COMPUTATION-ALIGNED PARAMETER SHARING ACCELERATES GROKKING BEYOND WEIGHT DECAY

Anonymous authors
Paper under double-blind review

ABSTRACT

In grokking, a model memorizes its training data long before it generalizes, and closing the gap can take an order of magnitude more optimization steps. Accelerating grokking would therefore make generalization more compute-efficient. Weight decay is known to drive the transition, but it supplies only a global preference for smaller parameter norms and does not determine which training examples update the same parameters. We introduce COMROUTE, a training incentive for computation-aligned parameter sharing on top of weight decay. Given labels for which examples share a reusable rule, COMROUTE encourages those examples to route through overlapping experts in a sparse MoE model, so that their learning signals accumulate in shared parameters instead of redundant copies. Theocratically, in a stylized fixed-route model, we characterize generalization time in the small-initialization limit. Computation-aligned routing generalizes earlier than size-matched random routing whenever it strictly improves the threshold's critical learning rate; we also give two sufficient conditions under which a single ungrouped block is slower. Empirically, we show that on a factorized modular-arithmetic task with held-out factor combinations, a 2.3M-parameter MoE trained with COMROUTE reaches 90% held-out accuracy after around 23K updates, 1.4$\times$ fewer than with size-matched random routing and 6$\times$ fewer than with weight decay alone. At 107M parameters and in OLMoE-1B-7B, COMROUTE shows a similar overall trend. In short, weight decay helps a reusable solution win eventually, and COMROUTE helps it win sooner.

1 INTRODUCTION

A central challenge on the path toward more general AI systems is to turn massive amounts of training data and computation into capabilities that transfer beyond the examples seen during training (Zhou et al., 2025; Burnell et al., 2026). However, achieving such generalization can require enormous amounts of computation (Cottier et al., 2025). When generalization emerges only long after memorization, as in grokking (Power et al., 2022), training must continue for many additional updates before the model acquires a solution that transfers to new problems. Reducing this delay could therefore lower the optimization required to reach a target level of generalization or yield stronger generalization under a fixed training budget.

Mechanistic studies suggest that grokking can involve a transition from example-specific memorization toward a reusable solution that generalizes across inputs (Nanda et al., 2023). Varma et al. (2024) further argue that weight decay can favor such generalizing solutions when they are more parameter-efficient than memorizing alternatives, suggesting that reducing redundant structure and implicitly learning computations that can be reused on new examples can help drive the transition. Work on modular and compositional learning takes a more explicit approach by representing reusable parts of a solution with distinct neural modules that can be composed and reused across tasks, improving compositional generalization and reducing the data needed for new tasks (Alet et al., 2019; Schug et al., 2024). Motivated by these perspectives, we introduce COMROUTE on top of weight decay, a training incentive for computation-aligned parameter sharing. Given gold labels identifying which examples share a reusable rule, COMROUTE adopts the same principle of computational reuse without requiring a predefined module for each reusable rule. Instead, it encourages examples sharing a computation to learn through overlapping parameter pathways, so that learning signal for the shared computation accumulates in common parameters rather than redundant copies. We show theoretically

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

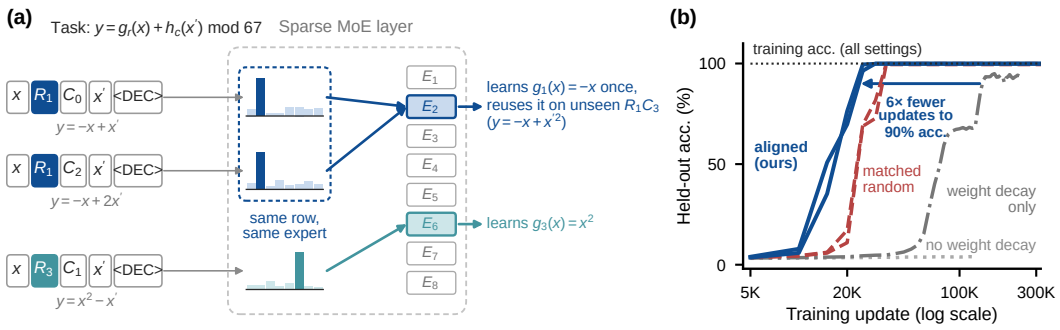

Figure 1: **Grouping examples by the computation they share makes generalization arrive sooner.** **(a)** ComRoute encourages examples sharing a rule (e.g., R_1) to route to the same expert, which learns the rule once and reuses it on unseen combinations. **(b)** Held-out accuracy of a 2.3M-parameter MoE (mean over runs; row and column grouping shown together). "Aligned" denotes computation-aligned grouping with ComRoute, and "matched random" its control with groups of the same number and sizes that share no rule. All settings memorize by 5K updates, but computation-aligned grouping reaches 90% held-out accuracy at about 23K updates, versus 33K for matched-random grouping and 139K for weight decay alone.

and in controlled MoE experiments that this explicit computation-aligned parameter-sharing incentive can accelerate grokking.

To understand why computation-aligned parameter sharing can accelerate grokking, §2 analyzes a stylized fixed-route model in which a network can either memorize individual training examples through private coordinates or learn reusable rules through shared parameter blocks. Weight decay is essential to the mechanism: it keeps memorization from absorbing the training signal completely, so a fixed fraction of the residual remains available to drive the reusable rules. How fast a rule is then learned depends on how that residual is routed. When all examples of a rule train the same block, their gradients add and the rule's growth rate is set by its full training mass; when they are spread across unrelated blocks, each copy sees only a fraction of that mass, and the rule must effectively be learned several times at a lower rate. Under stated sector and threshold assumptions, we prove that generalization time scales as $\log(\alpha^{-2})$ divided by twice the growth rate of the slowest required rule, so that computation-aligned routing generalizes earlier than any size-matched partition of the same examples, and earlier than a single ungrouped block under a gain or depth condition. A controlled Fourier simulation verifies the predicted ordering.

Empirically, as described in §3, we adopt mixture-of-experts models, where routing directly controls which experts process each input (Shazeer et al., 2017; Fedus et al., 2022). We evaluate ComRoute on a factorized modular-arithmetic task with reusable row and column computations, holding out selected combinations and comparing computation-aligned routing with size-matched random grouping, weight decay alone, and no weight decay as shown in Figure 1. Across MoEs from 2.3M parameters to the pretrained OLMoE-1B-7B, computation-aligned routing is the first to reach every held-out accuracy threshold; in the smallest model it generalizes with roughly $1.4\times$ fewer updates than matched-random routing and $6\times$ fewer than weight decay alone (Figure 1b). Expert analyses show that shared rules produce more coherent gradients and route to factor-specific experts during generalization.

2 Computation-Aligned Parameter Sharing Analysis

Grokking is delayed generalization where a model first memorizes its training examples and does not generalize until it later learns computations that can be reused on new examples. We study whether this delay can be shortened through *computation-aligned parameter sharing*, where examples that require the same reusable rule update the same parameters, so that their learning signals accumulate in a common computation path instead of being divided among separate copies. The theory assumes these gold rule labels are known and asks how parameter sharing should be organized around them.

We analyze a stylized fixed-route model with disjoint trainable blocks and common positive weight decay. Each comparison defines an *arm*; in the fixed-route setting, arm π is specified by a route R_π that assigns examples to blocks. Aligned routing sends examples with the same supplied rule to the same block, whereas size-matched unaligned routing splits them across blocks. We also study single-block comparator arms in which all rules share one pathway. We ask how these parameter-sharing patterns change the time from memorization to generalization. Theorem 1 characterizes generalization time for any admissible route, Proposition 2 shows that computation-aligned routing generalizes earlier than size-matched partitions when it strictly raises the critical rate, and Theorem 3 gives two sufficient conditions under which a single ungrouped block is slower. All results are conditional and asymptotic as $\alpha \to 0$; full details are in Appendix A.

2.1 Model, training, and generalization time

We first define the route-circuit model and the assumptions used in the analysis. Full technical details are given in Appendix A.1.

Definition 1 (Route-circuit model)**.** There are N training examples $\xi_i = (c_i, \nu_i, x_i)$, where $c_i \in \{1, \dots, G\}$ is the rule label, $\nu_i \in \{1, \dots, K\}$ is a fixed partition label, and $x_i \in \mathbb{R}^d$ contains the operand features. The target is $y_i = W_{c_i}^\top x_i$, where $W_c \in \mathbb{R}^d$ represents rule c.

An *arm* π denotes one model configuration, specified in the fixed-route comparisons by a route $R_\pi(c, \nu) \in \mathbb{R}^K$. Each of K disjoint parameter blocks contains the same J fixed features $F_q(c, x)$. Feature q in block k has effective trainable coefficient $z_{kq} = u_{kq} v_{kq}$. The routed feature is $D_{\pi,kq}(c, \nu, x) = R_{\pi,k}(c, \nu) F_q(c, x)$. For a hard route, exactly one entry of R_π is nonzero. The aligned arm A uses $R_{A,k}(c, \nu) = \mathbf{1}\{k = c\}, K \geq G$, so examples with the same rule update the same parameter block. Each training example also has a private scalar h_i, which vanishes on held-out examples. With memory mobility $\kappa > 0$, the prediction is

$$\widehat{y}_i = \sum_{k=1}^{K} \sum_{q=1}^{J} u_{kq} v_{kq} D_{\pi,kq}(\xi_i) + \sqrt{\kappa N}\, h_i. \tag{1}$$

The private coordinates can memorize individual training examples but cannot improve held-out prediction. They are an idealization of the data-spanning/null-space or signal/noise decompositions used in analyses of grokking and benign overfitting (Xu et al., 2026a; Cao et al., 2022), imposed here so that the sector reduction below is exact. In the MoE experiment, a block corresponds naturally to an expert receiving the token; the theoretical route is fixed, whereas the experimental objective encourages learned routes to align within a supplied computational group.

Assumption 1 (Training and initialization)**.** *All compared arms use squared-loss gradient flow with the same positive weight decay λ on every trainable parameter:*

$$\mathcal{J} = \frac{1}{2N} \sum_{i=1}^{N} (\widehat{y}_i - y_i)^2 + \frac{\lambda}{2} \left[\sum_{k=1}^{K} \sum_{q=1}^{J} (u_{kq}^2 + v_{kq}^2) + \sum_{i=1}^{N} h_i^2 \right]. \tag{2}$$

Writing $a = (k, q)$, the two factors start small and balanced, $u_a(0) = v_a(0) = \alpha\sqrt{c_a}, \qquad c_a > 0$, and private memory starts at $h_i(0) = 0$. Held-out examples are drawn from a fixed law P_* on which every private coordinate h_i is zero, so held-out prediction uses only the routed circuit. The training data, P_*, routes, features, dimensions, c_a, κ, and λ remain fixed as $\alpha \downarrow 0$; only the initialization scale α varies.

Let $e_i = y_i - \widehat{y}_i$, $\langle a, b \rangle_N = \frac{1}{N} \sum_i a(\xi_i) b(\xi_i)$. Gradient flow gives

$$\dot{u}_a = v_a \langle D_{\pi,a}, e \rangle_N - \lambda u_a, \qquad \dot{v}_a = u_a \langle D_{\pi,a}, e \rangle_N - \lambda v_a, \qquad \dot{h}_i = \sqrt{\kappa/N}\, e_i - \lambda h_i. \tag{3}$$

For a hard route, $\langle D_{\pi,kq}, e \rangle_N$ sums residual signal only from examples assigned to block k. Thus routing determines which examples update the same reusable coefficients.

Definition 2 (Generalization time)**.** Let P_* be the fixed held-out distribution and define $\langle a, b \rangle_* = \mathbb{E}_{P_*}[ab], \qquad Y_* = \|y\|_* > 0$. For relative held-out MSE threshold $\delta > 0$,

$$\mathcal{E}_\pi(t) = \frac{\|y - \widehat{y}_\pi(t)\|_*^2}{Y_*^2}, \qquad T_\pi(\delta) = \inf\{t \geq 0 : \mathcal{E}_\pi(t) \leq \delta\}. \tag{4}$$

Training first passage at MSE level $\eta > 0$ is $T_\eta^{\mathrm{tr}} = \inf\{t \geq 0 : \|y - \widehat{y}(t)\|_N^2 \leq \eta\}$.

$T_\pi(\delta)$ is the first time route π reaches held-out error δ, while T_η^{tr} is the first time the training error reaches η.

2.2 Reusable sectors and their growth

Definition 3 (Reusable sector)**.** A *sector* is one independent reusable learning direction. For arm π, routed columns are partitioned into sets $I_{\pi,j}$. All columns in sector j are scalar multiples of a common unit-norm training direction $\Phi_{\pi,j}$: $D_{\pi,a} = g_{\pi,ja}\Phi_{\pi,j}, a \in I_{\pi,j}$, where $g_{\pi,ja}$ is the signed gain of path a. The target coefficient in this direction is $\theta_{\pi,j} = \langle y, \Phi_{\pi,j}\rangle_N$. Orient each nonzero coefficient so that $\theta_{\pi,j} > 0$.

Assumption 2 (Independent and compatible sectors)**.** *Different sector directions are orthogonal on the training distribution. For each sector, let $\rho_{\pi,j} \geq 0$ be its training-to-held-out scale. On held-out inputs, each sector with $\rho_{\pi,j} > 0$ has a corresponding unit direction $\Phi^*_{\pi,j}$ satisfying $D_{\pi,a} = \rho_{\pi,j} g_{\pi,ja}\Phi^*_{\pi,j}$. These held-out directions are orthogonal, and their target coefficients satisfy $\langle y, \Phi^*_{\pi,j}\rangle_* = \rho_{\pi,j}\theta_{\pi,j}$. If $\rho_{\pi,j} = 0$, the held-out columns vanish and no $\Phi^*_{\pi,j}$ is needed. The remaining held-out target component is denoted $y^*_{\perp,\pi}$. We require the full admissibility conditions of Definition 5; in particular, at least one held-out target sector is present and every such sector has a positive-gain path.*

The sector assumption decomposes the target into independent reusable components with consistent training and held-out representations, so each component can be analyzed separately.

Residual signal and sector growth. For one arm, omit π. Define the per-example memory prediction $m_i = \sqrt{\kappa N}\, h_i$ and let $m = (m_1, \ldots, m_N)$. In sector j, define $U_j = \sum_{a\in I_j} g_{ja} z_a, m_j = \langle m, \Phi_j\rangle_N, E_j = \theta_j - U_j - m_j$. Omitting j, the exact component dynamics are $\theta = U + m + E, \dot{m} = \kappa E - \lambda m, \dot{z} = 2z(gE - \lambda)$. Define $\Delta = \kappa + \lambda, f = \frac{\lambda}{\Delta}$. Before the reusable circuit grows, memory relaxes to residual $E \approx f\theta$. A path grows exactly when $gE > \lambda$; after memory relaxation, while the circuit remains small, its growth is governed by $gf\theta - \lambda$. Writing $g_{\pi,j*} = \max_{a\in I_{\pi,j}} g_{\pi,ja}$, the winning path in sector j has early growth rate

$$r_{\pi,j} = f\theta_{\pi,j} g_{\pi,j*} - \lambda. \tag{5}$$

Thus weight decay prevents memory from absorbing the target completely, while routing controls how strongly the remaining signal reaches each reusable sector.

Assumption 3 (Threshold margin)**.** *For each compared arm, the held-out threshold δ lies strictly between two successive sector-learning stages. We write $r_{\pi,\delta} > 0$ for the rate of the slowest sector group that must emerge for the held-out error to cross δ. The stages and strict threshold inequalities are those of Assumption F in Appendix A.3.*

Different sectors may grow at different rates, so crossing δ need not require every reusable component to be learned. The critical rate $r_{\pi,\delta}$ therefore identifies the last rate group required for generalization at the chosen threshold.

2.3 First passage and the effect of grouping

Theorem 1 (Route-dependent first passage)**.** *Under Assumptions 1, 2, and 3 (formally, Assumptions D–F of Appendix A),*

$$T_\pi(\delta) = \frac{\log(\alpha^{-2})}{2r_{\pi,\delta}} + O(1). \tag{6}$$

Consequently, $r_{A,\delta} > r_{B,\delta}$ implies $T_A(\delta) < T_B(\delta)$ for all sufficiently small α. For each fixed $\eta > f^2\|y\|_N^2$, there is a finite t_η independent of α such that

$$T_\eta^{\mathrm{tr}} \leq t_\eta < T_\pi(\delta) \quad \textit{for all sufficiently small } \alpha. \tag{7}$$

Theorem 1 shows that memorization occurs on a bounded timescale, while generalization is governed by the growth of the slowest reusable component required to cross the held-out threshold. In comparisons, A denotes the aligned arm and B any other arm satisfying the same assumptions; $O(1)$ denotes a remainder bounded as $\alpha \downarrow 0$ with the model and strict margins fixed. The complete proof is in Appendix E.

Factorial lookup instance. We instantiate Theorem 1 on an additive factorial task $y = g_r(x) + h_c(x')$ where training observes a subset $\mathcal{T}$ of row–column pairs and evaluation uses held-out combinations $\mathcal{H}$. Mean-zero orthonormal features ϕ_j span the row and column rules. Each routed block contains a copy of the row dictionary $F^{\text{row}}_{rj\pm} = \pm\mathbf{1}_{\{R=r\}}\phi_j(x)$, while a separate always-on circuit contains the analogous column dictionary. Exact operand balance makes these row and column modes orthogonal. Let n_r be the number of observed cells containing row rule r, and let $c_{k,r}$ of them be routed to block k. The corresponding training mass is $m_{k,r} = \frac{c_{k,r}}{|\mathcal{T}|}$. Aligned routing pools all n_r cells of a row into one copy, the size-matched scramble distributes them across copies, and fragmentation spreads them evenly across blocks.

