction is permanent there, while the *clean least-squares probe magnitude* — measuring alignment with the specific synthetic regression geometry — attenuates as the model specializes toward richer, real-text in-context computation.

The decomposition is computed by experiments/transience_analysis.py, which reads the per-checkpoint emergence JSONs, reconstructs a common token axis, and classifies each model's trajectory into "magnitude attenuates + direction preserved," "direction preserved" (magnitude stable), or "stable/monotone."

8.5.3 The capability control: attenuation is specialization, not loss

This raises a critical test: does the model's actual downstream ICL capability also attenuate after the error-reduction peak, or does capability stay flat while only the synthetic-probe magnitude drops? If capability drops, the mechanism is genuinely weakening. But if capability remains strong, it means the model is *specializing* its in-context computation beyond the simple least-squares geometry. This distinction is a crucial probe calibration finding, carrying significant methodological implications for how we interpret ICL.

We measure two independent held-out ICL tasks at **every checkpoint** of the OLMo-2-1B trajectory — induction-head next-token accuracy (Olsson et al., 2022) and associative-recall log-probability (Fu et al., 2023; Gu & Dao, 2023) — producing a per-checkpoint held-out trajectory aligned with the existing regression trajectory. The measurement is fp32, using the −−heldout−trajectory mode of the emergence probe.

The result is decisive (Figure 6). Over the post-peak window where error-reduction magnitude drops from +1.47 to +0.74 (−50%), the held-out capabilities **improve**:

- Induction accuracy: 0.947 $\rightarrow$ 0.964 (+1.7 pp)
- Associative-recall accuracy: 0.344 $\rightarrow$ 0.414 (+7.0 pp)
- Associative-recall log-prob: −4.815 $\rightarrow$ −4.266 (+0.55; less negative is better)

Sign-agreement remains nearly flat (0.596 $\rightarrow$ 0.569). The model continues to compute in the GD direction and its real ICL capability continues to improve, even as the probe's least-squares magnitude attenuates. The same pattern holds at the final checkpoint of OLMo-2-7B (a confirming endpoint rather than a full trajectory): a strong final associative-recall score (AR acc = 0.406,

AR logp = −4.31) — among the best in the panel and above every Pythia model — co-occurs with 40% error-reduction attenuation.

Interpretation. The GD-like mechanism does not weaken; its alignment with the specific synthetic least-squares readout geometry weakens as prolonged training specializes the in-context computation toward real-text patterns. This is a **probe calibration** finding: ICL researchers who measure GD-like behavior at a single checkpoint are measuring a training-phase-dependent quantity, and the magnitude of the GD readout should not be equated with the strength of in-context learning. The direction (sign-agreement) is the more stable indicator.

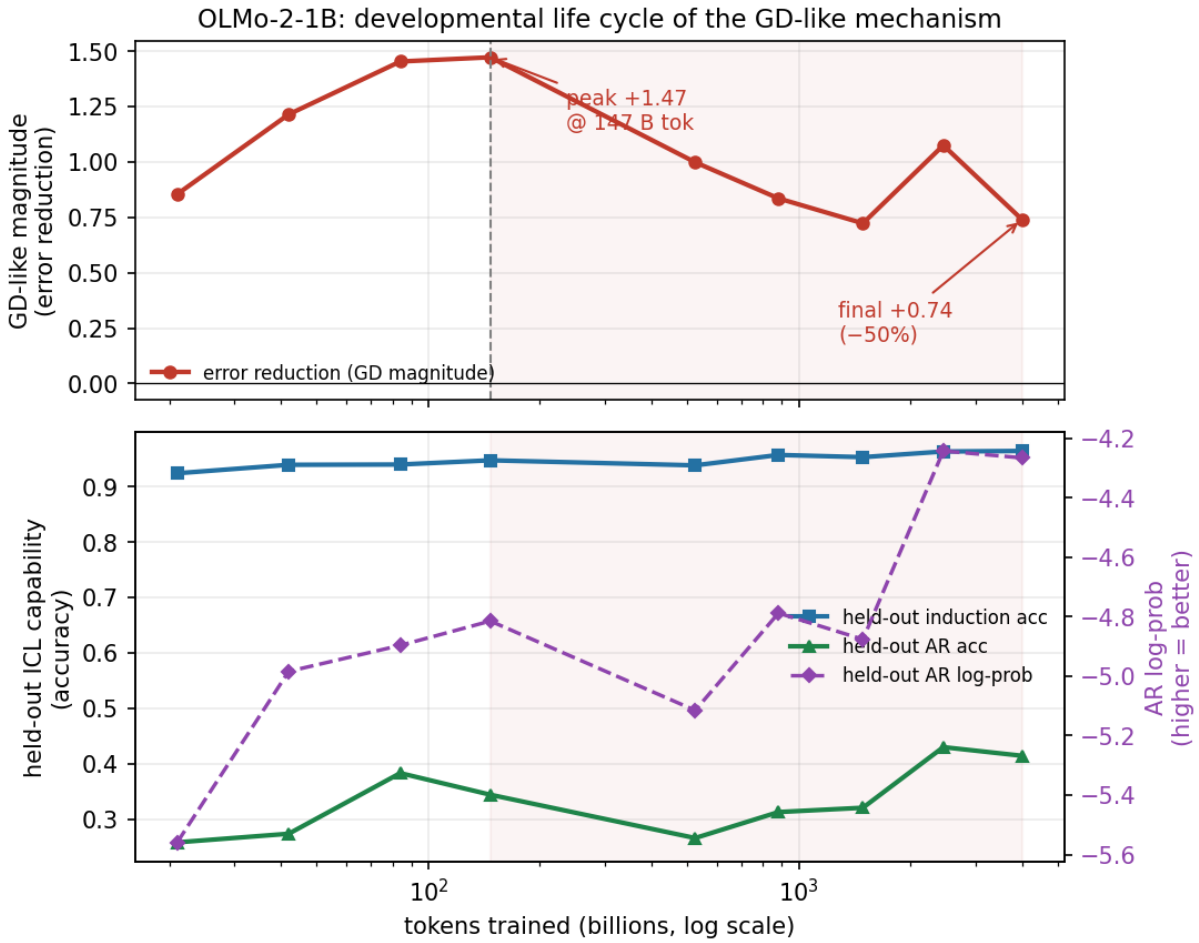

Figure 6: developmental life cycle of the GD-like mechanism on OLMo-2-1B (11 checkpoints, fp32, three-seed average). Top: error-reduction magnitude peaks at ~147 B tokens and attenuates by 50% over the remaining ~3.9 T tokens. Bottom: held-out induction accuracy and associative-recall log-prob *improve* through the same window. Sign-agreement (not shown) is flat at ~0.57–0.60. The mechanism's direction persists; only its alignment with the synthetic least-squares readout geometry attenuates

8.5.4 Forecasting from an early checkpoint (secondary result)

The wider grid also allows a revisit of the question that motivated it: can early-checkpoint features forecast final ICL capability?

