INFORMATIVE SHORTCUTS: HOW SPURIOUS CORRELATIONS CAN FACILITATE ROBUST GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A common response to spurious correlations is to weaken them, for instance by balancing the training data so that the shortcut feature no longer predicts the label. We find that in small transformers this can make the robust rule harder to learn. In our tasks, a shortcut feature agrees with the label on a fraction r of training examples and contradicts it on a held-out adversarial split. When the label is the sum parity of an integer sequence and the shortcut is the parity of its largest element, the fraction of two-layer seeds that reach 100% adversarial accuracy grows from 0% at $r{=}0.5$ to 53% at $r{=}0.9$, whereas for one-layer models it drops from 33% to 0%. The benefit depends on how far r is from chance rather than on the direction in which it deviates. On a ternary task, two-layer models generalize in 80–93% of seeds both when the shortcut is correlated and when it is anti-correlated with the label, and in no seeds at chance. In that setting they memorize the training set, a failure that disappears with $16\times$ more data. All models learn the shortcut-only predictor first, and in the runs that succeed the robust rule emerges while attention to the shortcut token remains high. Finally, models trained briefly with an informative shortcut and then moved to balanced data still generalize in most seeds, which suggests that the shortcut serves as an optimization scaffold rather than working by amplifying the gradient of the misclassified minority.

1 INTRODUCTION

Spurious correlations are one of the central failure modes of learned models (Geirhos et al., 2020; Shah et al., 2020). When a feature is highly predictive in training but unreliable at test time, models learn to use it as a shortcut. The standard remedies change the group structure of the data, either by minimizing worst-group risk (Sagawa et al., 2020a) or by subsampling the majority group so that shortcut-consistent examples no longer dominate gradient descent (Sagawa et al., 2020b).

We describe a regime that runs against the intuition behind these remedies. In our sweeps, a statically balanced dataset, which places the spurious ratio at the random-chance baseline, is the *worst* setting for sufficiently capable models. Six of the eight two-layer configurations we test generalize in 0% of seeds there, and the best reaches 10%, compared with 53–93% at ratios far from the baseline on three of the four tasks. Reducing the correlation can therefore remove an optimization signal that helps the model find the robust solution. We study the optimization dynamics of static spurious ratios and do not evaluate group-robustness methods such as Group DRO, subsampling or JTT. Our point is narrower: within a fixed data-generating process, the strength of the correlation affects whether the robust solution is found, so setting it to zero is not a neutral choice.

Our main task is MESP (Section 3), in which models must predict the sum parity of an integer sequence but can exploit the parity of its maximum element. There and on two of three related tasks, two-layer models generalize far more often when the shortcut is informative, and ternary-label variants show that this depends on the distance of r from chance rather than on a shortcut-consistent majority (Section 4.4).

We weigh two mechanisms in Section 6. Under *gradient amplification*, the part of the misclassified minority's gradient that the shortcut does not cancel pushes the model toward the rule. Under the *optimization scaffold* account, the cheap shortcut installs routing that the true rule later reuses. None

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.

of our observational measurements supports amplification, though they cannot rule it out. A transplant experiment favors the scaffold, subject to caveats we discuss.

Our contributions are:

1. **Correlation strength interacts with depth.** Raising the spurious ratio from 0.5 to 0.9 raises the two-layer, two-head generalization rate from 0% to 53% on MESP and from 0% to 77% on FESP, under both the anytime and terminal metrics, while one-layer models become trapped by the shortcut (on MESP, 1L, 1H falls from 33% to 0%). Where the depth threshold lies depends on the task (Appendices C and D).
2. **Distance from chance, not the direction of imbalance.** On FE-Mod3, two-layer, two-head models generalize in 93% of seeds at $r{=}0.15$, where the shortcut is anti-correlated with the label, and in 90% at $r{=}0.50$, but in none at the chance ratio $r{=}0.33$. The chance ratio is the worst two-layer setting on MESP, FESP and FE-Mod3 (Section 4.4, Appendix N).
3. **Two failure modes and the dynamics of escape.** One-layer models stay trapped on the shortcut and are wrong even on the anti-shortcut examples they train on, whereas two-layer models at the chance ratio memorize their training set. Every model first learns the shortcut-only predictor; in successful runs the robust rule forms while attention to the shortcut token is still high, and that attention is withdrawn only afterwards (Section 5).
4. **Interventions that separate the explanations.** Sixteen times more training data rescues the chance ratio (10/10 vs. 0/10 seeds), so that failure is data-limited. Transplanting models from $r{=}0.9$ into balanced training yields 6/10 generalizing seeds against 0/30 from scratch, while upweighting the minority at $r{=}0.9$ to equalize the loss lowers generalization from 16/30 to 1/10. Together these favor the shortcut acting as an optimization scaffold over a persisting gradient asymmetry (Section 6, Appendix K).