The full design are given in Appendix A.4. Proposition 2 shows that aligned routing pools same-rule signal into a single parameter copy, increasing the relevant reusable-component growth rates and, when the critical rate increases, yielding earlier generalization than unaligned or fragmented routing. Appendix F gives the full proof.

Proposition 2 (Factorial lookup instance)**.** *On the exact factorial design, every nonzero row coefficient in a trained copy has rate* $r_{k,r,j} = f m_{k,r}\beta_{rj} - \lambda, \qquad m_{k,r} = c_{k,r}/|\mathcal{T}|$. *For any fixed partition with its stated held-out selector, corresponding aligned row rates are no smaller and their equilibrium error contributions are no larger; column rates and contributions are unchanged. If both arms satisfy Assumptions D–F, then* $r_{A,\delta} \geq r_{\pi,\delta}$. *Whenever this inequality is strict,* $T_A(\delta) < T_\pi(\delta)$ *for all sufficiently small* α*; equality does not order the bounded remainders.*

For the three specified checks, if their threshold-dependent rates obey $r_{A,\delta} > r_{\text{scr},\delta} > r_{\text{frag},\delta}$, *then*

$$T_A(\delta) < T_{\text{scr}}(\delta) < T_{\text{frag}}(\delta) \quad \textit{for all sufficiently small } \alpha. \tag{8}$$

Under the globally slowest conditions equation 33 for all three arms, a sufficient condition is $1 < n_r < K$ *for every held-out-active row and a common column bottleneck no slower than the aligned row bottleneck.*

2.4 Aligned routing versus a single block

We next ask why adding computation-aligned grouping can accelerate a model that already uses weight decay. Theorem 3 isolates two sufficient mechanisms: weaker effective access to an otherwise available reusable computation, and the need to construct rule selectivity inside a shared pathway. These are stylized comparators for possible behavior of the learned weight-decay-only model, not an assumption that its router necessarily uses a single expert. Appendix G gives both proofs.

Two ungrouped comparators. We consider two stylized ways in which a weight-decayed model without computation-aligned routing may be slower. First, the *supplied* comparator $\mathsf{C}_{\text{sup}}(\gamma)$ has the same reusable directions as the aligned model, but scales their effective gain by $\gamma \in (0, 1]$. Second, the *built* comparator $\mathsf{C}_{\text{built}}(\gamma)$ represents a shared pathway that must learn rule selectivity internally: $\widehat{y}_{\text{built}}(c, x) = \gamma \sum_{n=1}^{n_u} o_n \underbrace{(w_n^\top e_c)}_{\text{rule selector}} \underbrace{(q_n^\top \phi(x))}_{\text{reusable feature}}$. Thus part (i) isolates weaker access to an already available reusable computation, while part (ii) bounds growth through the three-factor realization of internal rule selectivity. Both comparators use the same weight decay and private-memory mechanism as the aligned model. We use the built instance of Appendix A.5, with $c = r$ indexing the G row rules. The features ϕ span these rules and the fixed width satisfies $n_u \geq G$, so the row target is representable. Initialize every built-factor entry with magnitude at most α, and the column circuit and memory as in Assumption 1. The factorial instance requires the row component $y_{\text{row}} = g_r(x)$ to pass through this three-factor pathway, while the separate column circuit cannot explain it. Let $Q_{\text{row}} = \frac{\|y_{\text{row}}\|_*^2}{Y_*^2}$. The fixed constants C_0 and C_{out} used to bound factor growth and output size are defined in Appendix A.5.

Theorem 3 (Single-block comparison)**.** *(i)* Supplied block. *Under Assumptions D–F for both the aligned arm and* $\mathsf{C}_{\text{sup}}(\gamma)$, *the supplied rates and floor are*

$$r_j(\gamma) = f\gamma\theta_j g_{j*} - \lambda, \qquad L_\gamma = \frac{\|y_\perp^*\|_*^2 + \sum_{j\in J_A^+} \rho_j^2 \min\{\theta_j, \Delta/(\gamma g_{j*})\}^2}{Y_*^2}. \tag{9}$$

For $0 < \gamma < 1$, $T_{\mathsf{C}} > T_A$ *for all sufficiently small* α.

(ii) Built block. *In the built-block instance above, suppose the row energy satisfies $Q_{\mathrm{row}} > \delta$. Then $T_{\mathsf{C}} \geq 1/(2C_0\alpha)$ for all sufficiently small α. If the aligned arm also satisfies Assumptions D–F at this threshold, then $T_{\mathsf{C}} > T_A$ for all sufficiently small α. Within the same sufficiently-small-initialization regime, if also $\alpha < \lambda/C_0$ and $C_{\mathrm{out}}\alpha^3 < \sqrt{Q_{\mathrm{row}}} - \sqrt{\delta}$, then $T_{\mathsf{C}} = \infty$.*

With fixed positive decay, part (ii) gives permanent trapping for all sufficiently small α.

We validate the theoretical mechanism in a finite Python simulation of the same gradient-flow model on the factorial lookup task. The simulation verifies the predicted first-passage ordering $T_A < T_{\mathrm{scr}} < T_{\mathrm{frag}}$, the logarithmic dependence of passage time on α^{-2} with slope close to $1/(2r_{\pi,\delta})$, and the memorization-before-generalization separation at sufficiently small initialization. An independent integration of the full factor and memory dynamics also matches the reduced sector equations to numerical precision. Full simulation details, parameter values, trajectories, and additional Gaussian and built/bypass controls are given in Appendix H.

3 Empirical Evaluation of Computation-Aligned Routing

We empirically test the analysis in §2 with sparse MoEs at three scales, from a 2.3M-parameter model trained from scratch to a pretrained 7B OLMoE. Our benchmark exposes reusable row and column factors individually during training while withholding selected combinations, so that semantic routing can be compared directly with both cardinality- and size-matched random groupings under the same weight decay. The theoretical analysis uses fixed routes and exact sectors, whereas the Transformer learns its routes and representations and is not constrained or verified to realize that sector structure. The experiments test whether the predicted ordering persists.

3.1 Factorial compositional benchmark

We construct 20 modular operations from a 5×4 factorial grid over $\mathbb{Z}_{67}$. Each step contains opaque row and column markers R_r and C_c together with arguments (x, x'), and its target is

$$y = (g_r(x) + h_c(x')) \bmod 67, \qquad g_r(x) \in \{x, -x, 2x, x^2, x^3\}, \quad h_c(x') \in \{x', -x', 2x', (x')^2\}. \tag{10}$$

The model sees only the marker identities, never the formulas. A step is serialized as `V{x} R{r} C{c} V{x'} <DEC> V{y}`, and both cross-entropy and route alignment act only at the answer token following `<DEC>`. Four cells, R_0C_0, R_1C_3, R_3C_2, and R_4C_1, never appear in training, although each of their row and column markers appears in other cells. The training set contains 29,552 examples—single steps from the 16 observed cells and multi-step programs of length two to four—and every model fits it long before it generalizes. Our primary measure of generalization is accuracy on all 17,956 argument pairs of the four unseen cells (*factor recombination*); argument, composition, and crossed shifts are reported in Appendix J.4, and Appendix I.1 gives the complete data specification.

3.2 The routing objective

Consider a sparse MoE with L routing layers and router logits $W_\ell h_{\ell i}$ for scored position i at layer $\ell \in \{1, \ldots, L\}$. The native forward pass selects its usual top-k experts. For the auxiliary objective, we retain the full soft distribution $q_{\ell i} = \mathrm{softmax}(W_\ell \ \mathrm{stopgrad}(h_{\ell i}))$. The stop-gradient makes the auxiliary loss update the routers but not the hidden representations or experts directly.

Let G be the number of observed computational groups, let $g(i) \in \{1, \ldots, G\}$ denote the group of scored answer position i, and let D_g be all such positions in group g in the frozen training corpus. At every update, we first make a no-gradient pass over the complete corpus and form the group routing prototype $\mu_{\ell g} = \frac{1}{|D_g|} \sum_{i \in D_g} q_{\ell i}$. Holding these prototypes fixed during the update, we minimize the mean within-group Jensen–Shannon divergence, giving every observed group equal weight:

$$\mathcal{L}_{\mathrm{route}} = \frac{1}{L} \sum_{\ell=1}^{L} \frac{1}{G} \sum_{g=1}^{G} \frac{1}{|D_g|} \sum_{i \in D_g} \mathrm{JS}(q_{\ell i} \,\|\, \mu_{\ell g}) \,. \tag{11}$$

Setting	XS (2.3M)			M (107M)			OLMoE-1B-7B		
	T_{50}	T_{90}	T_{99}	T_{50}	T_{90}	T_{99}	T_{50}	T_{90}	T_{99}
Aligned row	**14.9K**	<u>23.8K</u>	<u>28.6K</u>	**3.1K**	**4.4K**	**5.1K**	<u>364</u>	<u>494</u>	888
Matched-random row	23.1K	32.3K	34.7K	<u>3.3K</u>	<u>4.7K</u>	<u>5.7K</u>	386	641	<u>855</u>
Aligned column	<u>16.8K</u>	**22.9K**	**24.9K**	3.7K	5.2K	6.0K	**346**	**481**	**759**
Matched-random column	23.4K	33.2K	34.9K	3.4K	4.9K	6.4K	432	738	1024
Weight decay only	75.6K	138.7K	>235K	>7K	>7K	>7K	374	608	977
No weight decay	>125K	>125K	>125K	>7K	>7K	>7K	613	1428	>1.5K
Gap, row / column (pts)	+24.3 / +25.7			+3.6 / −4.5			+2.4 / +9.6		

Table 1: **Updates to reach** 50%**,** 90%**, and** 99% **held-out accuracy** on unseen factor combinations, from the averaged curves of Figure 2 (OLMoE in fine-tuning steps). Bold marks the earliest crossing in each column and underline the second earliest; "$>$" denotes right-censoring at the end of the averaged curve. The last row gives the aligned minus matched-random gap for rows and columns, in accuracy points. "Aligned" denotes computation-aligned grouping with COMROUTE.

The complete objective is:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{route}}\mathcal{L}_{\text{route}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}} + \lambda_z\mathcal{L}_z, \tag{12}$$

with decoupled weight decay applied by the optimizer (Loshchilov & Hutter, 2019). Because the objective averages group means rather than examples, its strength depends on the partition. We therefore compare each computation-aligned grouping only with a fixed random partition having the same number of groups and the same trainable-cell group-size multiset.

Weight decay supplies a global preference for smaller parameter norms; it does not identify which training examples should update the same subnetwork. COMROUTE encourages examples governed by the same factor to select overlapping experts and thereby update shared parameters, and the aligned settings use the same weight decay as their matched-random and ungrouped controls. In the route model of §2, the two effects are distinct: decay fixes the residual fraction f, while the route fixes the sector gains that enter the escape rates of Theorem 1. For fixed partitions with the required sector structure, the model describes routing through the mass of each row assigned to each block; an earlier threshold crossing follows only when the critical rates and threshold margins imply it (Theorem 1, Proposition 2). The six-setting comparison tests whether computation-aligned grouping also advances generalization under learned routing.

3.3 MODELS, SETTINGS, AND METRICS

We train two sparse MoEs from scratch, **XS** (2.3M parameters) and **M** (107M), and fully fine-tune the pretrained **OLMoE**-1B-7B (Muennighoff et al., 2024) (Appendix I.2). At every scale we compare six settings that differ only in weight decay and route grouping: *no weight decay*; *weight decay only* (0.1); *computation-aligned row* and *column* grouping (hereafter *aligned*), which align routing within the five row or four column groups (Figure 1a); and a *matched-random* control for each, which scrambles the 16 observed cells into the same number of groups with the same sizes. All grouped settings use weight decay 0.1 and $\lambda_{\text{route}} = 0.3$, and each computation-aligned grouping is compared only with its own matched control (Appendix I.4).

We measure how soon a setting generalizes by T_τ, the first update at which mean held-out accuracy reaches $\tau \in \{50, 90, 99\}\%$. To summarize the whole transition, the *gap* is the time-averaged accuracy advantage of aligned over matched-random grouping while the grouped settings are still learning; a gap of $+10$ means ten points more accurate on average. Curves average independent runs and show a min–max band; Appendix I.6 specifies interpolation, censoring, and the averaging window.

4 RESULTS

4.1 COMPUTATION-ALIGNED GROUPING MAKES GENERALIZATION ARRIVE SOONER

Figure 2 and Table 4.1 show the grokking separation at every scale: all settings fit the training set almost immediately, while accuracy on the unseen combinations stays low for much longer, and

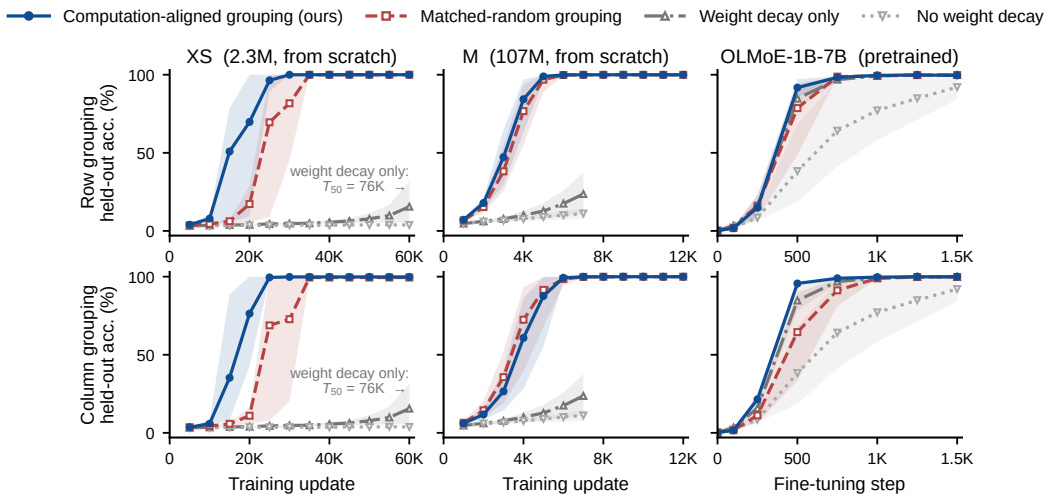

Figure 2: **Held-out accuracy on unseen factor combinations across three model scales** (mean over runs; bands span the lowest to the highest run). Top: row grouping; bottom: column grouping. Every setting fits the training set by the first evaluation.

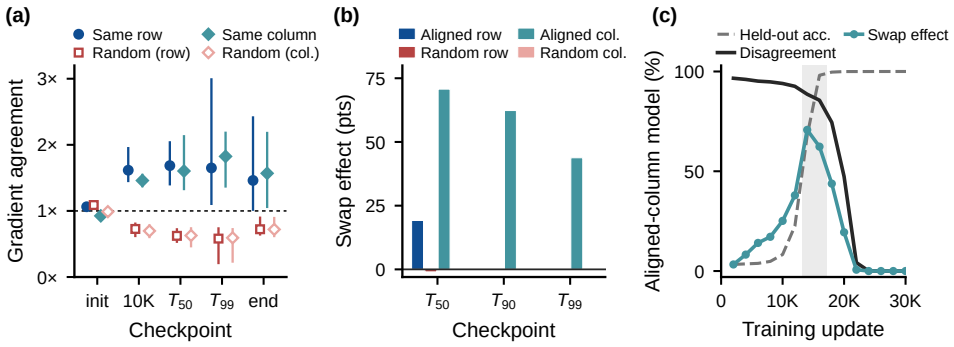

Figure 3: **Expert analysis of the XS model** (seed 23). "Aligned" denotes computation-aligned grouping with ComRoute and "random" its matched-random control. **(a)** Gradient agreement within an expert between examples from different cells that share a row, a column, or a matched-random group, relative to the average pair ($1\times$, dashed). Markers are means over the four trained models and whiskers span their range. **(b)** Route swap at <DEC>: the increase in held-out answers that become another group's answer when the last-layer route is replaced by that group's expert. **(c)** In the aligned-column model the swap effect peaks at the transition (grey band, T_{50}–T_{99}) and vanishes once the last-layer experts stop disagreeing. The row model is in Appendix Figure 11.

how long depends on how routing is grouped. At every scale and threshold, the earliest crossing belongs to a aligned setting. The effect is largest in the small model trained from scratch, where computation-aligned grouping reaches 90% held-out accuracy with about $1.4\times$ fewer updates than matched-random grouping and $6\times$ fewer than weight decay alone, before its matched control in every run and for rows and columns alike; without weight decay, no run generalizes within its horizon. At scale M, weight decay alone no longer generalizes within 12.4K updates, whereas every grouped setting does by 6.4K, so constraining routing at all provides much of the speedup; computation-aligned grouping then helps for rows ($+3.6$ points) but not for columns (-4.5). Pretrained OLMoE leaves less delay to remove, since weight decay alone reaches 90% within 608 steps, yet aligned column grouping still shortens it to 481, ahead of its matched control in every run. The other OOD splits follow the same ordering (Appendix J.4). These gains concern the timing of the transition rather than final accuracy, which every generalizing setting eventually matches (Appendix J.1).

4.2 What the experts learn

Why does computation-aligned grouping accelerate generalization? We analyze saved checkpoints of the XS seed-23 run, marking each setting's T_{50}, T_{90}, and T_{99}; Appendix K defines each measurement.