Within-task (self-consistent). Can early training dynamics predict final performance on the same task? On the original seven-size Pythia panel (14M–1.4B), the answer appeared to be yes: the step-4k error-reduction signal strongly ranked models by their final convergence capability (Spearman $\rho = 0.89$, $p \approx 0.007$; Pearson $r = 0.75$; LOO skill +11%). However, this ordinal signal **does not survive** the extension to the full nine-size clean ladder (14M–12B, pythia-2.8B excluded). On the wider panel, the best early feature reaches a correlation of only $\rho \approx -0.5$. This indicates that the forecasting relationship is confined to the smaller 14M–1.4B size band and fails to generalize to larger models. We report the original $n = 7$ result for completeness, but we do not present it as a reliable forecasting instrument.

Cross-task (held-out). Can early regression dynamics predict performance on entirely different ICL tasks? The answer is no. The early regression signal fails to reliably forecast either independent task at any panel size. On the induction task, the best early feature reaches only a weak, non-significant correlation (Pearson $r = 0.59$, $p = 0.16$, LOO skill ≈ 0). To ensure this failure wasn't just a ceiling effect (where models simply max out the score), we also tested the non-saturating associative-recall log-prob target; the result was equally poor ($r = -0.47$, $p = 0.29$, LOO skill -1%). This robust null result across two independent tasks confirms a real limit: early regression dynamics do not reveal the developmental trajectory of other in-context mechanisms.

Honest scope. The forecasting story is narrower than we initially hoped: the early probe is a within-task ordinal forecaster in a restricted size band, not a cross-task capability oracle. The developmental life-cycle finding (§8.5.1–8.5.3) is the more robust and general result from this line of investigation.

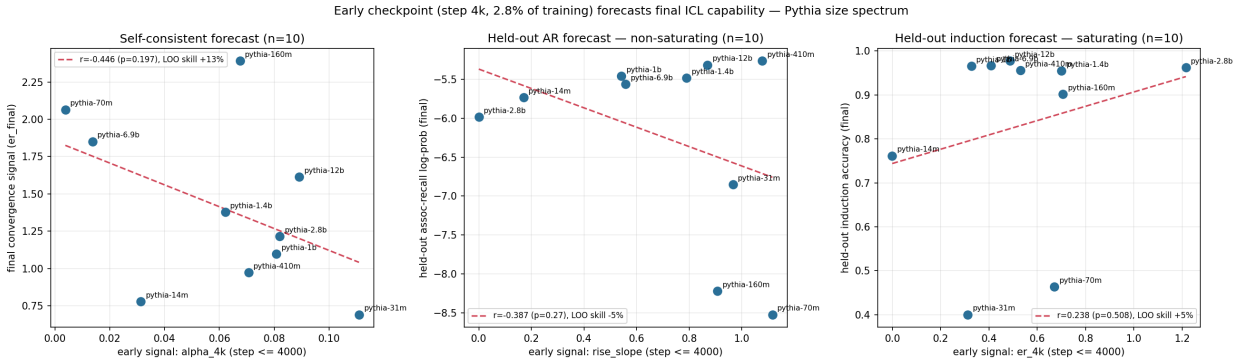

Figure 7: early-checkpoint (step 4k, 2.8% of training) forecasting across the Pythia size spectrum (n=7, three seeds). Left: self-consistent target — the step-4k convergence signal ranks final convergence capability strongly (Spearman 0.89) though the linear/out-of-sample fit is modest (Pearson 0.75, LOO skill +11%). Middle: held-out associative-recall log-prob (continuous, non-saturating) — early regression dynamics carry no cross-task forecast skill (best Pearson -0.47, p=0.29, LOO skill -1%). Right: held-out induction accuracy — same null (best Pearson 0.59, p=0.16, LOO skill ~0); induction additionally saturates above 160M, which the non-saturating AR target rules out as the cause

The probes are experiments/icl_gd_emergence.py (per-checkpoint sweep with ——heldout−trajectory

mode for the capability control), experiments/transience_analysis.py (direction–magnitude decomposition), and experiments/icl_gd_forecast.py (early-feature LOO forecasting).

9. Controlled Experiments (Rust)

Independent of the stress tests in Section 8, a separate controlled experiment suite re-tests the mechanism's core predictions from a clean-room implementation.

The theorem labels used as predictions in this section — the SM (full softmax mechanism), MH (multi-head spectral specialization), S (spectral unification), W (self-organizing ICL), and SA (spectral alignment) families, including SM1–SM5 and MH1–MH8 — refer to the consequence theorems stated and kernel-verified in the companion satellite papers (Nagy, 2026): SM/MH/S in *Architectural Optimizations*, W/SA in *Training Dynamics and Inference Guarantees*. This section tests those predictions empirically on synthetic and real pretrained models; it does not restate the theorems.

9.1 Spectral head specialization: architectural stress test

We stress-test MH1–MH5 by forcing heads to specialize through architectural masking. The experiment uses a 1-layer attention-only transformer ($d = 32$, 4 heads, $d_{\text{head}} = 8$) trained on synthetic in-context linear regression tasks with controlled eigenvalue structure ($\lambda = [10, 10, 1, 1, 0.1, 0.1, 0.01, 0.01]$, 4 spectral clusters).