2 Background and Related Work

Grokking and Delayed Generalization. Grokking, a sharp rise in test accuracy long after training loss has plateaued, was first reported by Power et al. (2022) on modular arithmetic. Later work studied these transitions in two ways, through mechanistic analysis of the circuit that forms (Nanda et al., 2023) and through progress measures showing that learning advances steadily while loss and error stay flat (Barak et al., 2022). Our setting is related but different. We see a range of outcomes, with some runs generalizing quickly, some slowly and some not at all, and we neither assume nor need a sharp phase transition. In grokking, stronger weight decay helps a model reach and keep the generalizing solution (Liu et al., 2023). In our setting it does the opposite. Raising weight decay from 0.1 to 0.4 leaves the anytime rate intact, cuts the terminal rate (53% to 23% at 2L, 2H, $r{=}0.9$) and does not rescue the null ratio (1/30).

Shortcut Learning and Simplicity Bias. Deep networks have a *simplicity bias* toward the simplest features that reduce training error (Shah et al., 2020; Teney et al., 2025), and under spurious correlations this becomes *shortcut learning* (Geirhos et al., 2020). Which feature a model uses depends jointly on its *predictivity* and its *availability*, meaning how easily it can be extracted from the input, an account with theoretical support from Neural Tangent Kernels (Hermann et al., 2024). Both of our shortcut features are maximally available, and their predictivity is what we sweep.

The Paradox of Data Imbalance. Standard prescriptions intervene on the training signal so that no single group dominates it. Examples are worst-group risk minimization (Sagawa et al., 2020a), majority subsampling (Sagawa et al., 2020b; Idrissi et al., 2022), error-based upweighting (Liu et al., 2021), difficulty-based reweighting (Sinha et al., 2020) and gradient surgery (Mansilla et al., 2021). We describe a regime in which imbalance instead acts as a catalyst. Work on *spurious memorization* localizes the fit to atypical minority examples in a few neurons (You et al., 2025). Our failing two-layer models also memorize the minority, although we have not localized where. Our setting also differs from last-layer retraining, which works when the core feature is already linearly present (Kirichenko et al., 2023). The closest work to ours is Qiu et al. (2024), who study boolean tasks with a tunable spurious feature and find that a stronger spurious correlation *slows* core-feature learning. Our one-layer models behave the same way, and the two-layer reversal is where we depart from them.

The settings differ enough that the two results need not conflict. Their parity has an unknown sparse support, whereas all five of our positions are relevant and known, and their data and architectures differ from ours. Theory gives the scaffold hypothesis a natural home. Correlated, easier features can make parity learnable by gradient descent (Daniely & Malach, 2020), and hierarchical "staircase" structure reduces the leap a network must make (Abbe et al., 2023).

Mechanistic Interpretability of Circuits. We build on techniques for identifying internal circuits in Transformers (Nanda et al., 2023). Where prior work studies prefix-matching induction heads (Olsson et al., 2022) or the Fourier modular-arithmetic "Clock" circuit (Nanda et al., 2023; Zhong et al., 2023), we study the transition between magnitude-sorting and parity-encoding circuits.

3 SETUP

3.1 TASKS

We study four tasks formed by the cross-product of two true labels and two shortcut features, all on sequences of 5 integers drawn i.i.d. from $[0, 20)$. The **true labels** are *sum parity* and *sum mod 3*; the **shortcut features** are the *first element* or the *max-valued element*, taken modulo the same base as the label. The four combinations are Max-Equal-Sum-Parity (MESP), First-Equal-Sum-Parity (FESP), First-Equal-Sum-Mod3 (FE-Mod3) and Max-Equal-Sum-Mod3 (ME-Mod3). An example is *shortcut-consistent* when the shortcut equals the true label and *anti-shortcut* otherwise, and the **spurious ratio** r is the fraction of shortcut-consistent training examples. Because the random-chance baseline is $1/2$ for binary and $1/3$ for ternary labels, the sweeps are $r \in \{0.5, 0.7, 0.9\}$ and $r \in \{0.15, 0.33, 0.5\}$ respectively. Attending to one token is mechanically cheaper than aggregating five, and this asymmetry is what a shortcut imbalance can exploit. The label classes themselves stay approximately balanced (Section 3.2).