Shared factors align expert gradients. Within each expert, training examples from different cells that share a row function have gradients that agree up to $3.0\times$ as much as an average pair, and those sharing a column function up to $2.2\times$, whereas pairs from the same matched-random group agree only 0.2–$0.9\times$ as much (Figure 3a). The ordering holds even in the ungrouped model, so grouping by a shared factor routes compatible updates to the same expert, consistent with the mechanism in §2.

The grouped route selects the factor function. On the unseen combinations, replacing the last layer's route at <DEC> with the expert that another column normally uses makes the aligned-column model output that column's answer 71, 62, and 44 points more often at T_{50}, T_{90}, and T_{99}; the row swap moves up to 19 points, and matched-random models do not respond (Figure 3b). When the model first generalizes, the aligned route decides which factor function is applied.

Specialization is a scaffold, not the endpoint. The swap effect vanishes about 5K updates after generalization, as the last-layer experts converge to the same function (Figure 3c). Matched-random and ungrouped models converge in the same way once they generalize (Appendix K), so computation-aligned grouping appears to accelerate grokking by letting the model first generalize through factor-specific experts that are later consolidated.

5 Related Work

Grokking and its acceleration. Grokking was first documented as generalization that arrives long after the training set is fit (Power et al., 2022). Weight decay can drive this transition by favoring parameter-efficient generalizing circuits (Varma et al., 2024), and recent methods shorten it through gradient filtering, representation transfer, or geometric regularization (Lee et al., 2024; Xu et al., 2025; Pasand & Dohmatob, 2026). None of these specifies which examples should share parameters; ComRoute supplies exactly this structure and leaves the optimizer unchanged.

Sparse experts and routing. Sparse MoE layers scale capacity by activating only a few experts per input (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). Routing objectives mostly target load balance and training stability (Lewis et al., 2021; Zoph et al., 2022), and even fixed hash routing can rival learned routers (Roller et al., 2021), so routes need not reflect meaningful structure. Routing conditioned on task or domain labels (Kudugunta et al., 2021; Gururangan et al., 2022) or aligned with task embeddings (Li et al., 2026) aims at efficiency or final accuracy, and modular architectures alone do not guarantee useful specialization (Mittal et al., 2022). We instead ask whether routing examples together because they share a known computation shortens the delay before generalization.

Compositional generalization. Neural sequence models often fail to recombine familiar parts in unfamiliar ways (Lake & Baroni, 2018; Keysers et al., 2020). Modular and routed architectures can help (Andreas et al., 2016; Rosenbaum et al., 2018; Alet et al., 2019; Schug et al., 2024), but only when their structure matches the task (Bahdanau et al., 2019); composition is also tied to grokking (Wang et al., 2024) and to skill reuse in agentic RL (Xu et al., 2026b). Our benchmark makes the decomposition explicit, so we ask not whether a model can compose but how aligning parameter sharing with it changes *when* it learns to.

6 Conclusion

We introduce ComRoute, a training incentive for computation-aligned parameter sharing on top of weight decay. Our theory shows that pooling examples that share a reusable computation can increase the growth rate of the components needed for generalization, while experiments show substantially earlier generalization than matched-random grouping and weight decay alone in the clearest regime. These results suggest that accelerating grokking depends not only on favoring reusable solutions, but also on organizing where they are learned.

AI USE STATEMENT

Generative AI tools were used to support the preparation of this work in three ways. First, they assisted with writing and presentation, including revising prose, improving the exposition of the theoretical arguments, and checking the scope and wording of claims. Second, they were used for retrieval and discovery, including locating and checking related work and references. Third, they supported research ideation and execution, including reviewing proof arguments, writing and running verification code, inspecting numerical checks, and assisting with LaTeX compilation and document layout.

REFERENCES

Ferran Alet, Tomás Lozano-Pérez, and Leslie P. Kaelbling. Modular meta-learning, 2019. URL https://arxiv.org/abs/1806.10166.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL https://arxiv.org/abs/1511.02799.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1811.12889.

Ryan Burnell, Yumeya Yamamori, Orhan Firat, Kate Olszewska, Steph Hughes-Fitt, Oran Kelly, Isaac R. Galatzer-Levy, Meredith Ringel Morris, Allan Dafoe, Alison M. Snyder, Noah D. Goodman, Matthew Botvinick, and Shane Legg. Measuring progress toward agi: A cognitive framework, 2026. URL https://arxiv.org/abs/2605.28405.

Yuan Cao, Zixiang Chen, Mikhail Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier ai models, 2025. URL https://arxiv.org/abs/2405.21015.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced, 2018. URL https://arxiv.org/abs/1806.00900.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022. URL https://aclanthology.org/2022.naacl-main.407.

Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity, 2022. URL https://arxiv.org/abs/2106.15933.

Daniel Keysers, Nathanael Sch"arli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021. URL https://arxiv.org/abs/2110.03742.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, 2018.

J. LaSalle. Some extensions of liapunov's second method. *IRE Transactions on Circuit Theory*, 7(4): 520–527, 1960. doi: 10.1109/TCT.1960.1086720.

Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Gradient filters for faster grokking, 2024. URL https://openreview.net/forum?id=UD0L74wQt9.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2021. URL https://arxiv.org/abs/2103.16716.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=AHOs7Sm5H7R.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. Routing manifold alignment improves generalization of mixture-of-experts LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/249b8d8f41970822651435629e68a6e1-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a modular architecture enough? In *Advances in Neural Information Processing Systems*, 2022.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Ali Saheb Pasand and Elvis Dohmatob. Egalitarian gradient descent: A simple approach to accelerated grokking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wCnHeql3ow.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.04426.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ry8dvM-R-.

Andrew Saxe, Shagun Sodhani, and Sam Jay Lewallen. The neural race reduction: Dynamics of abstraction in gated networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19287–19309. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/saxe22a.html.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL https://arxiv.org/abs/1312.6120.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wolczyk, Alexandra Maria Proca, Johannes von Oswald, Razvan Pascanu, Joao Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=H98CVcX1eh.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Vikrant Varma, Rohin Shah, Zachary Kenton, Janos Kramar, and Ramana Kumar. Explaining grokking through circuit efficiency, 2024. URL https://openreview.net/forum?id=7Zbg38nA0J.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15071.

Mingyue Xu, Gal Vardi, and Itay Safran. To grok grokking: Provable grokking in ridge regression. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=5nNNVY8NW4.

Zhikun Xu, Yu Feng, Jacob Dineen, Taiwei Shi, Jieyu Zhao, and Ben Zhou. Skill reuse as compression in agentic RL. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026b. URL https://arxiv.org/abs/2605.31509.

Zhiwei Xu, Zhiyu Ni, Yixin Wang, and Wei Hu. Let me grok for you: Accelerating grokking via embedding transfer from a weaker model. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4rEI2JdHH6.

Lexin Zhou, Lorenzo Pacchiardi, Fernando Martínez-Plumed, Katherine M. Collins, Yael Moros-Daval, Seraphina Zhang, Qinlin Zhao, Yitian Huang, Luning Sun, Jonathan E. Prunty, Zongqian Li, Pablo Sánchez-García, Kexin Jiang Chen, Pablo A. M. Casares, Jiyun Zu, John Burden, Behzad Mehrbakhsh, David Stillwell, Manuel Cebrian, Jindong Wang, Peter Henderson, Sherry Tongshuang Wu, Patrick C. Kyllonen, Lucy Cheke, Xing Xie, and José Hernández-Orallo. General scales unlock ai evaluation with explanatory and predictive power, 2025. URL https://arxiv.org/abs/2503.06378.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

A FORMAL SETUP AND COMPARISON INSTANCES

This appendix supplies the definitions and assumptions used by the three main results in §2. The detailed formulation below defines the task, routes, loss and rate mechanism; the model and sector assumptions then make that mechanism exact.

Guide to the proofs. Theorem 1 uses Assumptions D–F, stated in Appendices A.2–A.3. Its proof in Appendix E combines the exact reduction in Appendix C with the exit, settling and uniform bounds in Appendix D; the uniform bounds exclude an earlier transient threshold crossing. Proposition 2 applies that theorem to the design in Appendix A.4, with proof in Appendix F. Theorem 3 uses the comparators and built instance in Appendix A.5; Appendix G proves its supplied-gain and three-factor conclusions separately.Finite-sample checks are in Appendix H.

A.1 PROBLEM FORMULATION: COMPUTATION GROUPING, WEIGHT DECAY, AND TIME TO GENERALIZE

Task. Training examples are $\xi_i = (c_i, \nu_i, x_i)$, $i = 1, \dots, N$, with rule label $c_i \in \{1, \dots, G\}$, auxiliary partition label $\nu_i \in \{1, \dots, K\}$ and operand representation $x_i \in \mathbb{R}^d$. For one rule circuit, the target is

$$y_i = W_{c_i}^\top x_i, \tag{13}$$

where fixed $W_c \in \mathbb{R}^d$ represents the rule in these operand features. Sums of such circuits give the lookup target $g_r(x) + h_c(x')$: held-out law P_* uses unobserved (r, c) cells whose individual rules occur in training. Define $\langle a, b\rangle_N = N^{-1}\sum_i a(\xi_i)b(\xi_i)$ and $\langle a, b\rangle_* = \mathbb{E}_{P_*}[ab]$, with corresponding norms and target RMS $Y_* = \|y\|_* > 0$. For a model indexed by route π, relative held-out MSE and its first passage are defined in Eq. equation 4.

The quantity studied is $T_\pi(\delta)$, with fixed $\delta > 0$. Grokking means that training-error first passage T_η^{tr} precedes this held-out crossing; Theorem 1 specifies the admissible training level η.

Computation grouping. A hard grouping assigns examples to blocks through $\nu_\pi : \{1, \dots, N\} \to \{1, \dots, K\}$, implemented by a fixed route R_π:

$$R_\pi(c_i, \nu_i) = e_{\nu_\pi(i)}, \qquad \nu_A(i) = c_i, \quad R_A(c, \nu) = e_c \quad (K \geq G). \tag{14}$$

Here e_k is the kth coordinate vector. The aligned assignment A pools each rule into its own block. A size-matched unaligned partition has the same multiset of group sizes but splits rules among blocks; equal fragmentation distributes each rule evenly across all blocks. Two examples share a parameter block precisely when their assignments $\nu_\pi(i)$ agree. Routes are also defined on held-out inputs; their compatibility is required below.

Parameter sharing induced by a grouping. Block k holds products $z_{kq} = u_{kq}v_{kq}$ on fixed features $F_q(c, x)$, $q = 1, \dots, J$. A routed column $D_{\pi,kq}$ selects the examples that train this coefficient; private coordinate h_i fits example i alone. Equation equation 1 gives the prediction with memory mobility $\kappa > 0$.

In an MoE, a block corresponds to an expert. With residual $e_i = y_i - \widehat{y}_i$, hard grouping gives the coefficient-gradient identity

$$-\frac{\partial}{\partial z_{kq}}\left[\frac{1}{2N}\sum_i (y_i - \widehat{y}_i)^2\right] = \frac{1}{N}\sum_{i:\,\nu_\pi(i)=k} e_i F_q(c_i, x_i) = \langle D_{\pi,kq}, e\rangle_N. \tag{15}$$

Grouping therefore determines which examples update the same coefficient. On the exact lookup dictionary below, distinct rules occupy orthogonal directions, and alignment collects a row mode's signal in one copy; sharing alone does not guarantee coherent gradients for arbitrary dictionaries.

Weight decay. Gradient flow minimizes the regularized loss in Eq. equation 2, with common decay $\lambda > 0$.

Decay contributes $-\lambda$ times each parameter to its velocity. For memory prediction $m_i = \sqrt{\kappa N}h_i$ and balanced factors $u_{kq} = v_{kq}$,

$$\dot{m}_i = \kappa e_i - \lambda m_i, \qquad \dot{z}_{kq} = 2z_{kq}\big(\langle D_{\pi,kq}, e\rangle_N - \lambda\big). \tag{16}$$

With reusable output held at zero, memory equilibrates at

$$\kappa e_i = \lambda m_i, \qquad e_i = f y_i, \tag{17}$$

where the relaxation rate Δ and surviving residual fraction f are defined in Eq. equation 2.2.

Thus memory alone leaves a persistent learning signal under decay; a circuit coefficient grows only if its residual correlation exceeds λ. The auxiliary memory-only system with zero decay instead has residual tending to zero.

Why grouping changes the time to generalize. Under the sector assumptions of Appendix A.3, each independent component has target strength θ, reusable fit U, private fit m and remaining residual E:

$$\boxed{\theta = U + m + E.}$$

For a path product $z = uv$ with signed gain g on that component, U sums contributions gz, and the exact projected dynamics are

$$\dot{m} = \kappa E - \lambda m, \qquad \dot{z} = 2z(gE - \lambda).$$

After memory relaxes, while U remains small, $E \approx f\theta$. Index components by j and paths by $a = (k, q)$; let $g_{\pi,j*}$ be the largest signed gain in component j. A winning path has growth exponent $2r_{\pi,j}$, where

$$r_{\pi,j} = f\theta_{\pi,j} g_{\pi,j*} - \lambda, \qquad \dot{z}_a \approx 2r_{\pi,j} z_a \quad \text{for a winner } a. \tag{18}$$

Under Assumptions D–F, Theorem 1 turns this small-initialization growth into the first-passage law

$$T_\pi(\delta) = \frac{\log(\alpha^{-2})}{2r_{\pi,\delta}} + O(1). \tag{19}$$

Here α is the factor initialization scale and $r_{\pi,\delta}$ the slowest rate group the threshold requires. Grouping preserves f, decay and the underlying target, but can change both normalized coefficients θ and gains g. Let $m_{k,r} > 0$ be row-r mass in a trained copy, as a fraction of *all* training examples, and $\beta_{rj} > 0$ a Fourier coefficient magnitude; the exact lookup design gives

$$\theta_{k,r,j} = \beta_{rj}\sqrt{m_{k,r}}, \qquad g_{k,r,j*} = \sqrt{m_{k,r}}, \qquad r_{k,r,j} = f m_{k,r}\beta_{rj} - \lambda. \tag{20}$$

For n_r observed cells in row r and observed-cell set $\mathcal{T}$, alignment, the specified row-diverse scramble and fragmentation deliver masses $n_r/|\mathcal{T}|$, $1/|\mathcal{T}|$ and $n_r/(K|\mathcal{T}|)$, respectively. These formulas give two conditional claims:

1. *Alignment versus partitions.* For arms satisfying Assumptions D–F, Proposition 2 gives no-smaller aligned critical rates for any fixed partition on this design. Strict critical-rate separation then gives, for sufficiently small α,

$$r_{A,\delta} > r_{\mathrm{scr},\delta} > r_{\mathrm{frag},\delta} \quad \Longrightarrow \quad T_A(\delta) < T_{\mathrm{scr}}(\delta) < T_{\mathrm{frag}}(\delta). \tag{21}$$

2. *Alignment versus one block.* A supplied block multiplies every aligned column by fixed gain γ; for held-out target sectors J_A^+,

$$r_j(\gamma) = f\gamma\theta_j g_{j*} - \lambda < r_j(1) \quad (j \in J_A^+,\ 0 < \gamma < 1). \tag{22}$$

   Theorem 3(i) gives a later crossing if both arms satisfy Assumptions D–F. For the stated factorial built-block instance, row prediction passes through three small factors; when row energy is required by the threshold, Theorem 3(ii) instead gives

$$T_{\mathsf{C}}(\delta) \geq \frac{1}{2C_0\alpha} \quad \text{for sufficiently small } \alpha. \tag{23}$$

   The fixed gradient bound C_0 is defined in Appendix A.5. This lower bound exceeds the aligned logarithmic clock when the aligned assumptions hold. With fixed positive decay and the stated row-energy condition, the built arm is permanently trapped for all sufficiently small α. The built comparator changes factorization; it retains the same memory and decay.

Grouping as a training objective. The fixed route supplies parameter sharing directly. In the experiment, $\mathcal{L}_{\mathrm{route}}$ in Eq. equation 11 encourages each example's routing distribution to match its group prototype. It encourages within-rule agreement but does not force distinct rules to select distinct blocks: even a globally constant router has zero alignment loss. The theorems analyze the specified fixed routes; noncollapse, effective gains and held-out routing remain bridge conditions.

A.2 MODEL

Each block uses two-factor dictionary coefficients (Saxe et al., 2014); row and column circuits have disjoint factors and shared memory. Private coordinates vanish on held-out examples, leaving component error $\theta - U$ before held-out scaling. The assumptions fix the model as initialization shrinks; Figure 4 summarizes it.

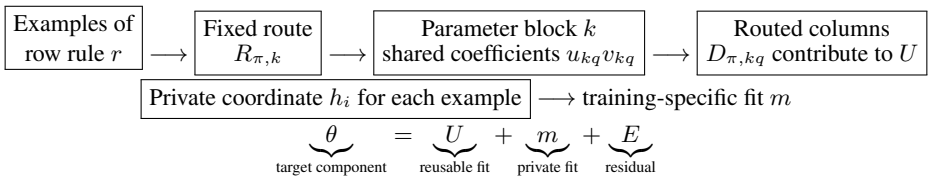

Figure 4: The route determines which examples train each reusable coefficient; private memory fits individual training examples and vanishes on held-out cells.