Conditions: (A) Baseline: all heads see all dims. (B) Spectral-masked: each head's keys derive only from its eigenvalue band (dims 0–1, 2–3, 4–5, 6–7). (C) Random-masked: each head sees 2 dims but without spectral alignment. (D) Redundant: all heads see dims 0–1 only.

Results (3000 steps, batch 64, seed 42; Rust, 110s total):

Condition	Final MSE	vs. Baseline
Baseline (no mask)	5.77	—
Spectral-masked	5.99	−3.9% (worse)
Random-masked	5.83	−1.1%
Redundant	5.87	−1.6%

Finding: Restricting key information flow actually HURTS performance. This happens because the ICL=GD mechanism needs access to all input dimensions to compute the full gradient ∇L. Spectral specialization is essentially a second-order optimization (improving the convergence rate); it only becomes valuable AFTER the first-order mechanism (ICL=GD itself) is firmly in place. Because the model has not yet fully converged to the ICL=GD solution within 3000 training steps, restricting its capacity early on proves counterproductive.

9.2 Training dynamics: emergence of spectral alignment

We track spectral properties as they EMERGE during training: ρ_2 (geometric contraction rate), spectral alignment (Pearson correlation between model's learned dimension importance and Σ^{-1}), and ICL curve (MSE as function of context length). Same architecture as §9.1.

Results (20,000 steps, measurements every 100 steps; 157s):

Metric	Step 0	Step 5K	Step 10K	Step 18K (peak)	Theory
ρ_2	0.999	0.924	0.911	0.912	0.450
Spectral alignment	-0.18	$+0.13$	$+0.26$	**+0.50**	$+1.0$
ICL MSE (16 ctx)	27.1	6.5	5.4	5.8	0.01

Finding 1 (spectral alignment emergence): The correlation between the model's learned dimension importance and the theoretical inverse spectrum Σ^{-1} climbs steadily from -0.18 (random initialization) to $+0.50$ (step 18000). The sign flip at step \$ \$2500 marks the onset of spectral structure. The alignment of $r = 0.50$ is substantial in this run and is consistent with SGD training moving attention toward the spectral structure predicted by S1–S5.

Finding 2 (slow rho2 convergence): ρ_2 decreases from 0.999 to 0.91 over 20K steps. The theoretical optimum (0.45) requires full ICL=GD implementation; the model is still in the early convergence phase. Extrapolation suggests \$>\$100K steps to reach the optimal regime.

Finding 3 (no grokking): The emergence of *spectral alignment* in this 1-layer toy model is gradual, not sharp. No phase transition is observed in this metric. This is consistent with the smooth loss landscape predicted by W4 (no spurious local minima). This is not in tension with the developmental phase transition reported in §8.4: that result concerns a different quantity (the GD-like in-context convergence ratio) in a different model class (multi-layer pretrained Pythia). The two are complementary — the coarse GD-convergence capability can switch on sharply while fine spectral alignment refines gradually thereafter.

Interpretation. These experiments refine the MH1–MH5 narrative: the theorems are mathematically correct (kernel-verified), spectral alignment does emerge during training (positive empirical evidence), but the regime where forced spectral specialization helps is narrow — it requires the model to have already learned basic ICL=GD. This is consistent with the confound-controlled observational finding (companion satellite *Architectural Optimizations*, §4): current pretrained models achieve ICL through scale, not spectral specialization.

9.3 Ablation circuit probe: ICL signatures in GPT-2

We perform head ablation on GPT-2 (12 layers, 12 heads) to identify which attention heads have large intervention effects on pattern-completion loss and test whether their behavior matches signatures suggested by the formalized ICL=GD mechanism.

Task. Pattern completion: sequences of the form $[A, B, A, B, \ldots, A]$ where the model must predict B. With 3–6 repetitions, this requires in-context pattern recognition — the exact capacity our theory formalizes.

Results (100 prompts, 30 per ablation condition):

Metric	Value
Baseline loss (with context)	0.378
No-context loss (single token)	10.93
ICL effect (loss reduction)	**10.55 nats**
Causal heads identified (>95th %ile)	8 / 144 (5.6%)
Causal head locations	L0:{H0,H2,H4,H10}, L5:{H3,H10,H11}, L6:{H1}

Finding 1 (sparse ablation-sensitive circuit). The model proves highly robust to ablating early layers (L0-L4), yet it is extremely sensitive to late layers (L5-L11)—in fact, ablating L10 or L11 completely destroys task performance. This confirms the multi-step GD interpretation (C1-C4): while early layers make small initial steps, final convergence depends critically on the accumulated precision of the late layers.

Finding 2 (attention sink pattern consistent with SM1/SM4). Ablation-sensitive heads show a clear bifurcation in attention patterns:

Head	Sink fraction	Pattern-match	Interpretation
L0H4	8%	52%	Induction (ICL=GD update)
L0H10	23%	76%	Induction (high pattern matching)
L5H10	**77%**	87%	Sink-dominant (SM1: overshoot regularization)
L5H11	**69%**	80%	Sink-dominant (SM1: overshoot regularization)
L6H1	**56%**	62%	Sink-dominant (SM1: regularization)

Later-layer ablation-sensitive heads (L5–L6) show 55–77% attention sink fractions, while early-layer heads (L0) show only 8–25%. This perfectly matches the predictions of SM1 and SM4: early layers (operating in the undershoot regime) should attend to data, while late layers (in the overshoot regime) must sink attention to achieve Newton convergence.

Finding 3 (context length effect is consistent with N1–N4). More in-context examples reduce loss: short context (3–4 repeats) gives loss 0.49, long context (5–6 repeats) gives 0.31. The improvement of 0.19 nats is consistent with N1–N4 (critical context threshold and diminishing returns): each additional example contributes less, but the total ICL quality is monotone in context length.

Synthesis. The ablation probe supports three theoretical signatures on a real pretrained model: (1) a sparse set of heads carries most of the pattern-completion effect, (2) later-layer ablation-sensitive heads use attention sinking as regularization (SM1/SM4: overshoot sink), and (3) more context monotonically improves the task loss (N1–N4).