3.2 DATASET SPLITS

Each experiment uses three fixed splits: a **train** set of 16,384 examples at spurious ratio r; a **val** set of 2,048 examples at ratio 1.00 (all shortcut-consistent), measuring shortcut learning; and an **adv** (adversarial) set of 2,048 examples at ratio 0.00 (all anti-shortcut), measuring true-rule generalization. "Adversarial" denotes the worst group under the shortcut, not a perturbed input.

Sampling procedure. Every split is built by rejection sampling: we draw a candidate sequence of five integers i.i.d. uniform on $[0, 20)$ and accept it only if it matches the shortcut status that slot requires, has not already been emitted, and (for the magnitude-shortcut tasks) has a *unique* maximum, so that r is exact by construction and both subgroups come from the same sequence distribution. Because we do not condition on the label, the label marginal varies slightly with r: it stays within 0.9 points of uniform on the position-shortcut tasks, and the largest deviation, at MESP r=0.9, is a class frequency of 0.55. On the four primary tasks the adversarial split is disjoint from training in every cell except two ME-Mod3 cells, where overlap is at most 0.3%; on the n11 variant it reaches 1.5% at r=0.9, and the n11 training files at $r \geq 0.7$ contain all 2,048 val sequences, so we do not use n11 val accuracy at those ratios. None of this is large enough to explain the effect, as the audit in Appendix I shows.

Generalization definition. We report two rates, because a run can reach the robust solution without keeping it. The **anytime** rate is the fraction of seeds that answer all 2,048 adversarial examples correctly (100% adversarial accuracy) at some logged epoch, with accuracy logged every 100 epochs. The first such epoch is the *generalization epoch*. The **terminal** rate is the fraction of seeds still at 100% adversarial accuracy at the final checkpoint (epoch 15,000). Every generalization-rate table reports both. The anytime rate answers whether a configuration can find the robust solution at all and conditions all mechanistic analyses in Section 5. The terminal rate is what a practitioner without an adversarial validation set would get, since shortcut-consistent validation accuracy stays near 1.0 and cannot be used to early-stop. At 30 seeds a 95% Wilson interval on a rate near 50% spans about ± 0.17; Appendix N gives intervals and Fisher exact tests for the comparisons we draw, and the peak adversarial accuracy behind the binary outcome.

3.3 Models and Training

Architecture variants are single-layer and two-layer transformers with $n_{\text{heads}} \in \{1, 2\}$, d_{model}=64, and d_{ff}=128 on the two binary tasks. The two ternary tasks (ME-Mod3, FE-Mod3) run at twice that width, d_{model}=128 and d_{ff}=256, because three residue classes did not fit reliably at the base width; depth and head count are swept identically in both cases, so the depth and ratio comparisons we draw are always within a task and at fixed width.

Input and readout. Each example is seven tokens (`START`, the five integers and `END`), with the delimiters taking the two ids above the integer range. Attention is *bidirectional* (no causal mask; the task is sequence classification) and runs over all seven positions, with delimiters excluded from attention statistics where noted. Token embeddings are summed with *fixed sinusoidal* positional encodings, which matters for FESP and FE-Mod3 because their shortcut lives at an absolute position. Each layer is attention then a two-layer ReLU MLP, both residual; there is no LayerNorm and there are no biases. Logits are read from the `END` position through one bias-free linear projection. Appendix H gives the remaining details.

Training runs for 15,000 epochs with AdamW (lr=10^{-3}, weight decay $\in \{0.1, 0.4\}$, batch size 4,096). Each configuration is run with 30 seeds for the two primary binary-label tasks (MESP and FESP) and the two mod 3 tasks (FE-Mod3 and ME-Mod3), and 10 seeds for the remaining ablation variants (n11; Appendix C); compute details are in Appendix O. Every 100 epochs we log training, shortcut-consistent and adversarial accuracy and loss, the subset gradient statistics of Section 5.1, QK/OV statistics and attention summaries.

4 Behavioral Results: Correlation Strength and Capacity

Sections 4.1–4.3 cover the binary-label tasks; Section 4.4 tests the ratio claim on both sides of the null with the ternary tasks.

4.1 Generalization Rate Increases With Ratio for 2-Layer Models

Table 1 shows the main interaction. At 2L, 2H the anytime rate is 0%, 43% and 53% at r=0.5, 0.7 and 0.9, and the terminal rate is 0%, 40% and 53%. Both informative ratios differ sharply from the null (Fisher p=5×10^{-5} and 2×10^{-6}) but not from each other (p=0.61), so our claim is that informative ratios beat the null, and we make no claim of monotonicity in r. Transient excursions do not produce the effect. At r=0.9 all 16 generalizing seeds at 2H and all 7 at 1H are still at 100% at epoch 15,000, four of the 2H seeds after brief collapses of one to eight logged epochs (Figure 2). At this weight decay, regression is confined to one or two seeds per cell (Table 1).