Assumption D (Model and initialization)**.** *Use the squared-loss gradient flow equation 1–equation 2, with disjoint private coordinates that vanish on held-out examples and common decay $\lambda > 0$ on every trainable parameter. The finite data, held-out law, dictionaries, routes, dimensions, $\kappa, \lambda > 0$, and initialization weights are fixed as $\alpha \downarrow 0$. Writing $a = (k, q)$ for a routed-column index, two-factor paths and private coordinates start at*

$$u_a(0) = v_a(0) = \alpha\sqrt{c_a}, \qquad h_i(0) = 0. \tag{24}$$

Here $c_a > 0$. A block with a different number of factors uses the entrywise initialization specified with that block.

Definition 4 (Routes compared)**.** Let $e_k \in \mathbb{R}^K$ be the kth coordinate vector and assume $K \geq G$ when using one block per rule. The *aligned* route is $R(c, \nu) = e_c$. The *matched-random* route is $R(c, \nu) = e_\nu$, where ν labels a fixed unaligned partition matched in group size; Proposition 2 specifies the finite partition used below. A *fragmented* route splits each rule's examples across fixed blocks; Proposition 2 specifies the exact realized balance used in our check.

A.3 COMPONENTS AND THRESHOLD

A sector makes the componentwise reduction in §A.1 exact: columns within it learn one direction, and orthogonal sectors do not directly interfere. On the grid these are row r, Fourier mode j, in block k, or column c, mode j. Normalized directions $\Phi_{\pi,j}$ identify components; $\rho_{\pi,j}$ transfers their coefficients to held-out directions $\Phi^*_{\pi,j}$. Write $z_a = u_a v_a$ for a path coefficient.

Definition 5 (Admissible rank-one route sectors)**.** For arm π, let $I_{\pi,j}$, $j \in \{1, \ldots, M_\pi\}$, partition its nonzero routed columns into M_π sectors. In sector j, let $\Phi_{\pi,j}$ be a unit-norm training function and $g_{\pi,ja} \in \mathbb{R}$ the signed gain of column a. Let $\rho_{\pi,j} \geq 0$ be its training-to-held-out multiplier. Require

$$D_{\pi,a} = g_{\pi,ja}\Phi_{\pi,j} \quad \text{on training data}, \qquad D_{\pi,a} = \rho_{\pi,j} g_{\pi,ja}\Phi^*_{\pi,j} \quad \text{on held-out data}. \tag{25}$$

The functions $\Phi_{\pi,j}$ are training-orthonormal. The functions $\Phi^*_{\pi,j}$ for $\rho_{\pi,j} > 0$ are held-out-orthonormal. If $\rho_{\pi,j} = 0$, the held-out column vanishes and no $\Phi^*_{\pi,j}$ is needed. Columns that vanish on both supports can be omitted from the sectors; their factors simply decay.

The training target coefficient is $\theta_{\pi,j} = \langle y, \Phi_{\pi,j}\rangle_N$. Orient each nonzero coefficient so that $\theta_{\pi,j} > 0$. Compatibility requires

$$\langle y, \Phi^*_{\pi,j}\rangle_* = \rho_{\pi,j}\theta_{\pi,j} \quad (\rho_{\pi,j} > 0). \tag{26}$$

Write $y^*_{\perp,\pi}$ for the held-out target component orthogonal to all active $\Phi^*_{\pi,j}$. A *winner path* has maximal gain $g_{\pi,j*} = \max_{a \in I_{\pi,j}} g_{\pi,ja}$; a *loser path* has strictly smaller gain. The residual growth rate is

$$r_{\pi,j} = f\theta_{\pi,j} g_{\pi,j*} - \lambda. \tag{27}$$

Require $J^+_\pi := \{j : \rho_{\pi,j} > 0, \theta_{\pi,j} > 0\} \neq \varnothing$ and $g_{\pi,j*} > 0$ for $j \in J^+_\pi$. Thus at least one held-out target direction is present, and every such direction has a correctly oriented path.

Assumption E (Sector structure)**.** *Each compared two-factor route has admissible rank-one sectors as in Definition 5.*

Threshold. Private memory vanishes on held-out examples, so held-out error depends on the reusable sector fits rather than the training residual. Define

$$U_{\pi,j}(t) = \sum_{a \in I_{\pi,j}} g_{\pi,ja} z_a(t).$$

By the held-out compatibility and orthogonality conditions in Definition 5,

$$\mathcal{E}_\pi(t) = \frac{\|y^*_{\perp,\pi}\|_*^2 + \sum_{j:\rho_{\pi,j}>0} \rho_{\pi,j}^2 \big(\theta_{\pi,j} - U_{\pi,j}(t)\big)^2}{Y_*^2}. \tag{28}$$

For each held-out target sector $j \in J_\pi^+$, an absent reusable component contributes its full target energy. At a learned winner's equilibrium, $E = \lambda/g_{\pi,j*}$ and $m = \kappa/g_{\pi,j*}$, leaving gap $\theta_{\pi,j} - U_{\pi,j} = \Delta/g_{\pi,j*}$; a nonpositive-rate component remains absent in the small-initialization limit. Define its absent energy $w_{\pi,j}$, learned error contribution $\ell_{\pi,j}$, and the unexplained target energy P_π by

$$w_{\pi,j} = \frac{\rho_{\pi,j}^2 \theta_{\pi,j}^2}{Y_*^2}, \qquad \ell_{\pi,j} = \frac{\rho_{\pi,j}^2}{Y_*^2} \min\left\{\theta_{\pi,j}, \frac{\Delta}{g_{\pi,j*}}\right\}^2, \qquad P_\pi = \frac{\|y^*_{\perp,\pi}\|_*^2}{Y_*^2}. \tag{29}$$

Thus $r_{\pi,j} \le 0$ implies $\ell_{\pi,j} = w_{\pi,j}$.

List the distinct positive rates on J_π^+ as $s_{\pi,1} > \cdots > s_{\pi,H_\pi} > 0$, where H_π is the number of positive-rate groups. At stage h, define

$$\mathcal{F}_{\pi,h} = \{j \in J_\pi^+ : r_{\pi,j} \ge s_{\pi,h}\}, \qquad \mathcal{F}_{\pi,0} = \varnothing.$$

Exact rate ties enter the same group. The corresponding limiting stage risk is

$$\mathcal{E}_\pi^{[h]} = P_\pi + \sum_{j \in \mathcal{F}_{\pi,h}} \ell_{\pi,j} + \sum_{j \in J_\pi^+ \setminus \mathcal{F}_{\pi,h}} w_{\pi,j}, \qquad h = 0, \dots, H_\pi. \tag{30}$$

The zero-circuit stage has risk $\mathcal{E}_\pi^{[0]} = 1$, and the final floor is $L_\pi = \mathcal{E}_\pi^{[H_\pi]}$. The margin below selects the slowest rate group that must be learned to cross δ, allowing weaker row or column modes to remain unlearned.

Assumption F (Threshold margin)**.** *The fixed relative-MSE threshold satisfies, for every compared arm and some $h_\pi \in \{1, \dots, H_\pi\}$,*

$$\mathcal{E}_\pi^{[h_\pi]} < \delta < \mathcal{E}_\pi^{[h_\pi - 1]}, \qquad r_{\pi,\delta} := s_{\pi,h_\pi}. \tag{31}$$

The index h_π is unique. Its rate $r_{\pi,\delta} > 0$ is the critical rate required by the threshold, not necessarily the globally slowest sector rate. Equality with a stage risk is not covered.

A.4 Exact factorial lookup design

Lookup design and compared partitions. Let $\mathcal{T}, \mathcal{H} \subseteq \mathcal{R} \times \mathcal{C}$ be disjoint observed and held-out cells, with row and column labels R, C. Every evaluated rule occurs in training and the held-out target is nonzero. At cell (r, c),

$$y = g_r(x) + h_c(x'),$$

where g_r, h_c are mean-zero tables and x, x' are independent and uniform on $\mathbb{Z}_p$, $p \ge 2$. Observed cells have equal mass, and operand balance holds exactly within every training and held-out cell and every nonempty routed training-cell subset. Mean-zero orthonormal features $\{\phi_j\}_{j=1}^{d_0}$, $d_0 \le p - 1$, span the rule tables, with supplied signed dictionary

$$F^{\mathrm{row}}_{rj\pm} = \pm \mathbf{1}_{\{R=r\}} \phi_j(x), \qquad F^{\mathrm{col}}_{cj\pm} = \pm \mathbf{1}_{\{C=c\}} \phi_j(x'). \tag{32}$$

Each of $K \ge |\mathcal{R}|$ row blocks contains a copy of the row dictionary, while a separate always-on circuit contains the column dictionary. Every signed feature has its own two-factor coefficient, and all circuits share the same private memory; this construction makes distinct row and column modes orthogonal.

Let n_r and n_c denote the numbers of observed cells containing row r and column c. If block k receives $c_{k,r}$ row-r cells, or the corresponding sum of assigned fractions, then

$$\sum_k c_{k,r} = n_r, \qquad m_{k,r} = \frac{c_{k,r}}{|\mathcal{T}|}$$

is the training mass of that row copy. Define

$$\beta_{rj} = |\widehat{g}_r(j)|, \qquad \widehat{g}_r(j) = \frac{1}{p}\sum_x g_r(x)\phi_j(x).$$

Each held-out row r is evaluated through a fixed trained copy $\kappa(r)$ with $m_{\kappa(r),r} > 0$, and m_r^*, m_c^* denote held-out row and column masses. Alignment places all n_r observed cells of a row in one copy; the size-matched row-diverse scramble places one cell in each nonempty copy; equal fragmentation assigns n_r/K cells to each block. Column predictions are unchanged.

For the special case in which the threshold requires the globally slowest rate group, define

$$r_{\pi,*} = \min_{j \in J_\pi^+} r_{\pi,j}, \qquad S_\pi = \{j \in J_\pi^+ : r_{\pi,j} = r_{\pi,*}\}, \qquad E_{\pi,\mathrm{slow}} = \sum_{j \in S_\pi} w_{\pi,j}.$$

If all needed rates are positive and

$$L_\pi < \delta < P_\pi + E_{\pi,\mathrm{slow}}, \tag{33}$$

then $r_{\pi,\delta} = r_{\pi,*}$. Proposition 2 shows how concentrating a row's training mass in one parameter copy increases its reusable-component growth rates and can therefore advance generalization.

A.5 Single-block comparators and built instance

Definition 6 (Single gated block)**.** Let $\gamma \in (0, 1]$ be a fixed output gain. The supplied block $\mathsf{C}_{\mathrm{sup}}(\gamma)$ collects the aligned arm's nonzero routed columns in one block and replaces each $D_{A,a}$ by $\gamma D_{A,a}$. It preserves the factor coordinates and their initial values; inactive coordinates are retained and simply decay. The built block $\mathsf{C}_{\mathrm{built}}(\gamma)$ instead has $n_u \geq 1$ multiplicative units and output

$$\widehat{y}_{\mathrm{built}}(c, x) = \gamma \sum_{n=1}^{n_u} o_n (w_n^\top e_c)(q_n^\top \phi(x)). \tag{34}$$

Here $e_c \in \mathbb{R}^G$ is the one-hot rule vector and $\phi(x) \in \mathbb{R}^{J_\phi}$ is a fixed operand-feature vector of dimension $J_\phi \geq 1$. The scalar $o_n \in \mathbb{R}$, rule-selector vector $w_n \in \mathbb{R}^G$, and feature vector $q_n \in \mathbb{R}^{J_\phi}$ are learned and decayed. Thus one output path contains three factors. Both comparators retain the aligned arm's private memory, mobility κ, and decay λ.

Define the fixed growth and normalized-output bounds

$$C_0 = \gamma \max_i \|\phi(x_i)\|_1 (\|y\|_N + 1), \qquad C_{\mathrm{out}} = \frac{\gamma n_u \sup_* \|\phi(x)\|_1}{Y_*}. \tag{35}$$

Here $\sup_*$ denotes the supremum over held-out inputs.

Built-block instance. Use the preceding finite factorial design and Assumption D's loss, decay and fixed-model conventions, with selector $c = r$, $G = |\mathcal{R}|$. Require ϕ to span the row rules and the fixed width to satisfy $n_u \geq G$, so every row rule is representable. Only the disjoint column circuit bypasses the three-factor row paths, and the row target $y_{\mathrm{row}} = g_r(x)$ has relative energy $Q_{\mathrm{row}} = \|y_{\mathrm{row}}\|_*^2 / Y_*^2$ ($1/2$ in the equal-energy instance) orthogonal to every column output by operand independence and mean-zero tables. Initialize every built-factor entry with magnitude at most α and the column circuit and memory as in Assumption D; the finite constants C_0, C_{out} in Eq. equation 35 bound factor growth from the residual and the resulting reusable output, respectively.

B Proof conventions

The first-passage proofs use Assumptions D–F; the built-block proof uses its separately stated architecture and initialization. A route index π is suppressed when only one arm is considered. All constants below may depend on the fixed data and dictionaries, initialization weights c_a, λ, κ, sector coefficients θ, g, ρ, exit levels, and strict positive-rate, winner–loser and threshold margins, but not on α. Distinct positive sector rates need not be separated by a prescribed gap.

C EXACT REDUCTION

Lemma 4 (Balance). $\frac{d}{dt}(u_a^2 - v_a^2) = -2\lambda(u_a^2 - v_a^2)$*; hence Assumption D preserves* $u_a = v_a > 0$ *and* $z_a = u_a^2$.

Proof. The loss terms cancel in $2u_a\dot{u}_a - 2v_a\dot{v}_a$, as in Du et al. (2018, Theorem 2.1); decay gives the exponential identity below. Uniqueness prevents an initially positive balanced pair from reaching the zero pair in finite time, so

$$u_a(t)^2 - v_a(t)^2 = e^{-2\lambda t}[u_a(0)^2 - v_a(0)^2] = 0, \qquad z_a(t) = u_a(t)^2 > 0, \tag{36}$$

which is the balance claim under Assumption D. □

Lemma 5 (Sector decoupling). *Set* $m = \sqrt{\kappa N}h$, $m_j = \langle m, \Phi_j\rangle_N$, $U_j = \sum_{a\in I_j} g_{ja}z_a$, *and* $E_j = \theta_j - U_j - m_j$*; then*

$$\dot{z}_a = 2z_a(g_{ja}E_j - \lambda), \qquad \dot{m}_j = \kappa E_j - \lambda m_j, \tag{37}$$

and

$$Y_*^2\mathcal{E}_\pi = \|y_\perp^*\|_*^2 + \sum_{j:\rho_j>0} \rho_j^2(\theta_j - U_j)^2. \tag{38}$$

Proof. By Lemma 4, $z_a = u_a^2$ and $\dot{z}_a = 2u_a\dot{u}_a$. The routed-feature reduction follows Saxe et al. (2022); the private-memory coupling and compatible held-out identity are derived here. Writing $e_i = y_i - \widehat{y}_i$, differentiation of the averaged loss gives

$$\dot{u}_a = v_a\langle D_a, e\rangle_N - \lambda u_a, \quad \dot{v}_a = u_a\langle D_a, e\rangle_N - \lambda v_a, \quad \dot{h}_i = \sqrt{\kappa/N}\, e_i - \lambda h_i. \tag{39}$$

The memory scaling therefore gives $\dot{m} = \kappa e - \lambda m$, while balance gives $\dot{z}_a = 2z_a(\langle D_a, e\rangle_N - \lambda)$. Let $y_\perp$ and $m_\perp$ be the training target and memory projections orthogonal to all sector directions. Training orthogonality identifies each path's residual correlation exactly:

$$e = \sum_j E_j\Phi_j + (y_\perp - m_\perp), \qquad \langle D_a, e\rangle_N = g_{ja}E_j \quad (a \in I_j). \tag{40}$$

Projection of the memory equation gives equation 37 and $\dot{m}_\perp = \kappa(y_\perp - m_\perp) - \lambda m_\perp$. There is no private coordinate on held-out examples, and compatibility gives

$$y - \widehat{y} = y_\perp^* + \sum_{j:\rho_j>0} \rho_j(\theta_j - U_j)\Phi_j^*. \tag{41}$$

Positive decay bounds every factor and private coordinate on the initial objective sublevel set, so the flow exists globally. Orthogonality removes every cross term, giving

$$\|y - \widehat{y}\|_*^2 = \|y_\perp^*\|_*^2 + \sum_{j:\rho_j>0} \rho_j^2(\theta_j - U_j)^2, \tag{42}$$

which is equation 38; the projected gradient equations above are equation 37. □

D EXIT AND SETTLING

Fix one positive-target sector and write $g = g_*$, $Z = \sum_{g_a=g} z_a$, $C_* = \sum_{g_a=g} c_a$, $r = fg\theta - \lambda$, and $B = g + \sum_{g_a<g} |g_a|c_a/C_*$.