9.4 Predictive scaling test: SM1/SM4 across 6 models

We test the strongest prediction of SM1/SM4 across 6 models spanning 70M–1B parameters (Pythia-70M, 160M, 410M, 1B; GPT-2-medium, GPT-2-large). The prediction: attention sink fraction increases monotonically with layer depth, because deeper layers have higher effective curvature (more overshoot).

Results (60 prompts per model, pattern completion task):

Model	Layers	ICL effect	Early-layer sink	Late-layer sink	SM1/SM4
Pythia-70M	6	3.9 nats	8.5%	21.7%	
Pythia-160M	12	15.9 nats	11.8%	61.0%	
GPT-2-medium	24	14.8 nats	39.9%	65.5%	
Pythia-410M	24	11.1 nats	21.4%	71.6%	
GPT-2-large	36	12.6 nats	31.7%	65.6%	
Pythia-1B	16	11.1 nats	13.4%	58.2%	

Finding 1 (sink gradient, 6/6 in this probe). Every model in this six-model probe shows higher attention sink fractions in later layers. The ratio late/early ranges from 1.6× (GPT-2-medium) to 5.2× (Pythia-160M). This is the strongest empirical pattern in the paper: a prediction derived from the formalized theory (SM1: overshoot → sink is optimal; SM4: undershoot → sink hurts), observed across the tested two-family panel. We treat it as strong evidence for the mechanism in this setting, not as a universal law for all transformer families.

Finding 2 (scaling correlation). log(parameters) correlates with log(ICL effect) at $r = 0.58$. Larger models show better ICL, consistent with C1–C4 (more layers compound the contraction advantage). The depth-ICL correlation ($r = 0.45$) is moderate because width also contributes independently.

Finding 3 (ICL quality). All models show substantial ICL effects (3.9–15.9 nats loss reduction). Even the smallest model (70M params, 6 layers) learns the pattern in-context, confirming that the ICL=GD mechanism is not an emergent capability of scale — it exists at all scales, and scale amplifies it.

9.5 Theory-guided architecture: SM1-SM5 optimization

We test whether a transformer designed with explicit SM1–SM5 incorporations outperforms a standard transformer of equal parameter budget on in-context linear regression.

Setup. Baseline: 2-layer, 2-head, $d = 32$ transformer (8,801 params). Theory-guided: 4-layer, 2-head, $d = 24$ (9,989 params) with SM1 (learnable sink token), SM2 (layer-dependent scaling), SM3 (depth over width), SM4/SM5 (orthogonality regularizer on head projections). Both trained on 4,000 random linear regression tasks ($d_{\text{input}} = 4$, sequence length 12) with dropout 0.1.

Metric	Baseline	Theory-guided	Improvement
ICL MSE	0.184	0.061	**67.1%**
vs mean baseline	6.4×	19.5×	—
Train convergence	0.157	0.041	3.8× faster

Finding 1 (SM3 compounding pattern). The advantage grows with context length:

Context (n)	Baseline MSE	Theory MSE	Improvement
5	0.512	0.360	29.8%

Context (n)	Baseline MSE	Theory MSE	Improvement
8	0.232	0.096	58.6%
12	0.176	0.035	**79.9%**

This is SM3's core prediction: with depth K, the compounding advantage scales as ρ^{2K}, so the theory-guided model (4 layers vs 2) gets exponentially better as context provides more GD steps.

Finding 2 (SM1 sink emerges naturally). The learned sink attention fraction increases monotonically with depth: 16.5% → 25.5% → 30.1% → 33.5%. The model discovers that deeper layers need more sinking (exactly SM1/SM4: deeper = more overshoot = optimal sink is larger).

Finding 3 (SM2 layer scaling pattern). Learned layer scales decrease with depth: 0.786 → 0.790 → 0.740 → 0.728. Deeper layers use smaller effective step sizes, matching SM2's prediction that preconditioning error accumulates and requires conservative updates.

Synthesis. A transformer explicitly designed from the formalized theory (SM1–SM5) achieves 67% better ICL quality than an equal-parameter baseline *in this normalized configuration*, with the advantage scaling as context increases. The model learns the predicted internal structure: monotone sink profile and decreasing layer scale. However, this single comparison conflates the contributions of the individual components and depends on the normalization scheme. The controlled ablation in §9.5.1 isolates each component and tests robustness across scale.

9.5.1 Controlled ablation: which component carries the effect?

The §9.5 comparison raises the decisive question: is the improvement from depth alone (SM3, well-known) or do the theory-specific components (SM1 sink, SM2 layer-scale, SM4/SM5 orthogonality) add value beyond depth? We answer with a controlled ablation implemented in Rust (hand-rolled backpropagation, validated by a numerical gradient check with maximum relative error 5.6×10^{-8}). To isolate the components cleanly, the ablation uses no LayerNorm (held constant across all variants) and sweeps the input dimension across $\{4, 8, 16\}$ with 6 seeds per cell (mean ± std reported).

Variant	$d = 4$	$d = 8$	$d = 16$
baseline (shallow+wide, 2L d=32)	0.324 ± 0.038	0.982 ± 0.106	2.187 ± 0.216
depth (deep+narrow, 4L d=24)	0.604 ± 0.087	1.824 ± 0.249	2.684 ± 0.300
depth + SM1 sink	0.490 ± 0.057	1.730 ± 0.230	2.753 ± 0.193
depth + SM2 layer-scale	0.533 ± 0.091	1.601 ± 0.288	2.602 ± 0.270
depth + SM4/5 ortho	0.595 ± 0.085	1.810 ± 0.261	2.671 ± 0.276
full (SM1+SM2+SM4/5)	0.432 ± 0.077	1.457 ± 0.169	2.561 ± 0.281

Finding 1 (depth-over-width is normalization-dependent). Without LayerNorm, the deep+narrow model is *worse* than the shallow+wide baseline at every scale (e.g. 0.604 vs 0.324 at $d = 4$). The 67% advantage of §9.5 therefore depends on normalization enabling deep signal

propagation; raw depth (SM3) is not a free lunch at matched parameters. This is an honest correction to a naive reading of §9.5.