One-layer models show the opposite pattern. The 1L, 1H model generalizes in 33% of seeds with balanced training, and this drops to 7% and then 0% as the ratio rises, so higher imbalance traps training on the shortcut. This fits a threshold between one and two layers, below which imbalance hurts and above which it helps. The threshold is not a matter of what the model can represent. A one-layer model's readout sees one attention-weighted average per head, which carries little beyond

Table 1: Generalization rate by configuration, MESP (weight decay = 0.1). *Anytime* counts seeds reaching 100% adversarial accuracy at any logged epoch; *terminal* counts seeds still at 100% at epoch 15,000.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.5	10/30 (33%)	8/30 (27%)	**1H,** r=0.5	**3/30 (10%)**	**3/30 (10%)**
1H, r=0.7	2/30 (7%)	2/30 (7%)	**1H,** r=0.7	**8/30 (27%)**	**7/30 (23%)**
1H, r=0.9	0/30 (0%)	0/30 (0%)	**1H,** r=0.9	**7/30 (23%)**	**7/30 (23%)**
2H, r=0.5	3/30 (10%)	3/30 (10%)	**2H,** r=0.5	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.7	3/30 (10%)	3/30 (10%)	**2H,** r=0.7	**13/30 (43%)**	**12/30 (40%)**
2H, r=0.9	3/30 (10%)	3/30 (10%)	**2H,** r=0.9	**16/30 (53%)**	**16/30 (53%)**

the shortcut once the only head at 1L, 1H is pinned to it. At 1L, 2H, however, one head could hold the shortcut while the other passes the token-parity count to the MLP, which can express the rule (1L, 1H itself learns it with one head at $r{=}0.5$). 1L, 2H is nonetheless trapped at $r{=}0.9$ (3/30 on MESP, 1/30 on FESP), so its trap is dynamical, and we cannot explain it.

Weight decay. Raising weight decay from 0.1 to 0.4 leaves the anytime rate broadly intact, shifts it across ratios and sharply lowers the terminal rate. 2L, 2H reaches 100% in 17/30 seeds at $r{=}0.7$ and 12/30 at $r{=}0.9$ but holds it in only 2/30 and 7/30, and no 2L, 1H cell in that sweep exceeds 1/30 terminal. Higher weight decay lets models find the generalizing basin, but most of them do not stay there. The anytime count at $r{=}0.7$ is higher than at $r{=}0.9$ in this sweep, though not significantly (17 vs. 12, $p{=}0.30$; Appendix A).

4.2 Shortcut-Trapping in 1-Layer Models

One-layer models at $r \geq 0.7$ rely heavily on the shortcut. Their accuracy on shortcut-consistent examples is near perfect (0.90–1.00 mean per configuration), while adversarial accuracy stays near zero (0.01–0.20) for the whole 15,000-epoch run. Accuracy this far below chance is itself the sign of reliance, since the adversarial split is built so that the shortcut contradicts the label and a model that follows it is systematically wrong. Appendix I shows that these models are wrong even on anti-shortcut sequences they were trained on. Two-layer models fail differently. At $r{=}0.9$ their non-generalizing seeds fit, on average, at least 0.90 of the anti-shortcut training examples while scoring 0.13 on held-out ones, so they memorize the minority.

4.3 Cross-Shortcut Generalization: Position vs. Magnitude

In FESP the shortcut is purely positional, so the model only has to read a fixed position and never ranks tokens by magnitude. The qualitative pattern carries over (Appendix B), with two-layer models generalizing far more often at informative ratios than at the null. FESP counts are higher in every two-layer informative cell (77% against 53% at 2L, 2H, $r{=}0.9$), but the difference is not significant ($p{=}0.10$), so we make no claim that the effect size depends on the shortcut type. High r hurts one-layer models on both tasks, and only with the magnitude shortcut are they also best at the null.