Lemma 6 (Exit time). *Suppose* $r > 0$*, Assumption D, and* $0 < \zeta \le r/[4gB(1+\kappa/\Delta)]$*. For sufficiently small* α*,* $E \ge 0$ *throughout, and the first exit* $Z(\tau_\zeta) = \zeta$ *satisfies*

$$\left|\tau_\zeta - \frac{\log(\alpha^{-2}) + \log(\zeta/C_*)}{2r}\right| \le \frac{g}{r}\left[\frac{(1-f)\theta}{\Delta} + \frac{B\zeta}{r}(1+\kappa/\Delta)\right]. \tag{43}$$

At exit, $z_a/Z \to 0$ *for every strict loser* $g_a < g$*, and memory lies in a fixed compact interval.*

Proof. The logarithmic escape scale is standard (Saxe et al., 2014; Li et al., 2021; Jacot et al., 2022); we bound the moving-memory correction explicitly. Take $C_*\alpha^2 < \zeta$ and $\alpha^2 \sum_a |g_a| c_a < \theta/2$, which make $E(0) > 0$; while $E \geq 0$, exact integration gives

$$\frac{z_a(t)}{Z(t)} = \frac{c_a}{C_*} e^{-2(g-g_a)\int_0^t E}, \quad \sum_{g_a<0} |g_a| z_a(t) \leq N_0 := \alpha^2 \sum_{g_a<0} |g_a| c_a. \tag{44}$$

Negative-gain paths decrease in this region, while all strict-loser ratios are nonincreasing, yielding $|U| \leq BZ$ and $E \leq \theta + N_0 - m$. The memory boundary points inward at $m = 0$, and scalar comparison yields

$$0 \leq m(t) \leq \frac{\kappa(\theta + N_0)}{\Delta} < \frac{3\theta}{2}. \tag{45}$$

At a hypothetical first zero of E, the exact equations give

$$\dot{E} = 2\lambda U + \lambda m = \lambda(2\theta - m) > 0, \tag{46}$$

which excludes a crossing and makes these bounds valid for all time. With $m^0 = (1-f)\theta(1-e^{-\Delta t})$, the exact transient decomposition is

$$E = f\theta + (1-f)\theta e^{-\Delta t} - U + H, \qquad H = \kappa \int_0^t e^{-\Delta(t-s)} U(s)\, ds. \tag{47}$$

Before exit, $|U| \leq B\zeta$ and $|H| \leq \kappa B\zeta/\Delta$, so the chosen exit level gives $(\log Z)' \geq r$ and forces a finite exit. Integrating $\dot{Z} \geq rZ$ and then exchanging the order in the convolution integral gives

$$\int_0^{\tau_\zeta} Z \leq \zeta/r, \qquad \int_0^{\tau_\zeta} |U| \leq B\zeta/r, \qquad \int_0^{\tau_\zeta} |H| \leq \kappa B\zeta/(\Delta r). \tag{48}$$

The exact winner logarithm is

$$\log \frac{\zeta}{C_*\alpha^2} - 2r\tau_\zeta = 2g \left[\frac{(1-f)\theta(1-e^{-\Delta\tau_\zeta})}{\Delta} - \int_0^{\tau_\zeta} U + \int_0^{\tau_\zeta} H \right]. \tag{49}$$

The preceding integral bounds give exactly equation 43, without assuming that H is nonnegative. Moreover,

$$\int_0^{\tau_\zeta} E = \frac{\log[\zeta/(C_*\alpha^2)] + 2\lambda\tau_\zeta}{2g} \longrightarrow \infty, \qquad \frac{z_a(\tau_\zeta)}{Z(\tau_\zeta)} \longrightarrow 0 \quad (g_a < g), \tag{50}$$

which is the remaining exit claim; the memory bound above supplies its compact interval. □

Lemma 7 (Settling, including the boundary)**.** *Under the initialization and fixed exit-level hypotheses of Lemma 6, if $g\theta > \Delta$, set $Z_\infty = (\theta - \Delta/g)/g$ and $m_\infty = \kappa/g$. For every $\varepsilon > 0$, a time $T_s(\varepsilon)$ independent of sufficiently small α makes $|U(t) - gZ_\infty| < \varepsilon$ for all $t \geq \tau_\zeta + T_s$. For the boundary case, allow any $\theta \geq 0$ and gains with $f\theta \max_a g_a \leq \lambda$; then the sector converges to $z = 0$, $m = \kappa\theta/\Delta$.*

Proof. For the positive equilibrium, put $v = E - \lambda/g$, $w = m - m_\infty$, and $d_a = \lambda(1 - g_a/g) > 0$ for every strict loser. The reduced equations and residual identity become

$$\dot{Z} = 2gZv, \quad \dot{z}_a = 2z_a(g_a v - d_a), \quad \dot{w} = \kappa v - \lambda w, \quad g(Z - Z_\infty) + \sum_{g_a<g} g_a z_a + w = -v. \tag{51}$$

Define the nonnegative entropy certificate

$$V = \tfrac{1}{2}[Z - Z_\infty - Z_\infty \log(Z/Z_\infty) + \sum_{g_a<g} z_a] + \frac{w^2}{2\kappa}. \tag{52}$$

Differentiation and the residual identity cancel the mixed terms exactly:

$$\dot{V} = g(Z - Z_\infty)v + \sum_{g_a<g} z_a(g_a v - d_a) + w(v - \lambda w/\kappa) = -D, \quad D := v^2 + \frac{\lambda}{\kappa} w^2 + \sum_{g_a<g} d_a z_a. \tag{53}$$

At exit, $Z = \zeta$, $0 \le z_a \le (c_a/C_*)\zeta$, and $0 \le m \le 3\kappa\theta/(2\Delta)$, so $V(\tau_\zeta) \le \overline{V}$ for a fixed finite $\overline{V}$. Because the entropy term diverges as $Z \downarrow 0$ or $Z \uparrow \infty$, this sublevel set lies in $a \le Z \le b$ with $a > 0$ and has bounded loser coordinates and memory, independently of α. Enlarging $[a, b]$ to include Z_∞, elementary integration gives

$$\frac{(Z - Z_\infty)^2}{2b} \le Z - Z_\infty - Z_\infty \log(Z/Z_\infty) \le \frac{(Z - Z_\infty)^2}{2a}. \tag{54}$$

Weighted Cauchy–Schwarz applied to the residual identity gives

$$g^2(Z - Z_\infty)^2 \le D\left(1 + \kappa/\lambda + \sum_{g_a<g} g_a^2 z_a/d_a\right). \tag{55}$$

For strict losers define $Q_- := \max_{g_a<g} g_a^2/d_a$ and $R_- := \max_{g_a<g} 1/d_a$, taking both maxima as zero when there are no losers. Since $\sum_{g_a<g} z_a \le 2\overline{V}$, the last coefficient in equation 55 is at most $2\overline{V}Q_-$, and

$$\frac{1}{2}\sum_{g_a<g} z_a \le \frac{R_-}{2}D, \qquad \begin{aligned} &\frac{w^2}{2\kappa} \le \frac{D}{2\lambda}, \qquad V \le CD, \\ &C := \frac{1 + \kappa/\lambda + 2\overline{V}Q_-}{4ag^2} + \frac{R_-}{2} + \frac{1}{2\lambda}. \end{aligned} \tag{56}$$

All quantities in C are fixed independently of α; it may diverge as a strict loser gain approaches g. The inequality $\dot{V} \le -V/C$ gives $V(\tau_\zeta + t) \le \overline{V}e^{-t/C}$. The entropy lower bound gives $|Z - Z_\infty| \le 2\sqrt{bV}$, while $\sum_{g_a<g} z_a \le 2V$. With $G_- = \max_{g_a<g} |g_a|$ (zero if there are no losers), this yields

$$|U(\tau_\zeta + t) - gZ_\infty| \le 2g\sqrt{b\overline{V}}\, e^{-t/(2C)} + 2G_-\overline{V}e^{-t/C}. \tag{57}$$

Choosing $T_s(\varepsilon)$ so that this right side is below ε proves the positive-equilibrium claim uniformly in α. For the boundary case, use the compact sublevels of

$$\Psi = \tfrac{1}{2}E^2 + \lambda\sum_a z_a + \lambda m^2/(2\kappa), \qquad \dot{\Psi} = -2\sum_a z_a(g_aE - \lambda)^2 - \kappa(E - \lambda m/\kappa)^2. \tag{58}$$

The derivative vanishes only at stationary points, because a zero path coordinate already has zero velocity. Any positive coordinates at a stationary point share a nonzero gain b, and $bE = \lambda$ and $\kappa E = \lambda m$ give

$$b^2 \sum_{a:z_a>0} z_a = b\theta - \Delta \le 0. \tag{59}$$

For $b > 0$ the final inequality follows from $b \le g$ and $g\theta \le \Delta$, while for $b < 0$ it is strict, so positive stationary coordinates are impossible. LaSalle's principle on the compact sublevel set (LaSalle, 1960, Theorem 1) therefore gives

$$z_a(t) \longrightarrow 0 \quad \text{for every path}, \qquad m(t) \longrightarrow \kappa\theta/\Delta, \tag{60}$$

which is the boundary claim. □

Lemma 8 (Uniform sector bounds)**.** *For a needed sector under Assumptions D–E, write $g = g_* > 0$, $r = fg\theta - \lambda$, $P_+ = \sum_{g_a>0} g_a z_a$, $n = \sum_{g_a<0} |g_a| z_a$, $n_0 = n(0)$, and $U = P_+ - n$. For sufficiently small α, uniformly over $t \ge 0$,*

$$0 \le \theta - U(t) \le \theta + n_0, \qquad \begin{cases} \theta - U(t) \ge \Delta/g - n_0/f, & r > 0, \\ |U(t)| = O(\alpha^2), & r \le 0. \end{cases} \tag{61}$$

Proof. The invariant-region proof in Lemma 6 does not use $r > 0$; it gives $E, m \ge 0$, $n(t) \le n_0 e^{-2\lambda t}$, and $m \le \kappa(\theta + n_0)/\Delta$. Thus $\theta - U = E + m \ge 0$, $U \ge -n_0$, and, because every positive gain is at most g,

$$\dot{P}_+ \le 2P_+(gE - \lambda). \tag{62}$$

If $r > 0$, put $u_\infty = \theta - \Delta/g > 0$, $m_\infty = \kappa/g$, and $\varepsilon_\alpha = n_0/f$. Choose fixed t_0 with $(\kappa\theta/\Delta)(1 - e^{-\Delta t_0}) > m_\infty$; fixed-time factor bounds give $P_+ = O(\alpha^2)$ on $[0, t_0]$ and $m(t_0) > m_\infty$ for

sufficiently small α. After t_0, the region $P_+ \le u_\infty + \varepsilon_\alpha$, $m \ge m_\infty - \kappa\varepsilon_\alpha/\Delta$ is invariant, since its two boundary derivatives satisfy

$$E \le \lambda/g - f\varepsilon_\alpha + n_0 = \lambda/g \implies \dot{P}_+ \le 0, \qquad \dot{m} \ge \kappa(\theta - u_\infty - \varepsilon_\alpha) - \Delta(m_\infty - \kappa\varepsilon_\alpha/\Delta) = 0. \tag{63}$$

The fixed-time bound also gives $P_+ < u_\infty$ before t_0, so $U \le u_\infty + n_0/f$ for all time. If $r \le 0$, set $V_+(t) = \max_{s \le t} P_+(s) > 0$ and use equation 47; at a running maximum, $H \le \kappa V_+/\Delta$, while otherwise the running maximum does not increase. The upper derivative and its integrated bound are

$$D^+ \log V_+ \le 2g[(1-f)\theta e^{-\Delta t} + n_0 e^{-2\lambda t}], \qquad \sup_t P_+(t) \le P_+(0) e^{2g(1-f)\theta/\Delta + g n_0/\lambda} = O(\alpha^2). \tag{64}$$

Combining this with $n \le n_0 = O(\alpha^2)$ proves equation 61, including $r = 0$. □

E Proof of Theorem 1

Proof. By Lemmas 4 and 5, the flow decouples into the sector systems equation 37 with held-out risk given by the exact identity equation 38; we apply Lemmas 6, 7 and 8 sector by sector. Suppress π and write $h = h_\pi$, $s = s_{\pi,h} > 0$, and $\ell_\alpha = \log(\alpha^{-2})$. Until the zero-target estimate below, every sector index ranges over J^+. For the lower bound choose fixed admissible exit levels in every positive-rate sector outside $\mathcal{F}_{h-1}$, small enough that

$$P + \sum_{j \in \mathcal{F}_{h-1}} \ell_j + \sum_{\substack{j \notin \mathcal{F}_{h-1} \\ r_j > 0}} \frac{\rho_j^2(\theta_j - B_j\zeta_j)^2}{Y_*^2} + \sum_{\substack{j \in J^+ \\ r_j \le 0}} w_j > \delta, \qquad B_j\zeta_j < \theta_j. \tag{65}$$

The strict inequality follows from $\delta < \mathcal{E}^{[h-1]}$. Let K_j be the right side of equation 43 in Lemma 6 for sector j, plus $|\log(\zeta_j/C_{j*})|/(2r_j)$. Thus $|\tau_j - \ell_\alpha/(2r_j)| \le K_j$. Outside $\mathcal{F}_{h-1}$, every positive rate satisfies $r_j \le s$, so $C_- := 1 + \max_{j \notin \mathcal{F}_{h-1},\, r_j > 0} K_j$ gives

$$t \le \ell_\alpha/(2s) - C_- \quad \implies \quad Z_j(t) \le \zeta_j \quad (j \notin \mathcal{F}_{h-1},\, r_j > 0) \tag{66}$$

for all sufficiently small α. Lemma 8 lower-bounds the faster sectors by their equilibrium errors minus $O(\alpha^2)$ at every earlier time, and keeps the nonpositive-rate outputs uniformly $O(\alpha^2)$. The risk identity equation 38 and equation 65 therefore exclude every crossing before $\ell_\alpha/(2s) - C_-$, even during faster-sector transients.

For the upper bound, choose fixed settling tolerances using

$$P + \sum_{j \in \mathcal{F}_h} \frac{\rho_j^2(\Delta/g_{j*} + \varepsilon_j)^2}{Y_*^2} + \sum_{j \notin \mathcal{F}_h} w_j < \delta. \tag{67}$$

Choose admissible exit levels for $j \in \mathcal{F}_h$ and define their K_j in the same way. Since $r_j \ge s$ on this set, Lemmas 6 and 7 put and keep each such sector within tolerance after $\ell_\alpha/(2s) + C_+$, where $C_+ := 1 + \max_{j \in \mathcal{F}_h}[K_j + T_{s,j}(\varepsilon_j)]$. Neither clock comparison requires a gap between distinct rates. Every remaining needed sector contributes at most $w_j + O(\alpha^2)$ by the first bound in equation 61, whether or not it has begun learning. For a zero-target sector, its nonincreasing objective gives

$$\lambda \sum_a z_a(t) \le \Psi(0) = \lambda\alpha^2 \sum_a c_a + \frac{\alpha^4}{2}\left(\sum_a g_a c_a\right)^2, \tag{68}$$

so its output is uniformly $O(\alpha^2)$ and its held-out error is $O(\alpha^4)$; sectors with $\rho_j = 0$ contribute nothing. These bounds and equation 67 put the risk below threshold for all times after the stated upper bound. Consequently there are $C_\pi < \infty$ and $\bar{\alpha}_\pi > 0$, with the fixed-quantity dependencies in Appendix B, such that

$$\left| T_\pi(\delta) - \frac{\log(\alpha^{-2})}{2r_{\pi,\delta}} \right| \le C_\pi \quad (0 < \alpha < \bar{\alpha}_\pi), \tag{69}$$

which is equation 6.

For the training claim, nonincreasing loss gives $\|e(t)\|_N \le M_0 := \|y\|_N + 1$ for sufficiently small α. For fixed T, the factor equation then gives the uniform circuit bound

$$\sup_{t\le T} \|\widehat{y}_{\mathrm{circ}}(t)\|_N \le K_T \alpha^2, \qquad K_T = \sum_a \|D_a\|_N c_a e^{2\|D_a\|_N M_0 T}. \tag{70}$$

Solving the linear memory equation and using this bound yields

$$\big\|e(T) - [f + (1-f)e^{-\Delta T}]y\big\|_N \le (1 + \kappa/\Delta) K_T \alpha^2. \tag{71}$$

Given $\eta > f^2\|y\|_N^2$, choose a fixed t_η with $b_\eta := [f + (1-f)e^{-\Delta t_\eta}]\|y\|_N < \sqrt{\eta}$. For sufficiently small α,

$$\|e(t_\eta)\|_N^2 \le [b_\eta + (1 + \kappa/\Delta) K_{t_\eta} \alpha^2]^2 < \eta, \tag{72}$$

which is the bounded training-fit-time claim. Combining equation 72 with the lower passage bound, it suffices to take $\alpha < \bar{\alpha}_\pi$ and

$$\alpha^2 < \min\left\{ \frac{\sqrt{\eta} - b_\eta}{(1 + \kappa/\Delta) K_{t_\eta}}, \quad e^{-2r_{\pi,\delta}(t_\eta + C_\pi + 1)} \right\}. \tag{73}$$

Then $T_\eta^{\mathrm{tr}} \le t_\eta < T_\pi(\delta)$, which is equation 7. Finally, for $r_{A,\delta} > r_{B,\delta}$ put $d_{AB} = 1/(2r_{B,\delta}) - 1/(2r_{A,\delta}) > 0$ and $\alpha_0 = \min\{\bar{\alpha}_A, \bar{\alpha}_B, e^{-(C_A + C_B + 1)/(2d_{AB})}\}$. For $0 < \alpha < \alpha_0$,

$$T_B - T_A \ge d_{AB} \log(\alpha^{-2}) - (C_A + C_B) > 0, \tag{74}$$

which is the stated strict route ordering. □

F Proof of the lookup proposition

Proof of Proposition 2. For each nonempty routed row subset $A_{k,r}$ of mass $m_{k,r}$, define, with $\mathrm{sgn}(0) = 1$,

$$\Phi_{k,r,j} = \mathrm{sgn}(\widehat{g}_r(j)) \mathbf{1}_{A_{k,r}} \phi_j(x)/\sqrt{m_{k,r}}, \quad \theta_{k,r,j} = \beta_{rj}\sqrt{m_{k,r}}, \quad g_{k,r,j,\pm} = \pm\,\mathrm{sgn}(\widehat{g}_r(j))\sqrt{m_{k,r}} \tag{75}$$

where the $\Phi_{k,r,j}$ are orthonormal: different routed subsets have disjoint supports, and realized operand balance preserves the dictionary inner products within each subset. The conditional product distribution also gives $\mathbb{E}[\phi_j(x)\phi_l(x') \mid A_{k,r}, c] = 0$, eliminating every row–column cross term. Columns have the same construction with $m_c = n_c/|\mathcal{T}|$, and hence their signed route columns form rank-one Gram blocks as required by Assumption E, so Lemma 5 applies and each (k, r, j) is one sector. The product $g_*\theta = m_{k,r}\beta_{rj}$ now yields equation 2. Routing held-out row r to $\kappa(r)$ gives

$$\rho^2_{\kappa(r),r,j} = m^*_r/m_{\kappa(r),r}, \quad \rho\theta = \beta_{rj}\sqrt{m^*_r}, \qquad \rho_{k,r,j} = 0 \quad (k \ne \kappa(r)). \tag{76}$$