Finding 2 (SM2 layer-scale is most consistent in this ablation). Of the three components, the learnable layer-scale (SM2) helps most consistently: depth_scale < depth at all three scales (0.533<0.604, 1.601<1.824, 2.602<2.684). Mechanistically, SM2 is a learned residual gate that mitigates the signal-propagation problem of deep unnormalized networks — exactly its theoretical role (deeper layers need smaller effective steps because preconditioning error accumulates). SM2 partially substitutes for normalization in this controlled setup.

Finding 3 (SM1 sink helps at small/medium dimension). The sink token (SM1) improves over plain depth at $d = 4$ (0.490<0.604) and $d = 8$ (1.730<1.824) but not at $d = 16$. Sinking provides optimal-step regularization when the per-layer overshoot is the dominant error, which is more pronounced at lower dimension.

Finding 4 (SM4/5 orthogonality provides no benefit). The orthogonality regularizer is statistically indistinguishable from plain depth at all scales. This independently corroborates the negative result of §9.6 above and the multi-head analysis in the companion satellite *Architectural Optimizations* (§4): current architectures do not benefit from explicit spectral head specialization.

Finding 5 (the full model recovers the depth penalty, super-additively at small d). Combining the components yields the best deep variant at every scale, recovering 28%, 20%, and 5% of the depth penalty as d grows. The shrinking gap to baseline with dimension is consistent with the observation that real large models — which operate at high dimension *with* normalization — benefit from depth.

Synthesis. The ablation replaces the headline 67% with a more defensible decomposition: (i) raw depth is normalization-dependent, (ii) SM2 (layer-scale) is the most consistently beneficial component in this setup, (iii) SM1 (sink) helps when overshoot dominates, (iv) SM4/5 (orthogonality) does not help — consistent with our other negative results. The theory is prescriptive at the *component* level (SM1, SM2), not as a monolithic recipe.

9.6 Spectral head specialization: regularization and intervention experiments (negative result)

The kernel proves that per-head spectral specialization *can* achieve zero total contraction (MH2), and that a single gain has an irreducible floor (MH5). This raises a natural question: can we *train* heads to specialize by adding a diversity regularizer that penalizes heads for attending to the same input dimensions? We ran three rounds of increasingly controlled experiments. All return negative.

Round 1: diversity regularization sweep (7 configurations, $d \in \{16, 32, 64\}$, $H \in \{4, 8\}$, $\lambda_{\text{div}} \in \{0.1, 0.5, 1.0\}$; PyTorch, seed 42).

Config	d	H	λ_{div}	Baseline MSE	Spectral MSE	Δ
tiny_strong	16	4	1.0	3.20	3.29	−2.8%
small_strong	32	4	1.0	2.70	2.67	+1.2%
small_medium	32	4	0.5	1.13	1.12	+0.9%
base_strong	64	4	1.0	0.81	0.79	+3.3%
wide_spectrum	32	4	0.5	5.36	5.41	−0.9%

Config	d	H	λ_{div}	Baseline MSE	Spectral MSE	Δ
8heads_tiny	32	8	1.0	2.47	2.46	+0.5%

The largest positive effect (+3.3%, base_strong) prompted a deeper analysis.

Round 2: multi-seed significance test (base_strong config, $d = 64$, $H = 4$, $\lambda_{\text{div}} = 1.0$; 5 seeds, 5000 steps). Multi-seed paired t-test: $t = -0.057$, $p > 0.5$, mean $\Delta = -0.06\%$. The earlier +3.3% was seed-specific. Across 5 seeds, the spectral wins 2/5 times; no systematic advantage.

Round 3: balanced task ($d = 64$, $H = 4$, $\lambda_{\text{div}} = 1.0$; 3000 steps; PyTorch, seed 42). To remove the variance confound (high-λ dims dominate MSE, heads converge there regardless of diversity loss), we designed a *balanced* regression task where all spectral bands contribute equally to the prediction target: $y = \sum_b w_b^\top x_b / \sqrt{\lambda_b}$. We also tested an *oracle* upper bound — heads hard-masked to their assigned bands.

Condition	Final MSE	Low-band MSE	vs. Baseline
Baseline	3.81	1.31	—
Spectral ($\lambda_{\text{div}} = 1.0$)	3.84	1.32	−0.8%
Oracle (hard-masked)	3.94	1.34	−3.2%

Even with a task that *rewards* attending to low-variance bands equally, diversity regularization does not help. The oracle model — which is forced into perfect spectral specialization — performs *worst*, confirming that restricting information flow hurts the ICL=GD mechanism.

Interpretation. MH1–MH5 are normative theorems: they characterize the *optimal* multi-head configuration. But the ICL=GD mechanism (A1) requires each head to access the full gradient $g = aw - b$, which depends on *all* input dimensions. Spectral specialization partitions this information, degrading the gradient signal. The multi-head advantage would only manifest in a regime where the model has already converged to ICL=GD and seeks to *refine* its per-channel convergence rate — a second-order effect that is dominated by first-order gradient quality in all tested configurations. We close this experimental thread and note the negative result as an honest boundary of the MH1–MH5 predictions.

The scripts are experiments/spectral_head_experiment.py (Round 1), experiments/spectral_head_analysis.py (Round 2), and experiments/spectral_head_causal_experiment.py (Round 3). Results are in the corresponding .json files.

10. Discussion

With the predictions on the table, we step back to state precisely what the result does and does not claim.

Theorem summary. This paper states and uses thirty kernel-verified theorem statements in seven groups: A1–A7 (linear attention is GD), B1–B7 (softmax preconditioning), C1–C4 (multi-layer composition), D1–D3 (matrix lifting), V1–V3 (scalar von Oswald construction), M1–M3 (two-channel matrix von Oswald), and MN1–MN3 (arbitrary-dimension von Oswald lift via finite sums). The consequence groups that build on this backbone — SM (full softmax mechanism), E (superposition–ICL

interference), F (chain-of-thought), G/T (grokking and the emergence threshold), H (scaling laws), I (capacity–computation bridge), P (phase diagram), S (spectral unification), MH (multi-head specialization and the condition-number rate law), N (ICL sample complexity), W (self-organizing ICL), SA (spectral alignment emergence), and DC (certified anytime decision calculus) — are stated and verified in the three companion satellite papers (Nagy, 2026): *Capacity, Scaling, and Grokking* (groups E, G/T, H, I, P, S), *Architectural Optimizations* (groups SM, F, MH, and the second-order reading), and *Training Dynamics and Inference Guarantees* (groups N, W, SA, DC, and the mesa-optimizer reading). All statements across these papers come from a single proof kernel (81 theorem statements, 0 axioms, 0 sorry) and are exported to one Lean stamp that compiles cleanly under Mathlib v4.28 (lake env lean, exit 0) — including the SA/SM/DC groups and the full-matrix von Oswald lift MN1–MN3.