4.4 Ternary Labels: Deviation From the Null in Either Direction

The effect follows the distance from the uninformative correlation, and the size of a shortcut-consistent majority does not explain it. In FE-Mod3 the true label is sum mod 3 and the shortcut is the first element mod 3, so the null correlation is $1/3$ and the sweep $r \in \{0.15, 0.33, 0.50\}$ places informative ratios on both sides of it. The 2L, 2H model generalizes in 93% of seeds at $r{=}0.15$ and 90% at $r{=}0.50$, and in 0% at $r{=}0.33$. At $r{=}0.15$ the shortcut is *anti*-correlated with the label and shortcut-consistent examples are the minority, yet this is where generalization peaks. The effect appears whenever the shortcut is informative, in either direction, and disappears when it is uninformative. Nothing saturates here either. The shortcut only rules out one class, and models sit at 0.50 accuracy and loss 0.86 on the anti-shortcut majority before they generalize (Appendix D). The collapse at the null depends on depth. One-layer models do far worse at the two informative ratios (7–37% against 80–93%), but at $r{=}0.33$ the order reverses, with 1L, 1H at 11/30 (37%) and both two-layer configurations at 0/30 (Table 9).

On the second ternary task the ratio effect is weak and the depth effect inverts. On ME-Mod3 (magnitude shortcut, ternary label) the null is still the lowest two-layer cell, but the evidence is thin. 2L, 2H generalizes in 8/30 seeds at $r{=}0.15$ against 1/30 at the null ($p{=}0.026$), and 2L, 1H sits at 0/30 both at the null and at $r{=}0.50$. Two-layer generalization on this task comes late (median $12{,}850$ of $15{,}000$ epochs), so these counts are probably right-censored. The best cell is a *one*-layer model, 1L, 1H at $r{=}0.50$, which generalizes in 15/30 seeds (50%), more than any two-layer cell (at most 8/30, 27%); the same ordering holds at weight decay 0.4. The capacity threshold between one and two layers therefore holds on three of our four tasks (Appendix D).

5 MECHANISTIC ANALYSIS

This section examines what happens during training on MESP. Once the shortcut is learned, the shortcut-only posterior produces an $r/(1-r)$ gradient asymmetry, which is not on its own evidence of amplification (Section 5.1). In capable models the robust rule then forms (Section 5.2), and attention to the shortcut token is withdrawn afterwards (Section 5.3). These analyses are correlational, and apart from the layer marker of Section 5.3, none of them clearly tells a seed that will generalize from one that will not before it does so. Gradient statistics are raw (not Adam-preconditioned) and are computed on the held-out splits at every logged epoch, and for MESP at $r{=}0.9$ also on the training subsets (Appendix F). In failing models the true rule never forms. Sum parity is not linearly decodable from the readout (never more than 0.018 from chance in any cell), while the shortcut is decodable at 0.99, and last-layer retraining on a group-balanced set does not lift adversarial accuracy above chance (Appendix J).

5.1 GRADIENT CONFLICT DYNAMICS

We compute the cosine similarity between the gradients of the shortcut-consistent and anti-shortcut subsets. Strongly negative values mean the subsets pull the parameters in opposite directions, and values near zero mean the gradients are orthogonal. We also compute the ratio of the anti-shortcut to the shortcut-consistent gradient norm.

Gradient conflict appears early in every configuration, whether or not the model generalizes. Shortcut-trapped one-layer models at $r \geq 0.7$ then keep the cosine similarity near -0.9 for the rest of training (Figure 1, Appendix F). Per-configuration standard deviations are comparable to the means (Table 17), so we look at individual seeds. Split by outcome, the final cosine similarity is higher for generalizing than for non-generalizing seeds in all ten configurations where both occur (Table 18), for example -0.087 vs. -0.813 and $+0.389$ vs. -0.796 at 2L, 2H with $r{=}0.7$ and 0.9. Final values are a weak test, because the gradients of a model at near-zero loss are poorly determined. At the last logged epoch before adversarial accuracy first exceeds 0.25, generalizing seeds at $r{=}0.9$ are still anti-aligned (median $\cos\theta{=}-0.88$), so positive alignment follows escape and does not precede it.

Resolving the conflict is not enough to generalize. Two-layer models at $r{=}0.5$ resolve it without any gradient asymmetry (final cosine 0.122 and ratio 0.987 at 2L, 2H), yet they are the two weakest two-layer cells in Table 1.

The norm ratio also separates outcomes, but the extreme values belong to the *trapped* models, at roughly $9\times$ for $r{=}0.9$ (8.96 at 1L, 1H and a median of 9.2 among non-generalizing 2L, 2H seeds). For one-layer models this is the value the calibrated shortcut posterior implies (Section 6); for the two-layer memorizers we have no derivation of it. Generalizing seeds end well below it (medians 2.70 at 2H and 2.85 at 1H) as a consequence of generalizing. Before escape the ratio does not separate outcomes. Compared with trapped seeds of the same cell at the same epoch, the nine late 2L, 2H escapers (first adversarial accuracy above 0.25 at or after epoch 1,900) are higher in 8 of 9 on the held-out splits just before escape ($p{=}0.04$), an epoch already inside the transition. On the training subsets, which the force-balance argument concerns, the count is 6 of 9 ($p{=}0.51$), and at fixed leads of 500 and 1,000 epochs it is 7 of 9 ($p{=}0.18$) and 4 of 9 (Table 19).