Column ratios are $\rho^2_{c,j} = m^*_c/m_c$, and target-spanning dictionaries give $y^*_\perp = 0$. For any partition, $0 < m_{\kappa(r),r} \le m_r = n_r/|\mathcal{T}|$, so each aligned row rate is no smaller. For a needed row mode ($m^*_r > 0$, $\beta_{rj} > 0$), the common held-out energy and mass-dependent equilibrium contribution are

$$w_{rj} = \frac{m^*_r \beta^2_{rj}}{Y^2_*}, \qquad \ell_{rj}(m) = w_{rj} \min\{1, (\Delta/(m\beta_{rj}))^2\}. \tag{77}$$

The contribution is nonincreasing in m, while columns are unchanged. Index each needed held-out row mode by its selected trained copy, and include the needed column modes unchanged. At any positive rate cutoff q, aligned routing either settles the same mode with a no-larger floor, or settles a mode still absent under π; in the latter case $\ell_{A,j} \le w_j$. Modes absent under both arms contribute the same w_j. Summing over these matched needed modes gives

$$\mathcal{B}_A(q) \le \mathcal{B}_\pi(q), \qquad \mathcal{B}_\pi(q) := \sum_{\substack{j \in J^+_\pi \\ r_{\pi,j} \ge q}} \ell_{\pi,j} + \sum_{\substack{j \in J^+_\pi \\ r_{\pi,j} < q}} w_{\pi,j}. \tag{78}$$

Taking $q = r_{\pi,\delta}$ makes $\mathcal{B}_\pi(q) < \delta$, so the aligned critical rate is at least q; if the inequality of critical rates is strict, equation 74 gives strict first-passage ordering. For the three special arms, the masses are $n_r/|\mathcal{T}|$, $1/|\mathcal{T}|$, and $n_r/(K|\mathcal{T}|)$. Under equation 33, $1 < n_r < K$ gives strict row-bottleneck

ordering; the common-column condition extends it to the arm bottlenecks, which then equal their threshold-dependent rates. Under the proposition's hypotheses, equation 74 therefore gives

$$T_{\mathrm{scr}} - T_A > 0, \qquad T_{\mathrm{frag}} - T_{\mathrm{scr}} > 0 \quad (0 < \alpha < \min\{\alpha_{0,A,\mathrm{scr}}, \alpha_{0,\mathrm{scr,frag}}\}), \tag{79}$$

which is equation 8. □

Remark 1 (Noncrossing when the threshold margin fails). Under Assumption D and the design of Appendix A.4, suppose that instead of Assumption F an arm π admits a set S_0 of held-out-active row modes with

$$m_{\kappa(r),r}\beta_{rj} \le \Delta \quad ((r,j) \in S_0), \qquad \sum_{(r,j)\in S_0} m_r^* \beta_{rj}^2 > \delta Y_*^2. \tag{80}$$

Then $T_\pi(\delta) = \infty$ for all sufficiently small α: the arm never reaches held-out error δ.

Proof. The first condition in equation 80 says $fg\theta \le \lambda$ for every sector in S_0, so these sectors have nonpositive rate. In such a sector, relabel the positive- and negative-gain paths as z_+ and z_-, and write $U = gz_+$, $n = gz_-$, and $V(t) = \max_{s\le t} U(s)$. The invariant-region argument of Lemma 6 does not require positive r, so $n(t) \le n_0 e^{-2\lambda t}$ and memory remains bounded. At a running maximum, the convolution of the signed output satisfies $H \le (\kappa/\Delta)V$, and therefore

$$E \le f\theta + (1-f)\theta e^{-\Delta t} - fV + n. \tag{81}$$

Using $fg\theta \le \lambda$ in the winner equation and integrating the resulting nonnegative envelope gives

$$D^+ \log V \le 2g[(1-f)\theta e^{-\Delta t} + n_0 e^{-2\lambda t}], \qquad \sup_t U(t) \le U(0) e^{2g(1-f)\theta/\Delta + g n_0/\lambda} = O(\alpha^2). \tag{82}$$

Both signed path outputs are $O(\alpha^2)$ uniformly in time, so the held-out energy of the S_0 modes is never recovered, and the second condition in equation 80 gives

$$\inf_{t\ge 0} \mathcal{E}_\pi(t) \ge \frac{\sum_{(r,j)\in S_0} m_r^* \beta_{rj}^2}{Y_*^2} - C\alpha^2 > \delta, \qquad T_\pi(\delta) = \infty, \tag{83}$$

for sufficiently small α. □

G Proof of the single-block comparison

G.1 Notation for the built-block instance

For the instance in Theorem 3(ii), write $B_{\phi,N} = \max_i \|\phi(x_i)\|_1$ and $B_{\phi,*} = \sup_* \|\phi(x)\|_1$ for the fixed feature bounds in equation 35. Let V_0 be the held-out column-output span and let $P_{V_0^\perp}$ denote orthogonal projection onto its complement. The factorial design gives $y_{\mathrm{col}} \in V_0$ and $y_{\mathrm{row}} \perp V_0$; private memory is zero on held-out examples. Thus the proof abbreviates

$$Q := \frac{\|P_{V_0^\perp} y\|_*^2}{Y_*^2} = Q_{\mathrm{row}}. \tag{84}$$

The constant row-bypass control is outside this instance.

Remark 2 (Supplied block below the rate floor). Theorem 3(i) assumes the threshold margin for the supplied arm. Without it, if some $j \in J_A^+$ has $r_j(\gamma) \le 0$ and $\rho_j^2\theta_j^2 > \delta Y_*^2$, then $\mathcal{E}_{\mathsf{C}}(t) > \delta$ for all large t. This is an eventual-error statement, not a claim that the arm never crosses; it shows that the gain can push a needed sector below its rate floor rather than merely slow it.

Proof of Theorem 3. (i). Scaling every column by γ scales the Gram matrix by γ^2 and both route gains by γ, leaving θ_j, ρ_j unchanged; Theorem 1 yields equation 9, with

$$r_j(\gamma) = \gamma(r_j(1) + \lambda) - \lambda, \qquad T_{\mathsf{C}} = \frac{\log(\alpha^{-2})}{2r_{\mathsf{C},\delta}} + O(1). \tag{85}$$

The order and ties of sector rates are unchanged, and the supplied positive-rate groups form a prefix of the aligned groups. For every stage h in that prefix, $\mathcal{E}_{\mathsf{C}}^{[h]} \geq \mathcal{E}_{A}^{[h]}$, because supplied equilibrium errors are no smaller and absent energies are identical. Thus $h_A \leq h_{\mathsf{C}}$, and

$$r_{A,\delta} \geq s_{A,h_{\mathsf{C}}}, \qquad r_{\mathsf{C},\delta} = \gamma(s_{A,h_{\mathsf{C}}} + \lambda) - \lambda < s_{A,h_{\mathsf{C}}} \leq r_{A,\delta} \quad (\gamma < 1). \tag{86}$$

The constants from equation 69 give

$$T_{\mathsf{C}} - T_A \geq \log(\alpha^{-2}) \left(\frac{1}{2r_{\mathsf{C},\delta}} - \frac{1}{2r_{A,\delta}} \right) - (C_A + C_{\mathsf{C}}) > 0 \quad (0 < \alpha < \alpha_{0,A,\mathsf{C}}), \tag{87}$$

where $\alpha_{0,A,\mathsf{C}}$ is defined as in equation 74. At $\gamma = 1$, the same columns, factor initializations, and memory give identical differential equations and hence identical flows. This proves (i). For Remark 2: if $r_j(\gamma) \leq 0$, Lemma 7 instead gives

$$U_j(t) \to 0, \qquad \liminf_{t\to\infty} \mathcal{E}_{\mathsf{C}}(t) \geq \rho_j^2 \theta_j^2 / Y_*^2; \tag{88}$$

when $\rho_j^2 \theta_j^2 > \delta Y_*^2$, this gives $\mathcal{E}_{\mathsf{C}}(t) > \delta$ for all large t, as claimed.

(ii). Set $M = \max_n\{|o_n|, \|w_n\|_\infty, \|q_n\|_\infty\}$. The built entries are initially at most α, so their degree-three prediction is at most $\gamma n_u B_{\phi,N} \alpha^3$ in magnitude. Every retained two-factor branch has $O(\alpha^2)$ initial output by Assumption D, private memory starts at zero, and the initial decay penalty is $O(\alpha^2)$; therefore $\mathcal{J}(0) = \|y\|_N^2/2 + O(\alpha^2)$; nonincreasing regularized loss then bounds the residual by $\sqrt{2\mathcal{J}(0)} \leq \|y\|_N + 1$ for sufficiently small α.

One-hot e_c and $|q_n^\top \phi| \leq M\|\phi\|_1$ bound each coordinate derivative of a unit's prediction by $\gamma B_{\phi,N} M^2$, so Cauchy–Schwarz bounds its loss derivative by $C_0 M^2$, where $C_0 = \gamma B_{\phi,N}(\|y\|_N + 1)$. At a coordinate attaining M, decay contributes $-\lambda M$ to its absolute-value derivative; taking the upper right Dini derivative $D^+ M$ and applying scalar comparison, for $C_0 > 0$, gives

$$D^+ M \leq C_0 M^2 - \lambda M, \qquad M(t) \leq \frac{\alpha}{1 - C_0 \alpha t} \leq 2\alpha \quad \left(t \leq \frac{1}{2C_0\alpha} \right). \tag{89}$$

Write the held-out prediction as $v + u$ with $v \in V_0$ and degree-three output u, whose norm is at most $8\gamma n_u B_{\phi,*} \alpha^3$. Orthogonal projection removes v and the reverse triangle inequality gives $\|y - v - u\|_* \geq \|P_{V_0^\perp} y\|_* - \|u\|_*$; choose α so that $\|u\|_*/Y_* < \sqrt{Q} - \sqrt{\delta}$, making the right side positive before squaring:

$$\mathcal{E}_{\mathsf{C}}(t) \geq \left(\sqrt{Q} - \frac{8\gamma n_u B_{\phi,*} \alpha^3}{Y_*} \right)^2 > \delta \tag{90}$$

for sufficiently small α on that interval, proving $T_{\mathsf{C}} \geq 1/(2C_0\alpha)$. For permanent noncrossing, put $b_* = \gamma n_u B_{\phi,*} \alpha^3 / Y_*$ and choose $b_* < \sqrt{Q} - \sqrt{\delta}$. The two cases conclude separately:

$$C_0 = 0: \quad M(t) \leq \alpha e^{-\lambda t}, \quad \inf_{t\geq 0} \mathcal{E}_{\mathsf{C}}(t) \geq (\sqrt{Q} - b_*)^2 > \delta, \quad T_{\mathsf{C}} = \infty; \tag{91}$$

$$C_0 > 0, \alpha < \lambda/C_0: \quad M(t) \leq \alpha e^{-(\lambda - C_0\alpha)t}, \quad \inf_{t\geq 0} \mathcal{E}_{\mathsf{C}}(t) \geq (\sqrt{Q} - b_*)^2 > \delta, \quad T_{\mathsf{C}} = \infty. \tag{92}$$

These are the two noncrossing cases in (ii). If the aligned arm also satisfies Assumptions D–F, then, for $C_0 > 0$,

$$T_{\mathsf{C}} - T_A \geq \frac{1}{2C_0\alpha} - \frac{\log(\alpha^{-2})}{2r_{A,\delta}} - C_A > 0 \quad \text{for all sufficiently small } \alpha. \tag{93}$$

For $C_0 = 0$ the built arm has $T_{\mathsf{C}} = \infty$ while Theorem 1 gives $T_A < \infty$; these two cases prove the stated comparison. □

H Additional finite-sample checks

H.1 Validation and further checks

Figure 5 checks Proposition 2 in its memorization-first regime, with $\alpha = 10^{-9}$ first and 10^{-6} second. We use its 5×4 mask, $p = 7$, $K = 8$, $\kappa = .006$, $\lambda = .0005$, and a fixed complete real

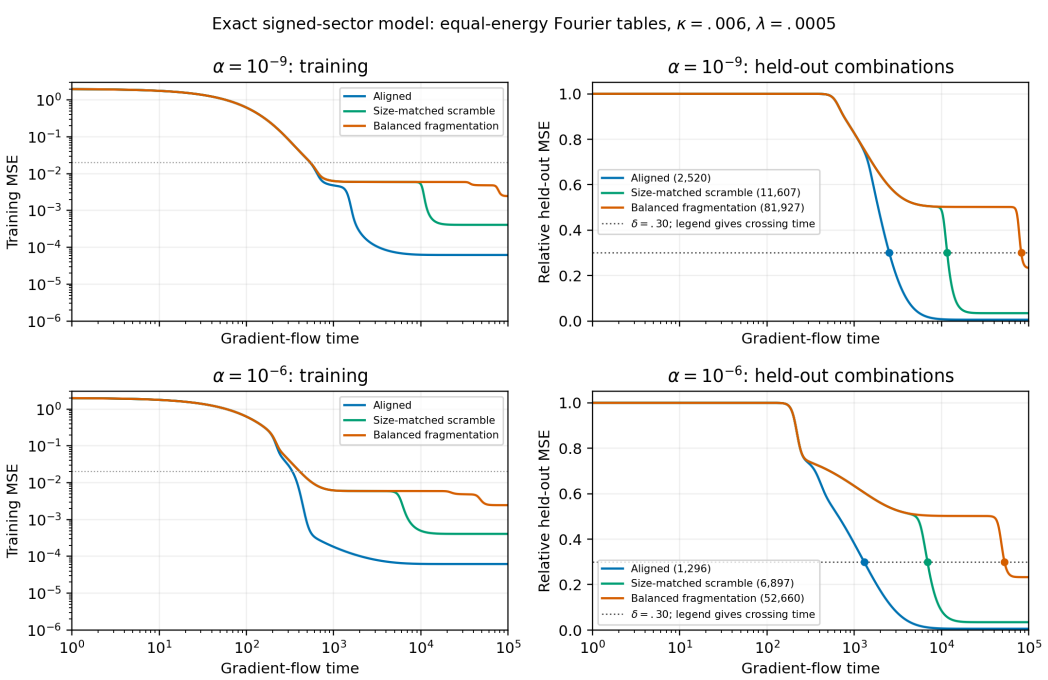

Figure 5: Exact signed-sector gradient flow on the disclosed equal-energy Fourier instance. Left: training MSE, with fit level .02. Right: relative held-out MSE, with first-passage level $\delta = .30$. Training fit precedes every held-out crossing for both $\alpha = 10^{-9}$ (top) and 10^{-6} (bottom). Times are continuous gradient-flow units, not Transformer updates.

Fourier dictionary. Each rule has six Fourier coefficients of magnitude $1/\sqrt{6}$ with independent seed-0 signs. The theorem check uses this designed equal-energy Fourier target distribution. These values are simulation choices, not quantities derived from the Transformer: $\kappa > \lambda$ makes the residual fraction $f \approx .077$ small; at the displayed initializations, the measured crossings verify memorization before held-out generalization; the equal-energy coefficients give $r_{\pi,*} > 0$; and $\delta = .30$ satisfies both equation 33 and the stage margin for all three routes, whose floors are reported below. No hypothesis of Theorem 1 or Proposition 2 depends on these numbers. Gaussian-table controls can have nonpositive globally weakest modes and fail equation 33; the threshold-dependent theorem retains those modes as absent energy rather than discarding them. Their coverage is determined by the stage margin, not by positivity of every mode.

Every observed operand pair is replicated eight times with distinct private identifiers, giving $N = 6272$ examples and exact realized balance in every routed cell subset. Held-out row r uses $\kappa(r)$, its largest training copy, with ties resolved by block index. All three routes retain 1056 circuit factors and 6272 private coordinates. We integrate the exact invariant reduction of Eq. 2 with separate positive and negative balanced pairs: $z = u_+v_+ - u_-v_-$ and $u_\pm = v_\pm > 0$. Log-product integration preserves this equality; inactive factors decay analytically. The printed sector checks give Gram and held-out compatibility errors below 7×10^{-15}.

An independent implementation constructs the routed dictionary from raw features and masks, and evolves every active factor and all 6272 private coordinates through $t = 2500$. It does not use projected memory or sector residuals. Factors on columns verified zero on both supports decay analytically. At $\alpha = 10^{-3}$, its aligned/scrambled first passages are 113.090293747 and 2248.633215617. Their discrepancies from independently integrated reduced trajectories are 9.30×10^{-9} and 1.08×10^{-6}. Across all examples at 2503 sampled times per arm, the largest training/held-out prediction differences are $3.01 \times 10^{-6}/2.88 \times 10^{-6}$. These are sampled maxima on the unit-spaced grid plus both event times. A tenfold tighter tolerance reduces the scrambled crossing discrepancy to 3.88×10^{-8}. Against the original reduced runs underlying Table 2, which use looser tolerances, the maximum crossing-time difference is 4.68×10^{-5}.