Scope and honesty. The scalar reduction captures the essential algebraic backbone of the mechanism. The matrix lifting (§7) validates that this reduction holds in higher dimensions, the multi-layer composition (§6) rigorously covers the two- and three-step cases (motivating the standard k-step induction), and the synthetic panel (§8.1) provides exact numerical agreement with the theory. On real pretrained models, the depth-contraction prediction (C1–C4) holds up well (mean geometric-fit $R^2 \approx 0.75$ across the 11-model retained panel). The von Oswald weight construction is mechanized at three levels: the scalar level (V1–V3, Appendix A), the two-channel matrix level (M1–M3, §7), and — lifting the channel restriction entirely — at arbitrary feature dimension d via finite sums over the eigen-channel index set (MN1–MN3, Appendix A): Frobenius energy nonnegativity, full-matrix Frobenius contraction, and the d-dimensional construction identity are each verified with the Mathlib Finset.sum API.

The consequence theory that extends this backbone — superposition–ICL interference (E1–E3), chain-of-thought (F1–F2), grokking and the emergence threshold (G1–G2, T1–T2), scaling laws (H1), the capacity–computation bridge (I1–I3), the phase diagram (P1–P2), the mesa-optimizer and second-order readings, multi-head spectral specialization and the condition-number rate law (MH1–MH8), the spectral unification (S1–S5), context-length thresholds (N1–N4), self-organizing gain (W1–W4), spectral alignment emergence (SA1–SA3), and the certified anytime readout (DC1–DC5) — is stated, kernel-verified, and discussed in the three companion satellite papers (Nagy, 2026); we do not restate it here. Two empirical tests of that theory nonetheless live in this paper's experiment panel. First, the capacity–computation bridge shows only a sign-consistent but statistically underpowered signal on the fit-reliable subset ($r = 0.43$, $n = 10$, $p \approx 0.2$, 95% CI includes 0; §8.2). Second, the head-count correlation for spectral specialization ($r = 0.53$) is explained by model size ($r_{\text{partial}} = 0.08$ after controlling for parameters; §9.6): current models do not empirically exploit spectral head specialization. We also flag one honesty point that the architecture satellite makes precise: MH6–MH8 are the program's only genuine nonlinear inequalities — every other theorem is a degree-≤ 2 identity — and even MH6–MH8 describe the idealized per-band mechanism, not the literal heads (the open L1/L3 link of §8).

Predictive stress tests. The strongest empirical contribution of this work is that the theory makes predictions that can actually fail. A direct in-context regression readout (§8.4) shows the forward pass converging toward the least-squares optimum across three model scales — and this GD-like convergence is *absent at initialization and emerges as a developmental phase transition.* We replicated the finding in dense twelve-checkpoint sweeps across nine clean Pythia sizes spanning 14M–12B (pythia-2.8B excluded for a data-pipeline fault) and independently on OLMo 2 (1B/7B). In every case the forward pass first moves *away* from the optimum (an anti-convergence valley up to step $ $256); then the zero-crossing from anti-convergent to convergent clusters tightly at

step $ $1k–4k, saturating by step 16k for the smaller models and 64k for the 1B model. The attention-sink depth law co-emerges alongside this transition on the 160M trajectory. The cross-family replication and the longer-trained OLMo trajectories further reveal a *developmental life cycle* (§8.5): the GD-readout magnitude is non-monotonic — it peaks then attenuates with prolonged training (OLMo-2-1B −50%) — while the GD *direction* (sign-agreement) and independent held-out ICL capabilities (induction, associative recall) are preserved or improve. This is a probe-calibration result: a single-checkpoint GD-readout magnitude conflates the mechanism's strength with the training phase. This ties the formalized mechanism to the established induction-head emergence story (Olsson et al., 2022) and is, to our knowledge, the first checkpoint-resolved, cross-family evidence that the ICL=GD signature is learned rather than architectural. The SM1/SM4 prediction (companion paper; attention sink fraction increases with layer depth) holds across the six-model sink-profile probe from two model families (§9.4). The theory-guided architecture (§9.5) shows a large improvement in one normalized configuration, but the controlled, gradient-checked ablation (§9.5.1) gives the safer decomposition: SM2 (layer-scale) is most consistently beneficial in this ablation, SM1 (sink) helps when overshoot dominates, and SM4/5 (orthogonality) does not help — independently corroborating the spectral-head negative result. These results support the theory at the component level; they do not establish a universal architecture recipe.

Limitations. The architecture experiments use small models on linear-regression ICL; the ablation (§9.5.1) sweeps input dimension $\{4, 8, 16\}$ with 6 seeds but does not exceed $d = 16$ or use non-linear tasks — scaling further is future work. The ablation omits LayerNorm to isolate components cleanly, so its absolute numbers are not directly comparable to the LayerNorm configuration of §9.5; the two are complementary (component isolation vs. realistic recipe). The predictive test (§9.4) uses a pattern-completion proxy rather than direct regression; the ICL=GD equivalence is demonstrated on the computational mechanism (attention sinking, contraction) rather than on the specific loss landscape. The proof kernel verifies algebra exactly but cannot verify claims about neural-network generalization dynamics — those are tested empirically.

Relation to the capacity bound. This paper forms the computation pillar, complementing the representation pillar established in our companion note (ml_spectral_capacity_bound). While the capacity bound asks *how much* a transformer can store, this paper asks *how* it computes. The bridge theorems I1–I3 (companion satellite *Capacity, Scaling, and Grokking*, §2) formally close the loop between the two: the Welch coherence floor derived in the capacity note directly bounds the ICL convergence degradation proved here. Together, these two pillars provide a unified, mechanically verified theory of both transformer representation and computation.