5.2 CIRCUIT EVOLUTION

We say a head has developed a *robust circuit* at checkpoint t when activation ablation and cross-split patching both implicate it. Ablating the head must reduce the correct-class logit margin on *both* the shortcut-consistent and adversarial splits, and cross-split patching must transfer in both directions with a mean score above 0.5. Appendix H gives the formal criteria and calibrates them. The diagnostic is specific at $r \geq 0.7$ but fires on noise at balanced ratios, so the $r{=}0.5$ column below is not evidence of circuit formation. We report the earliest epoch at which any head in any layer meets the criterion (Table 16).

One-layer, one-head models at $r \geq 0.7$ almost never meet the criterion (2/30 and 0/30 seeds) and stay in the magnitude-sorting regime, and the two seeds that do meet it do so very late. One-layer, two-head models form a robust head at $r{=}0.9$ in only 3/30 seeds, the same three that generalize, so at this depth the criterion and the outcome coincide and both are rare.

Table 2: QK/OV circuit fingerprint across training. QK ρ (peak) is the maximum over all checkpoints and heads, (final) the last-checkpoint value; final values are measured after the outcome is decided and are not predictors of it. OV parity at QK peak is measured at the epoch of highest QK ρ; at gen. at the first epoch of 100% adversarial accuracy, over that row's generalizing seeds only, with the final checkpoint substituted across all 30 seeds in the two rows where *no* seed generalizes. No row mixes the two conventions, and tied ranks cap OV parity Spearman at 0.867, so values near 0.86 are at ceiling. OV parity is signed; in the 1L, 1H r=0.5 row, where the sign varies across seeds, $|\rho|$ at generalization is 0.468 ± 0.289, and in the r=0.9 rows signed and absolute values coincide. Appendix H gives both details and disaggregates by outcome.

	Shortcut circuit (QK ρ)		Robust indicator (OV parity Spearman)	
Config	Peak	Final	At QK peak	At gen.
1L, 1H, r=0.5 (mixed)	0.213 ± 0.104	0.111 ± 0.101	0.033 ± 0.361	-0.062 ± 0.568
1L, 1H, r=0.9 (trapped)	$\mathbf{0.976 \pm 0.010}$	$\mathbf{0.956 \pm 0.015}$	0.742 ± 0.093	0.628 ± 0.232
2L, 2H, r=0.5 (bal.-trapped)	0.295 ± 0.090	0.217 ± 0.090	0.290 ± 0.116	0.284 ± 0.116
2L, 2H, r=0.9 (16 gen., 14 not)	$\mathbf{0.970 \pm 0.019}$	0.754 ± 0.243	0.812 ± 0.055	$\mathbf{0.855 \pm 0.035}$

Two-layer models at r=0.9 go through a structural transition, with the first robust head appearing at epoch 4420 ± 5140 at 1H and 4785 ± 4939 at 2H. Adding a head does not speed this up; the two timings are indistinguishable given the spread, also when the event is gated on above-chance adversarial accuracy (5212 ± 5250 against 5300 ± 5643; Appendix H). At the generalization epoch 94–100% of *all* heads in two-layer models meet the criterion, which the outcome nearly implies for a model correct on both splits. At the final checkpoint the robust fraction is 0.96 (2H) and 1.00 (1H) among generalizing seeds, against 0.00 and 0.04 among the rest.

5.3 QK/OV Circuit Fingerprint of Shortcut vs. Robust Models

Circuit evolution shows when the transition happens, and the QK/OV fingerprint shows what changes. Because the MESP shortcut is the *parity* of the maximum element, a head that implements it already needs two parts, a query-key circuit that selects the maximum token and an output-value circuit that projects it onto a parity direction. Parity projection is therefore common to both circuit types. The shortcut-trapped 1L, 1H model at r=0.9 reaches an OV parity Spearman of 0.742 at its QK peak (Table 2), so we read the statistic as a sign of maturity. Attention to the shortcut token does not indicate shortcut use either, because the max element and $\text{seq}[0]$ are summands of the true rule. Reliance on the shortcut lies in the readout, which attention weights cannot show. The OV statistic is signed and measures only a head's direct path to the logits (Appendix H).