Seed	α	Aligned	Scrambled	Fragmented
0	10^{-6}	774.56	3944.33	18253.37
0	10^{-9}	1908.48	6495.77	28225.32
1	10^{-6}	646.73	3617.70	17896.85
1	10^{-9}	1734.05	6185.89	28851.73

Table 3: Gaussian-table relative-MSE first passages at $\delta = .30$, in continuous gradient-flow time. All modes, including nonpositive-rate modes, are retained.

The aligned, scrambled and fragmented bottleneck rates are .00538820, .00146273, and .000236025; their relative risk floors are .005633, .034476, and .232769. Each set of tied slow row sectors carries relative energy .50, so $\delta = .30$ satisfies the globally slowest sufficient margin equation 33. That certificate relies on the designed ties; the generalized stage theorem does not require them. Table 2 reports first crossings and least-squares slopes against $\log(\alpha^{-2})$. The fitted slopes over the three displayed initializations differ from the asymptotic predictions by at most 6.4%; tightening the integration tolerance changes the slowest crossing by less than .001. For the aligned arm, the residual slope gap reflects the bounded post-exit settling remainder, especially the faster column sectors settling on an approximately λ^{-1} timescale, rather than a different leading coefficient for slow-sector escape.

At $\alpha = 10^{-9}$, training MSE reaches .02 at 538.27, 542.22, and 542.22, before held-out first passages at 2519.69, 11606.97, and 81926.87. The 10^{-6} runs also memorize first. The fit threshold exceeds $f^2\|y\|_N^2 = .0118343$. A sufficient memorization-first condition is equation 73: the training residual bound ensures fit by a fixed time, and the first-passage lower bound keeps the held-out crossing later. This is a sufficient asymptotic condition; the displayed finite runs establish separation through their measured crossings. Table 2 retains $\alpha = 10^{-3}$ as an ordering-only check: aligned crosses at 113.09, before training fit at 119.33.

Gaussian-table check. The separate Gaussian control uses seeds $0, 1$ with the same design, decay, and threshold. Both have nonpositive globally weakest modes. Numerical evaluation of equation 30 nevertheless brackets $\delta = .30$ in a positive-rate stage for every arm. The seed-0 critical rates are approximately .00714134, .00271493, and .000672418, respectively; the seed-1 rates are .00754326, .00257840, and .000654402. Table 3 reports the reproduced crossings. These are numerical checks of stage quantities and finite-initialization dynamics, not symbolic certificates or exact finite-initialization slope predictions; the scripts separate exact-family certificates from numerical diagnostics.

Arm	$\alpha = 10^{-3}$	10^{-6}	10^{-9}	Fit	Theory
Aligned / supplied (1)	113.09	1296.29	2519.69	87.10	92.80
Scrambled	2248.63	6897.45	11606.97	338.69	341.83
Fragmented	23392.41	52659.77	81926.87	2118.43	2118.42
Supplied $(.4)$	2387.65	5822.22	9364.27	252.49	269.50

Table 2: Faithful-model first passages at relative held-out MSE .30. The supplied gain-1 comparator is exactly aligned. Slope columns compare the three-point fit with $1/(2r_{\pi,\delta})$; finite-initialization fits are not exact asymptotic coefficients.

For (i), supplied gain .4 multiplies *every* circuit column, including the shared columns. Its floor .035208 and forced energy .50 verify the same threshold condition. Table 4 separately checks the exact degree-three built model and the control obtained by replacing $w_n^\top e_r$ with $1 + w_n^\top e_r$. The latter admits a two-factor bypass and is not an instance of the depth-trapping conclusion.

The conservative permanent-trap certificate in (ii) fails at $\alpha = 10^{-3}$, so the censored runs establish only finite-horizon noncrossing. It holds at 10^{-6} and 10^{-9} for both pure built gains. The simulation code records the certificate and full-parameter gradient checks.

Comparator	$\gamma = .4$	$\gamma = 1$
Supplied	2387.65	113.09
Built, no row bypass	> 200000	> 200000
Built, constant row bypass	4508.19	1347.24

Table 4: Built and bypass controls at $\alpha = 10^{-3}$ and $\delta = .30$, which aligned passes at 113.09. A supplied arm has 1056 circuit parameters (216 active); built/bypass arms have 816 active circuit parameters. All have 6272 private parameters: circuit counts are *not matched*. "> 200000" is censoring, not a proof of permanent trapping.

I Complete Experimental Specification

This appendix records everything needed to reproduce Section 3.

I.1 Data scales and splits

Table I.1 records the final $p = 67$ mixed-family setup used by the mask-corrected protocol. For each of the 16 observed operations, a stable hash partitions the $67^2 = 4{,}489$ argument pairs into 1,347 training, 1,571 development, and 1,571 final pairs. Training contains all 21,552 single-step examples from the training pools and 8,000 distinct programs of length two, three, or four whose adjacent operation pairs exclude a fixed set of held-out transitions; 52.7% of scored positions occur in multi-step programs. We also define an affine control family $y = (a_r x + b_c x') \bmod 67$, but the mixed nonlinear family is the primary experiment.

Quantity	Count
Operations	20
Held-out / observed operations	4 / 16
Argument pairs per operation	4,489
Train / development / final pairs per observed operation	1,347 / 1,571 / 1,571
Single-step training examples	21,552
Multi-step training programs	8,000
Total training examples	29,552
Scored training positions	45,554
Argument-OOD examples / scored positions	25,136 / 25,136
Factor-recombination examples / scored positions	17,956 / 17,956
Composition-OOD examples / scored positions	2,000 / 6,002
Crossed-OOD examples / scored positions	2,000 / 6,002
Chance accuracy	1.49%

Table 5: Final $p = 67$ benchmark scale. "Programs" are multi-step examples of length two, three, or four.

The four evaluation regimes are:

1. **factor recombination** (primary): all argument pairs from the four row–column cells R_0C_0, R_1C_3, R_3C_2, and R_4C_1, which never appear in training, while every constituent row and column marker appears in other observed cells;
2. **argument OOD**: development-pool argument pairs from the 16 observed cells;
3. **composition OOD**: length-two-to-four programs with exactly one held-out ordered transition between otherwise observed operations, using training-pool arguments;
4. **crossed OOD**: the same held-out-transition condition using held-out argument pairs.

I.2 Model configurations

Table I.2 summarizes the three scales. All models use the serialization and scoring of Section 3 and the objective of Equation 11.

	XS	M	OLMoE-1B-7B
Layers / width	2 / 128	6 / 512	16 / 2048
Experts per layer / active	8 / top-1	8 / top-1	64 / top-8
Parameters	2.3M	107M	6.9B (1.3B active)
Initialization	scratch	scratch	pretrained
Learning rate	10^{-3}	10^{-3}	10^{-3}
Evaluation interval	5K updates	1K updates	50–250 steps

Table 6: Model scales. XS and M are trained from scratch; OLMoE is the pretrained OLMoE-1B-7B-0125 checkpoint, fully fine-tuned.

XS. XS is a two-layer, pre-LayerNorm decoder-only Transformer with width 128, four attention heads, learned absolute positional embeddings, maximum sequence length 40, and no dropout. Each feed-forward block is an eight-expert MoE with expert hidden width 512 and hard top-1 straight-through routing. All expert outputs are computed in the implementation before the hard gate is applied; sparsity is therefore a scientific routing intervention rather than a claim of sparse-kernel speedup. Every optimizer update accumulates over the complete 29,552-example training corpus. The microbatch size of 4,096 is a memory partition: cross-entropy and the group-balanced alignment objective are accumulated to reproduce their full-corpus gradients exactly. The nonlinear load-balancing and router z losses are computed per microbatch and then weighted by its share of scored positions; padding is excluded from both. The grouped settings minimize

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + 0.3\,\mathcal{L}_{\mathrm{route}} + 0.01\,\mathcal{L}_{\mathrm{bal}} + 10^{-4}\mathcal{L}_z, \tag{94}$$

and the two ungrouped settings keep the same load-balancing and z terms with the route coefficient set to zero.

M and OLMoE. M is a six-layer decoder with width 512 and eight top-1 experts per layer, trained from scratch. OLMoE starts from the pretrained OLMoE-1B-7B-0125 checkpoint (Muennighoff et al., 2024) and fine-tunes all parameters with learning rate 10^{-3}; its routers select the top eight of 64 experts per layer. M has 8 attention heads and experts of width 2,048 (4× the model width), with 107.2M total and 19.0M active parameters. Training follows XS except for the chunk size: each update is the exact gradient of the full training corpus, accumulated over chunks of 2,048 programs (4,096 at XS), with the same optimizer, warmup, constant schedule, clipping and weight decay. $\mathcal{L}_{\text{route}}$ is applied at the <DEC> positions of all six MoE layers, averaged over layers, and updates only the router weights; the coefficients are those of XS. At both scales, all settings of a run start from the same initialization; none branches from a parent checkpoint. OLMoE starts from the pretrained OLMoE-1B-7B-0125 checkpoint (16 layers, 64 experts per layer, top-8 routing) and fine-tunes all parameters with AdamW at learning rate 10^{-3} (300 warmup steps, constant schedule). Instead of new embeddings, each task symbol is mapped to a single existing vocabulary token. Updates use minibatches of 1,024 programs rather than the full corpus. $\mathcal{L}_{\text{route}}$ is applied at the <DEC> positions of all 16 MoE layers and updates only the routers. Its group prototypes are averaged over the last 25 minibatches, and its coefficient is calibrated over the first 20 updates and then frozen. OLMoE's own load-balancing and router z-losses are kept. All settings start from the same pretrained checkpoint.

I.3 Optimization and evaluation schedule

The XS model is trained with BF16 autocast and FP32 parameters using AdamW with learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, a 500-update linear warmup, and a constant learning rate thereafter. The global gradient norm is clipped at 1.0. For the weight-decay settings, coefficient 0.1 is applied to parameters with at least two dimensions and is therefore active on embeddings, attention matrices, expert matrices, router gates, and the output head; biases and LayerNorm parameters are excluded. Checkpoints are saved every 2K updates. The logged trajectory metric is teacher-forced answer-token accuracy; this distinction matters on the multi-step composition and crossed splits, whereas the single-step factor-recombination metric is unaffected by teacher forcing. Recorded horizons differ across settings because runs that had clearly generalized were stopped early: at XS, the grouped settings run for 25–65K updates and the ungrouped settings for 125–260K; at M, all settings run for 7–12.4K updates; at OLMoE, for 1,500–4,250 steps.

Cost per update. All comparisons are stated in optimizer updates. To convert them to time, we ran each setting alone on one NVIDIA H200 for 40 updates with its exact training configuration and report the median wall-clock time over updates 11–40, excluding evaluation and checkpointing. At XS, computing the group prototypes of Equation 11 takes an additional gradient-free forward pass over the training corpus, so a grouped update costs about $1.8\times$ an ungrouped one: 0.66–0.69 s for the aligned and 0.65–0.68 s for the matched-random settings, against 0.37 s for both ungrouped settings. At M, the prototypes are formed from the router inputs already computed in the training forward pass, which yields the same update up to floating-point summation order, and all six settings cost 0.94–0.98 s per update. Aligned and matched-random settings therefore cost the same at both scales, so the reductions in updates between them in Section 4.1 are also wall-clock reductions; the same holds against the ungrouped settings at M. At XS, the extra pass is outweighed by the reduction in updates: computation-aligned grouping reaches 90% held-out accuracy after about 4.4 h (22.9–23.8K updates), compared with about 14 h for weight decay alone (138.7K updates), roughly $3\times$ sooner in wall-clock time. At OLMoE the per-update cost is higher and the gap in updates is smaller. One update (1,024 examples on one H200) takes about 19 s of wall-clock time: 19.2–19.8 s with the routing term and 18.5–19.1 s without it, so the extra pass adds about 4% per update. Computation-aligned grouping reaches 90% held-out accuracy after 436–542 updates (column 436–477, row 439–542), which is about 2.5 h for column routing and 2.7 h for row routing. Weight decay alone takes about 3.1 h (483–684 updates), so computation-aligned grouping is roughly 1.2× sooner in wall-clock time. The seed-to-seed spread, about 100 updates, is comparable to this gain.

I.4 Setting matrix and controlled branching

Table I.4 lists the six settings. Each matched-random control is a fixed scrambling of the 16 observed cells with the same number of groups and the same group-size multiset as its aligned partition after the four-cell holdout. We compare each aligned partition only with its own matched control: comparing the five row groups with the four column groups would change both the shared computation and the strength of the constraint, since the objective averages group means rather than examples. The matched controls equalize partition size and nominal objective weight, but they do not guarantee equal realized route-loss gradients. All settings use AdamW with decoupled weight decay on every weight matrix, including routers and embeddings. In the XS seed-23 run used for the expert analysis of Section 4.2, the two ungrouped settings train from the same initialization with identical frozen data and deterministic microbatch order; after the weight-decay parent reaches an interpolation checkpoint, the four grouped settings branch from that exact checkpoint, inheriting its parameters and optimizer state. This construction gives a particularly clean comparison among the aligned and matched-random interventions, but the comparison with the ungrouped settings should be read as a difference in post-interpolation training regime rather than as independent from-scratch runs.

Setting	Weight decay	Route grouping	λ_{route}
`vanilla_wd0`	0	none	0
`vanilla_wd01`	0.1	none	0
`factor_row_wd01`	0.1	5 row groups	0.3
`random_row_matched_wd01`	0.1	5 fixed matched groups	0.3
`factor_col_wd01`	0.1	4 column groups	0.3
`random_col_matched_wd01`	0.1	4 fixed matched groups	0.3

Table 7: Six controlled settings, shared by all three scales.

I.5 Implementation invariants

For reproducibility, the following properties of the XS runs are treated as assertions rather than informal intentions:

- the two ungrouped settings share their initialization; the four grouped settings inherit the exact parameters and optimizer state of the weight-decay parent at its interpolation checkpoint;
- every setting trains on the same frozen 29,552-example corpus, fixed independently of the initialization;

- the route loss sees all eight soft expert probabilities at scored answer positions, while task execution uses hard top-1 straight-through routing;
- route-loss hidden states and full-corpus group prototypes are detached, so the auxiliary path directly updates router weights only;
- groups, and then positions within groups, receive equal weight as specified in Equation 11;
- gradient accumulation exactly reproduces the full-corpus cross-entropy and alignment gradients for any microbatch size;
- padding contributes to neither load balancing nor router z-loss;
- random group assignments are fixed, are formed only over trainable cells, and match the corresponding aligned group-size multiset after the holdout;
- all settings share deterministic computation and corpus order, and thresholds not reached within the recorded horizon are right-censored.

I.6 Averaging, crossings, and the gap

Common grid. Averages use only the evaluation steps that every averaged run of a model shares: every 5K updates for XS, every 1K for M, and steps 0, 100, 250, 500, …, 1,500 for OLMoE. Differences smaller than one interval are within measurement resolution.

Mean curve and band. Every curve averages all independent training runs at that scale. At each grid step the curve is the mean over runs, and the shaded band spans the lowest to the highest run. Several runs were stopped once they had generalized. A run whose last held-out accuracy is at least 95% is held at its last evaluated value at later grid steps, so that it remains in the average; the carried value is the observed one and is never raised. Runs stopped before generalizing are not extended, so the averaged curve for that setting ends where the shortest such run ends—for example, the M ungrouped averages end at 7K, the shortest horizon among those runs.

Crossings. T_{50}, T_{90}, and T_{99} are measured on factor-recombination accuracy; thresholds not reached within the recorded horizon are right-censored and reported as "$>$" the last averaged step. Each is the first step at which the averaged held-out curve reaches 50%, 90%, and 99%, linearly interpolated between neighbouring evaluations, starting from 0% at step 0 when there is no step-0 evaluation. A crossing can only be located to within one evaluation interval; this matters most for XS, where a jump from 20% to 100% between two evaluations places all three crossings within the same 5K interval.

Learning window and gap. The learning window of a model runs from its first grid step to the first step at which all four grouped settings have mean held-out accuracy of at least 99% (XS: 5–35K; M: 1–7K; OLMoE: 0–1,250). The gap is the time-weighted mean of the aligned minus matched-random accuracy difference over this window,

$$\mathrm{gap} = \frac{1}{t_{\mathrm{end}} - t_{\mathrm{start}}} \int_{t_{\mathrm{start}}}^{t_{\mathrm{end}}} \big(\mathrm{acc}_{\mathrm{aligned}}(t) - \mathrm{acc}_{\mathrm{random}}(t)\big)\, dt, \tag{95}$$

computed with the trapezoidal rule so that unevenly spaced evaluations are weighted by the time they cover, separately for rows and columns.

J Additional Experimental Results

J.1 Scope: the observed effects are properties of the training phase transition

Our curves show that every effect we report is confined to the phase transition from memorization to generalization, and we state this limitation explicitly.

Timing, not final accuracy. Every setting that generalizes within its horizon converges to essentially perfect held-out accuracy (Figures 2 and 6). The advantage of computation-aligned grouping is therefore visible only during the transition: the gap in Figure 7 rises as the grouped settings begin to

generalize and returns to zero once both have finished. Computation-aligned grouping changes *when* the transition occurs, not the quality of the solution it ends in, and our T_τ and gap statistics measure only this shift. We make no claim about asymptotic generalization; with a sufficiently long budget, weight decay alone also reaches the same accuracy at scale XS.