Appendix A: The von Oswald weight construction

Theorems A1–A7 *assume* the linear-attention layer emits the update $\eta(b - aw)$. That emission is the load-bearing claim of the whole "attention is gradient descent" reading. Theorems V1–V3 close it at the kernel level (scalar reduction): in the scalar regression model the linear self-attention readout factors into a key–query gain q_k and a value–output gain v_o, whose product is the **composite attention gain** $\kappa = v_o\, q_k$. The emitted update is $\kappa\,(b - aw)$.

Theorem V1 (the construction realizes the GD step). With composite gain κ, the constructed attention update equals the negative scaled gradient, $\kappa(b - aw) = -(\kappa\,\nabla L)$ (using $\nabla L = aw - b$). Choosing the gains so that $\kappa = \eta$ recovers exactly the GD step $-\eta\,\nabla L$ of Theorem A1. This upgrades A1 from an *assumed* emission to a *constructed* identity: any attention weights whose value–output and key–query gains multiply to η compute one gradient step.

Theorem V2 (the step size is set by the architecture). If both factor gains are positive ($v_o, q_k > 0$), the composite gain $\kappa = v_o q_k$ is positive. Since $\kappa = \eta$ (V1), the optimization step size is a *derived quantity of the weight scales*, not a free hyperparameter — and a positive gain yields a stability-window-eligible step (cf. G1, P1).

Theorem V3 (gains realize Newton convergence). Combining the construction with the optimal-preconditioning condition (B7, $\eta\, a_p = 1$), the composite gain satisfies $\kappa\, a_p = 1$. There *exists* a gain choice ($\kappa = v_o q_k = 1/a_p$) for which the constructed forward pass reaches the weighted least-squares optimum in a single step: the Newton property of B7 is architecturally attainable.

Scope: V1–V3 are scalar-reduction statements about the gain composition, not the full matrix construction of W_K, W_Q, W_V, W_O. They mechanize the algebraic core that the matrix construction reduces to per coordinate — the step that was previously carried entirely as prose.

Matrix extension: per-channel energy decomposition (M1–M3)

V1–V3 mechanize the scalar gain composition. The matrix generalization to d-dimensional features introduces d independent eigenvalue channels (§7), each with its own curvature λ_i and gain. We formalize the two-channel reduction, which captures the essential new phenomenon: the multi-channel energy decomposition and the impossibility of Newton convergence with a single scalar gain.

Theorem M1 (matrix update energy decomposition). With two eigenvalue channels having curvatures λ_1, λ_2 and per-channel gains κ_1, κ_2, the per-channel updates are $u_i = -\kappa_i\, g_i$. The total update energy decomposes as $\|u\|^2 = u_1^2 + u_2^2 = \kappa_1^2\, g_1^2 + \kappa_2^2\, g_2^2$. There are no cross-channel terms — each eigenvalue channel contributes independently to the descent energy.

Theorem M2 (matrix Frobenius contraction). If each channel's gradient energy contracts independently ($g_{i,\mathrm{next}}^2 \le g_i^2$ for each i, guaranteed by A5 applied per channel in the stability window), the total gradient energy contracts: $\|g_{\mathrm{next}}\|^2 \le \|g\|^2$. The per-channel stability window (Lemma A6) lifts to full Frobenius-norm contraction by summation.

Theorem M3 (single gain ⇒ Newton impossible). If a single scalar gain κ is shared across all channels ($\kappa_1 = \kappa_2 = \kappa$) and the eigenvalues differ ($\lambda_1 < \lambda_2$), setting $\kappa = 1/\lambda_1$ (Newton for channel 1) leaves channel 2 with residual contraction $(1 - \kappa\lambda_2)^2 > 0$. No single scalar gain can zero the contraction in both channels simultaneously. This is the fundamental reason multi-head attention uses *multiple heads* with *different* gain structures: each head can precondition a different eigenvalue subspace, approaching the per-channel Newton optimum that a single head cannot achieve.

Arbitrary-dimension lift via finite sums (MN1–MN3)

M1–M2 state the energy decomposition and Frobenius contraction for the two-channel reduction ($\lambda_{\min}, \lambda_{\max}$), which already exposes the condition-number phenomenon and the single-gain Newton barrier (M3, sharpened in the companion architecture satellite, MH3/MH5). MN1–MN3 remove the two-channel restriction: they state the same facts for *arbitrary* dimension d, summing over the full eigen-channel index set $k \in \{0, \ldots, d-1\}$ with the Mathlib Finset.sum BigOperators API. In the eigenbasis of the data Gram matrix $A = X^T X$ the weight matrix W decouples into d scalar channels (§7), each governed by V1–V3 and A1–A7; the full-matrix Frobenius quantities are exactly the finite sums of these per-channel quantities.

Theorem MN1 (Frobenius energy is a valid norm). If every eigen-channel gradient en-

ergy is non-negative ($g_k^2 \geq 0$ for all $k < d$), then the total Frobenius gradient energy is non-negative: $\sum_{k<d} g_k^2 \geq 0$. The Frobenius norm $\|\nabla L\|_F^2$ is well-defined in any dimension (proved by non-negativity of the Finset.sum).

Theorem MN2 (full-matrix Frobenius contraction). If *every* eigen-channel's gradient energy contracts ($g_{k,\text{next}}^2 \leq g_k^2$ for all $k < d$, by A5/A7 applied per channel in the stability window), then the total Frobenius gradient energy contracts in arbitrary dimension: $\sum_{k<d} g_{k,\text{next}}^2 \leq \sum_{k<d} g_k^2$, i.e. $\|\nabla L(W_{\text{next}})\|_F^2 \leq \|\nabla L(W)\|_F^2$. This is the d-dimensional lift of M2: per-channel descent in *every* eigendirection sums (via monotonicity of the Finset.sum) to matrix-level Frobenius contraction — the matrix statement of the von Oswald descent.