All models with $r \geq 0.7$ develop a strong shortcut circuit early (Table 2), with QK Spearman ρ peaking between 0.95 and 0.98 whatever the eventual outcome. The models diverge afterwards. In the shortcut-trapped 1L, 1H model at r=0.9, QK ρ stays near its peak (0.956 at the final checkpoint), while OV parity falls back below its value at the peak (0.628 vs. 0.742). In the 2L, 2H model, QK ρ drops from 0.970 to 0.754 by the end of training, pooled over outcomes, and OV parity reaches 0.855 ± 0.035 at generalization, close to the 0.867 ceiling. The max-selecting circuit forms with similar strength in all high-r models, and OV parity develops in failing seeds too (0.739; Appendix H). The drop in final QK ρ comes from the generalizing seeds, whose attention leaves the max token after they generalize, so it is a consequence of the outcome and does not precede it.

At r=0.5 the 2L, 2H model's QK ρ peaks at only 0.295, so no strong max-selecting circuit forms. On both binary tasks, attention leaves the shortcut token only after generalization. On MESP at r=0.9, measured over the whole split at the generalization epoch, the busiest head of a generalizing two-layer model still puts 0.80 (2H) and 0.88 (1H) of its attention on the max token (means over seeds), at least 0.5 in 21 of 23 seeds, and 0.48 and 0.58 averaged over heads. Non-generalizing seeds at epoch 15,000 sit at 0.87 and 0.81 on the busiest head and 0.53 and 0.59 averaged over heads (uniform attention gives $1/7$). By the end of training the generalizing seeds have fallen to 0.26 and 0.16. On FESP the busiest head's $p_{\text{seq}[0]}$ is 0.887 at generalization. Since the model is already perfect when the head withdraws, the simplest reading is that weight decay prunes a redundant head, and the withdrawal is not what enables generalization (Appendices G and B).

One statistic does separate outcomes early, namely *which layer* attends to the max token. In generalizing 2L, 1H models the pinned head is the layer-1 head (median 0.95 at generalization), while the layer-0 head moves off the max token beforehand (0.77 at epoch 100 and 0.27 at generalization). As early as epoch 100, layer-1 minus layer-0 max-token attention is positive in generalizing seeds and negative in failing ones (median $+0.21$ against -0.27 at 2L, 1H, AUC 0.80, p=0.019; $+0.19$ against -0.26 at 2L, 2H, AUC 0.82, p=0.003), and among 2L, 2H seeds not yet generalized the AUC reaches 0.90 by epoch 1,000 (Table 24). We regard this marker as exploratory. It is one of 14 pre-generalization comparisons we report, all on MESP at r=0.9, the 2L, 1H result does not survive a Bonferroni correction over them, and attention weights alone do not show that the shortcut is routed through the last layer (Appendix G).

5.4 Subset Trajectory: Two-Phase Accuracy Dynamics

Tracking accuracy separately on the shortcut-consistent and anti-shortcut subsets splits training into two phases. In Phase 1, common to every configuration at r=0.9, shortcut-consistent accuracy is near perfect by the first logged epoch while anti-shortcut accuracy collapses well below chance. One-layer models stay there for all 15,000 epochs in all but three seeds. Two-layer models enter Phase 2 at a seed-dependent epoch, from a few hundred to over 10,000 (7/30 seeds at 2L, 1H and 16/30 at 2L, 2H; Appendix M), and anti-shortcut accuracy rises toward 1.0. Late escapers memorize first. The 12 seeds escaping at or after epoch 1,900 fit at least half of their anti-shortcut *training* examples a median of 6,450 epochs before escape, and 0.75–1.00 of them just before it, while held-out adversarial accuracy is still at most 0.25 (Appendix G).

6 Discussion

What the force-balance ratio does and does not show. Under empirical risk minimization $\nabla\mathcal{L} = r\,\bar{g}_{\text{SC}} + (1-r)\,\bar{g}_{\text{anti}}$, with $\bar{g}_{\text{SC}}, \bar{g}_{\text{anti}}$ the *per-example mean* subset gradients. Any model that has stopped moving short of zero loss has $r\,\bar{g}_{\text{SC}} + (1-r)\,\bar{g}_{\text{anti}} \approx \mathbf{0}$, forcing $\|\bar{g}_{\text{anti}}\|/\|\bar{g}_{\text{SC}}\| \approx r/(1-r)$ at $\cos\theta \approx -1$. The trapped 1L, 1H models, at $8.96\times$ with $\cos\theta = -0.977$, therefore show only that they have stopped moving and say nothing about amplification. At the first logged epoch every model on every task sits within 0.02 of the loss of a shortcut-only predictor calibrated to r (Appendix D). At that posterior the ratio is $r/(1-r)$ by construction, exactly in logit space and approximately in parameter space, where the Jacobian depends on the input. Under AdamW, stationarity means $m/\sqrt{v} \approx -\lambda\theta$, and a $9\times$ raw-norm ratio is not a $9\times$ larger update. At the shortcut posterior the minority's per-example gradients are larger, but the two subsets' aggregate contributions cancel along the shortcut direction, and that cancellation is the force balance. What can move the model is the uncancelled part, the component of $(1-r)\bar{g}_{\text{anti}}$ orthogonal to $\bar{g}_{\text{SC}}$, about $0.47\,(1-r)\|\bar{g}_{\text{anti}}\|$ at the pre-escape $\cos\theta \approx -0.88$.