The transitions are abrupt. At scale XS, held-out accuracy in each run rises from near chance to above 90% within a few 5K-update evaluation intervals, so T_{50}, T_{90}, and T_{99} often fall within the same interval and are resolved only to that precision (Appendix I.6). Differences between settings are best read as shifts in the onset of a sharp transition rather than as differences in a gradual learning rate.

The expert signatures are transient. The mechanistic evidence of Section 4.2 is likewise tied to the transition. The route-swap effect and the factor-selective experts peak between T_{50} and T_{99} and disappear within about 5K updates as the last-layer experts converge (Figures 3c and 11), and the same convergence occurs in the matched-random and ungrouped models once they generalize. These measurements therefore describe how the model passes through the transition, not the structure of its final solution, and they do not show that factor-specific experts are necessary for generalization.

The transition depends on the regime. The pretrained OLMoE model fits and generalizes within a few hundred steps, so it shows at most a weak delayed-generalization phase and correspondingly smaller differences between settings. At scale M, part of the acceleration is shared by any routing constraint. Whether computation-aligned grouping shortens the transition in regimes with longer or less abrupt delays remains to be tested.

J.2 Full training horizon at scale XS

Figure 6 extends the XS curves of Figure 2 to the full recorded horizon. Weight decay alone does eventually generalize, but its averaged curve first reaches 50% only at 76K updates, and no run of the no-decay setting generalizes at all.

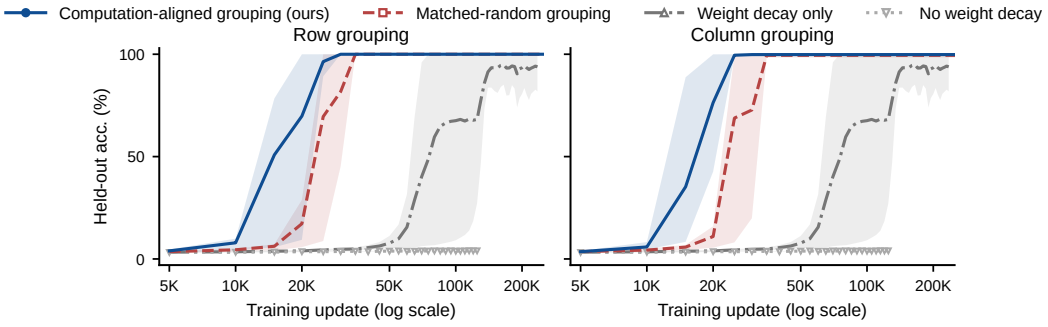

Figure 6: **XS over the full training horizon** (log step axis; mean over runs with min–max band). The weight-decay-only mean plateaus below 100% because not every run completes the transition within its horizon.

J.3 The aligned advantage over time

Figure 7 plots the accuracy difference between each aligned setting and its matched-random control at every evaluation. The advantage is concentrated in the transition and returns to zero once both settings have generalized, which is why we summarize it over the learning window rather than at the end of training.

J.4 Additional evaluation splits

Figure 8 reports the three secondary splits. Argument and crossed OOD closely track factor recombination at every scale, with the same ordering among settings. Composition OOD, which recombines only observed operations, is learned much earlier by every setting and does not separate them.

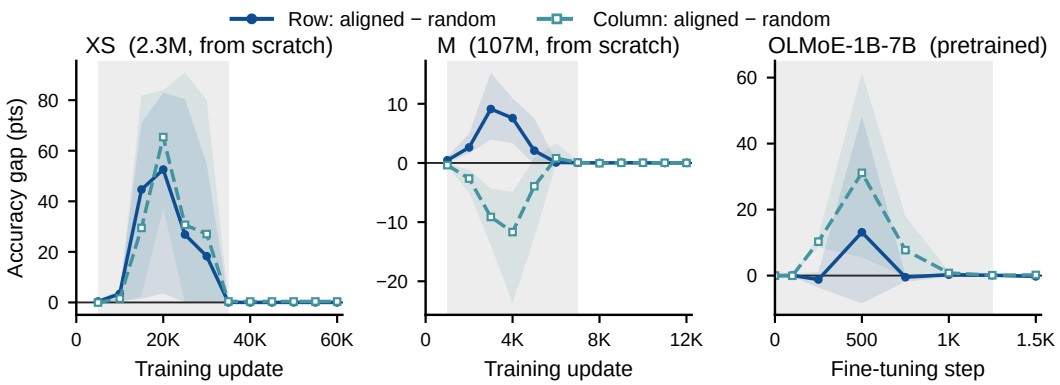

Figure 7: **Aligned minus matched-random held-out accuracy** (mean over runs with min–max band), where "aligned" denotes computation-aligned grouping with ComRoute. The grey region is the learning window over which the gap of Table 4.1 is averaged.

J.5 Training loss

Figure 9 shows the training cross-entropy at the answer tokens. At XS and M, every setting has essentially fit the training set before the first evaluation, so the differences in Figure 2 concern generalization rather than fitting.

K Expert Analysis Details

Setup. All measurements use the saved checkpoints (every 2K updates) of the XS seed-23 run, in which the six settings share initialization and data order. For each setting we use the checkpoint nearest its T_{50}, T_{90}, and T_{99}, estimated from 2K-update evaluations of held-out accuracy (aligned column: 13.2K, 15.5K, 17.1K; aligned row: 14.4K, 15.7K, 16.0K; matched-random column: 22.3K, 23.7K, 24.0K; matched-random row: 20.6K, 21.7K, 22.0K).

Gradient agreement. We take 96 training examples from each of the 16 training cells and pass all of them through the same expert. For each example we compute the gradient of its loss with respect to that expert's weights, and measure the similarity of two examples by the cosine between their gradients. We average this cosine over pairs from different cells that share a row, a column, or a matched-random group, and divide by the average over all pairs. This is repeated for all 16 experts (2 layers $\times$ 8) and for the five trained settings; Figure 3a reports the mean and range over settings. The ordering is absent at initialization and appears at every later checkpoint, in all 16 experts for columns and in 14–16 of the 16 experts for rows.

Route swap. With weights frozen, for each held-out example we change only the last layer's route at <DEC>, to the expert used by most training examples of another column (or row) in that layer, and let that expert's output replace the original one. We count how often the answer becomes the one that the other column (row) would produce, subtract the same count without the swap, and average over the other columns (rows). In the matched-random models the effect stays within 3 points in magnitude at every checkpoint.

Forcing every expert. To remove the dependence on the chosen swap target, we send all held-out <DEC> tokens through one last-layer expert at a time. For each column (row), we count how often the answer becomes that column's (row's) answer among tokens that belong to other columns (rows), and subtract the same count without forcing (Figure 10). At T_{50} in the aligned-column model, each of the eight experts implements exactly one column function, two experts per column, shifting 60–74 points of answers to that column.

Expert disagreement. We send all held-out examples through each last-layer expert in turn and, for every pair of experts, count the share of examples on which their answers differ, averaging over

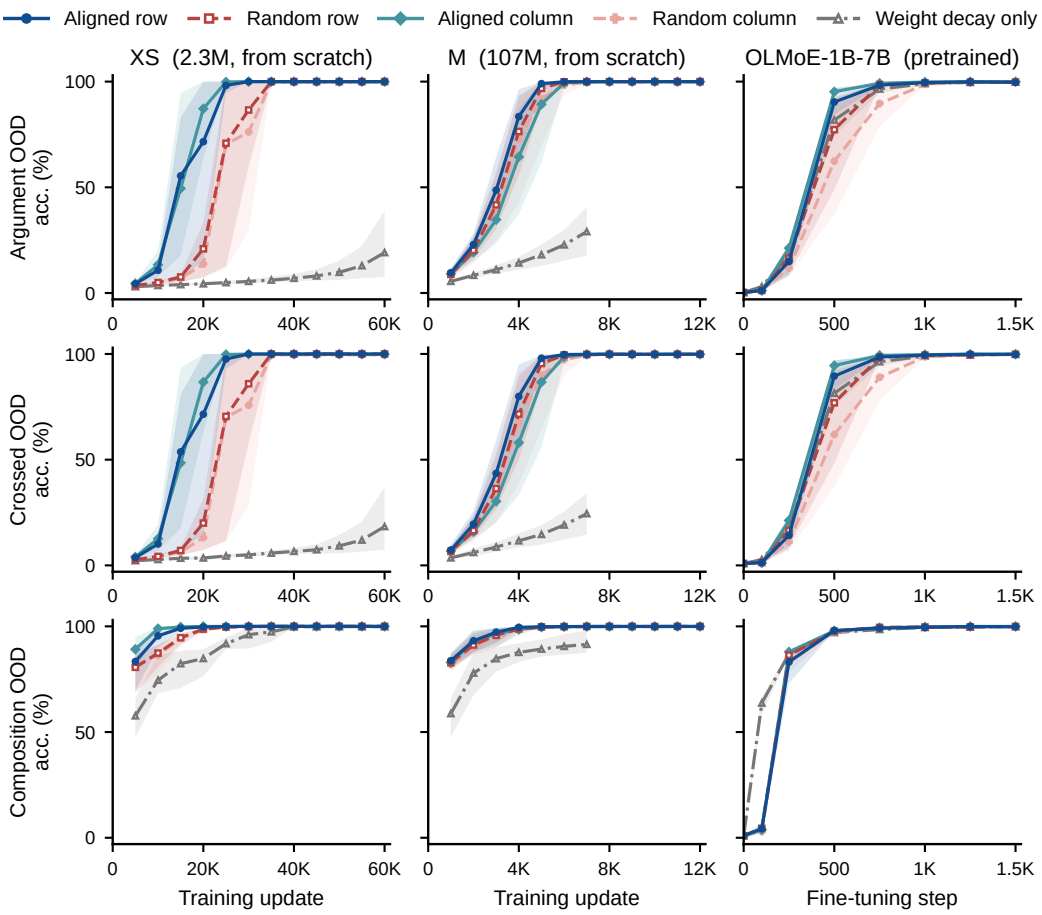

Figure 8: **Argument, crossed, and composition OOD accuracy** (teacher-forced answer-token accuracy; mean over runs with min–max band). "Aligned" denotes computation-aligned grouping with ComRoute and "random" its matched-random control. The no-decay setting is omitted for readability.

the 28 pairs. Figure 11 shows that the experts of every trained model stay distinct until that model generalizes and then converge within a few thousand updates. In the aligned-column model, two experts still disagree on 75% of held-out examples at T_{99}, when held-out accuracy is already 99.7%, and on none about 5K updates later. In the aligned-row model, disagreement falls from 90% at T_{50} to 16% at T_{90} and to zero by 18K updates.

L Limitations

First, the theory is deliberately stylized. It proves asymptotic first-passage comparisons for a fixed-route model with a separate private-memory channel, balanced small initialization, squared-loss gradient flow, and exact reusable sectors. These assumptions make the mechanism analytically tractable, but they do not describe the full dynamics of the Transformer, whose routes and representations are learned jointly. We therefore treat the theory as identifying sufficient mechanisms for faster generalization, rather than as a direct mechanistic proof of the Transformer results.

Second, the computation groups are supplied by the benchmark. This allows a controlled test of whether aligning parameter sharing with a known reusable computation accelerates generalization, but it does not address how such groups would be obtained in realistic settings. Extending the method beyond controlled tasks will require discovering or inferring reusable computations from

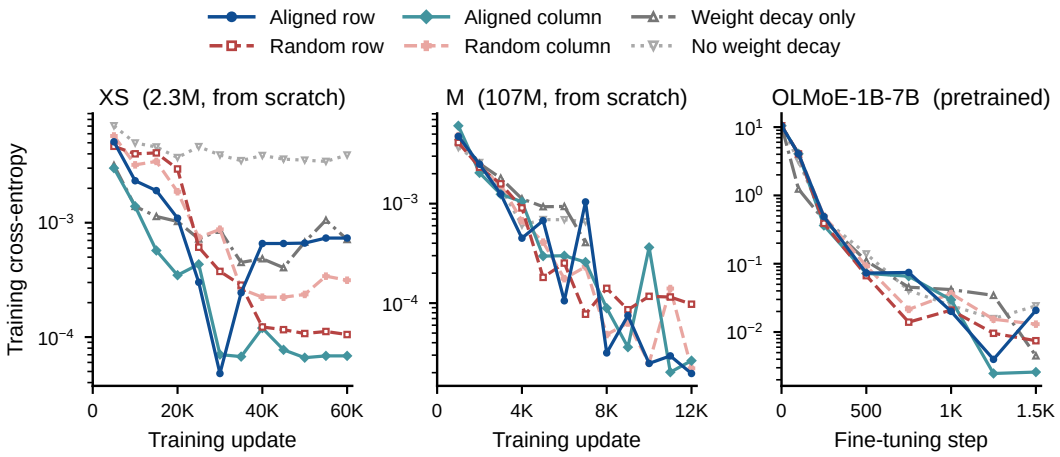

Figure 9: **Training cross-entropy** at each evaluation (log scale; mean over runs). "Aligned" denotes computation-aligned grouping with ComRoute and "random" its matched-random control.

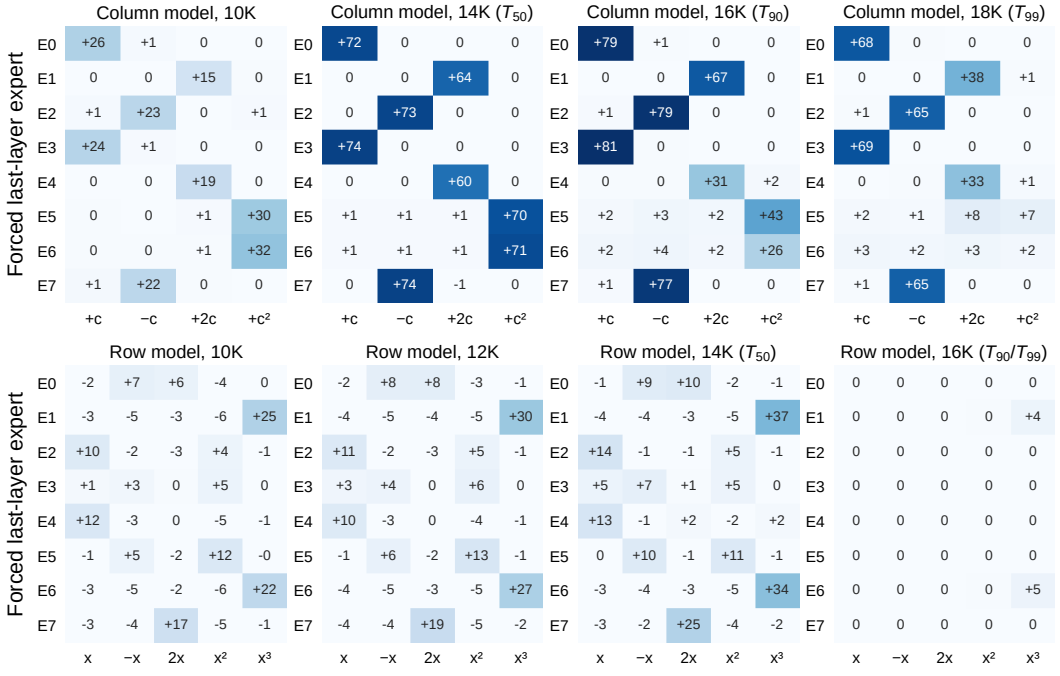

Figure 10: **Forcing each last-layer expert** on all held-out <DEC> tokens (XS, seed 23). The column and row models are the computation-aligned column and row groupings. Each cell is the increase, in percentage points, in answers that become the given column's (top) or row's (bottom) answer. In the aligned-column model every expert implements a single column function, two experts per column, strongest at T_{50}–T_{90}; the row model shows weaker row selectivity that vanishes by T_{90}.

data, representations, execution traces, or other forms of supervision rather than relying on gold rule labels.

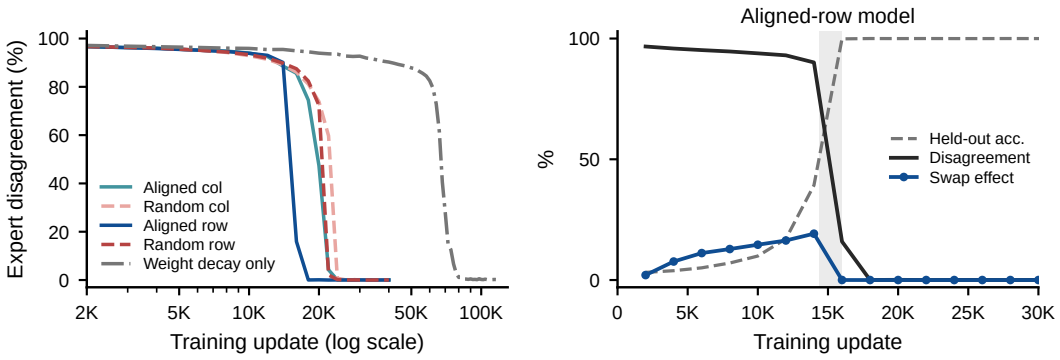

Figure 11: **Expert convergence follows generalization.** "Aligned" denotes computation-aligned grouping with ComRoute and "random" its matched-random control. Left: last-layer expert disagreement for all trained settings. Each curve collapses shortly after that setting's own transition. Right: the aligned-row counterpart of Figure 3c.