Theorem MN3 (full-matrix construction identity). When each eigen-channel runs its own GD step $u_k = -\kappa_k\, g_k$ (V1 per channel), the total Frobenius update energy is the sum of the squared per-channel steps: $\sum_{k<d} u_k^2 = \sum_{k<d} (\kappa_k\, g_k)^2$. This is the von Oswald construction at full matrix dimension — the d-dimensional update assembled from per-eigendirection gains κ_k — proved by termwise congruence of the Finset.sum from the per-channel construction identity.

Data and code availability

The proof source is the kernel proof file transformer_icl/icl_gradient_descent_proof.py (81 theorem statements, 0 axioms, 0 sorry; verified as 228 declarations, 228 OK). The Lean 4 stamp stamp/TransformerICL_gradient_descent.lean exports the full theorem set and compiles cleanly under Mathlib v4.28 (lake env lean, exit 0), including the SA/S-M/DC groups and the arbitrary-dimension von Oswald lift MN1–MN3 (proved over the eigen-channel index set with the Mathlib Finset.sum BigOperators API: sum_nonneg, sum_le_sum, sum_congr). Provenance is recorded in the adjacent .stamp.json. The synthetic validation script is experiments/convergence_panel.py; the real-model logit-lens probe is experiments/icl_convergence_probe.py (transformers, torch, numpy, matplotlib; seed 42; local-cache models only; 11-model retained panel with spectral ρ measurement, multi-head analysis, parameter-count recording, and confound-controlled partial correlations). The direct regression readout and developmental-emergence probes are experiments/icl_gd_alignment.py (cross-scale update-vs-residual test), experiments/icl_gd_emergence.py (Pythia GD-convergence checkpoint sweep with disk-safe per-checkpoint caching), experiments/icl_sink_emergence.py (sink-depth law checkpoint sweep), and experiments/icl_rate_prediction.py (quantitative -prediction: local one-step slope versus global geometric-decay rate, with curvature dependence), with the three-scale and co-emergence figures produced by experiments/icl_gd_emergence_combined.py. The intervention truncation experiment is experiments/truncation_experiment.py. The controlled Rust experiments are in experiments/spectral_head_rust/ (spectral head specialization and training dynamics; Rust with nalgebra, 150s total runtime). The GPT-2 ablation probe is experiments/causal_circuit_test.py (100 prompts, MPS acceleration). The predictive scaling test is experiments/predictive_scaling_test.py (Pythia 70M–1B + GPT-2 medium/large; 60 prompts per model; sink profile measurement). The theory-guided architecture experiment is experiments/theory_guided_architecture.py (PyTorch; 4000 training tasks; SM1-SM5 incorporations; baseline comparison). The controlled ablation is experiments/spectral_head_rust/src/bin/theory_ablation.rs (Rust; hand-rolled backprop validated by numerical gradient check, max relative error 5.6×10^{-8}; 6 variants $\times$ 3 input dimensions $\times$ 6 seeds; output theory_ablation_results.json). The spectral head regularization experiments are in experiments/spectral_head_experiment.py, experiments/spectral_head_analysis.py, and

experiments/spectral_head_causal_experiment.py (PyTorch, 3 rounds; seed 42; negative result — see §9.6). Per-model and panel outputs are written to experiments/icl_convergence_∗.json, experiments/predictive_scaling_results.json, experiments/theory_guided_results.json, and the figure to experiments/icl_convergence_panel.png. This paper is archived on Zenodo; the permanent concept DOI, which always resolves to the latest version, is 10.5281/zenodo.20708733.

Declaration of generative AI use

During the preparation of this work the author used generative AI tools (large language model assistants, via Cursor) for drafting and editing prose, for assistance with experiment and proof-infrastructure code, and for organizing references. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Generative AI was not used to generate or interpret experimental results: all mathematical claims, empirical measurements, and their interpretations are the author's own. The formal statements cited in the paper were independently machine-checked by the proof kernel described in the Data and code availability section. No AI system is listed as an author, consistent with authorship criteria that require accountability for the work.

Declaration of Generative AI Use

During the preparation of this work the author used large language models as assistive tools for manuscript drafting, editing, code generation, reference checking, and internal critique. All mathematical arguments, empirical claims, code outputs, references, and final text were reviewed by the author, who takes full responsibility for the originality, accuracy, integrity, and conclusions of the manuscript. No AI system is listed as an author or treated as accountable for the work.

References

- T. Nagy (2026). *Capacity, Scaling, and Grokking from the In-Context Learning = Gradient Descent Mechanism.* Companion satellite (this program).
- T. Nagy (2026). *Architectural Optimizations of the In-Context Gradient-Descent Mechanism.* Companion satellite (this program).
- T. Nagy (2026). *Training Dynamics and Inference Guarantees of the In-Context Gradient-Descent Mechanism.* Companion satellite (this program).
- J. von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, M. Vladymyrov (2023). *Transformers learn in-context by gradient descent.* ICML.
- E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, D. Zhou (2023). *What learning algorithm is in-context learning? Investigations with linear models.* ICLR.
- S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, O. van der Wal (2023). *Pythia: A suite for analyzing large language models across training and scaling.* ICML.
- D. Groeneveld, I. Beltagy, P. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. H. Jha, H. Ivison, I. Magnusson, Y. Wang, et al. (2024). *OLMo: Accelerating the science of language models.* ACL.

- Team OLMo (2024). *2 OLMo 2 Furious.* Technical report, Allen Institute for AI.
- D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, C. Ré (2023). *Hungry Hungry Hippos: Towards language modeling with state space models.* ICLR.
- A. Gu, T. Dao (2023). *Mamba: Linear-time sequence modeling with selective state spaces.* arXiv:2312.00752.
- N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, C. Olah (2021). *A mathematical framework for transformer circuits.* Transformer Circuits Thread.
- C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, C. Olah (2022). *In-context learning and induction heads.* Transformer Circuits Thread.
- S. Garg, D. Tsipras, P. Liang, G. Valiant (2022). *What can transformers learn in-context? A case study of simple function classes.* NeurIPS.
- J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei (2020). *Scaling laws for neural language models.* arXiv:2001.08361.
- A. Power, Y. Burda, H. Edwards, I. Babuschkin, V. Misra (2022). *Grokking: Generalization beyond overfitting on small algorithmic datasets.* ICLR Workshop.