Two failure modes. One-layer models are wrong even on the anti-shortcut examples they train on. Two-layer failures at r=0.9 fit those examples (at least 0.97 at 2L, 2H) while scoring 0.097 on held-out ones, and at the null they fit 0.95–0.996 of the training set with held-out accuracy at chance. Failing one-layer models fit only 0.56–0.68, a much weaker, partial memorization (Appendix I). Both depths have ample parameters. The one-layer readout may limit memorization, but it does not explain the 1L, 2H trap (Section 4.1). We cannot explain why two-layer memorizers sit at $9\times$ on the held-out splits (Appendix F).

Why the null ratio fails. The null ratio does not fail because parity is hard. 1L, 1H learns the rule in 10/30 seeds on MESP (p=0.0008 against 0/30 for 2L, 2H), and 2L, 2H learns it in 10/10 on n11. The failing cells combine depth with an uninformative shortcut, and in those cells the models memorize. With $16\times$ the data (8% of sequence space), 2L, 2H generalizes at the null in 10/10 seeds within about 4,000 steps, whereas a same-hardware control run for 16,000 steps generalizes in 0/10 (Appendix K). The failure is therefore data-limited, but whether memorization causes it is still open. More data also pins down the rule more tightly, and late escapers at r=0.9 memorize most of the minority long before they generalize (Section 5.4). We trained only the null ratio at the larger size, so whether informative ratios still help there is unmeasured. Coverage also plausibly explains n11, where the null is not the worst ratio (Appendix C).

Two competing explanations. Two explanations are consistent with these observations. Under *H1, gradient amplification*, once the shortcut posterior is reached, the uncancelled component of the minority's gradient, orthogonal to the shortcut direction, drives a capable model toward the rule. Under *H2, the optimization scaffold*, the cheap shortcut breaks the symmetry of initialization and installs token routing that brings the parity computation within reach (Daniely & Malach, 2020; Abbe et al., 2023). Both predict that training passes *through* a strong shortcut circuit, and both account for the anti-correlated FE-Mod3 cell (Appendix D). The one prediction specific to H1, a larger gradient ratio or uncancelled component in escapers than in trapped seeds at the same epoch, does not hold on the training subsets or at a fixed lead (Appendix F), so H1 has no observational support.

In the transplant experiment (Appendix K), models started from $r{=}0.9$ checkpoints at the shortcut posterior and trained at $r{=}0.5$ generalize in 6/10 seeds, against 0/30 from scratch on TPU ($p \approx 5{\times}10^{-5}$) and 4/10 against 0/10 over the 4,000 epochs a same-hardware control covers ($p{=}0.087$). The shortcut is abandoned within 100 epochs. Because the transplanted model starts at the $r{=}0.9$ posterior with equal group counts, its first updates are dominated by the anti-shortcut examples even more than at $r{=}0.9$, so the transplant removes a *persisting* asymmetry but not the asymmetry itself. The optimizer reset, the cross-hardware comparison and the fact that five of the ten seeds also generalize at $r{=}0.9$ are further caveats (Appendix K). With them, the transplant favors H2. Minority upweighting at $r{=}0.9$ (16/30 on TPU to 1/10 on GPU, $p{=}0.026$) fits both accounts, because it also makes the shortcut uninformative.

Limitations. Our models are one- or two-layer transformers without LayerNorm or biases, trained at one learning rate. Width is not swept within a task, so the "capacity threshold" is a depth threshold, and it inverts on ME-Mod3. The sweeps use $N{=}16{,}384$, and at the larger N we ran only the null ratio. The interventions cover MESP only, at 10 seeds, and are partly compared across hardware. Binary ratios below 0.5 are unswept, and the $15{,}000$-epoch horizon probably censors ME-Mod3. Gradients are raw rather than Adam-preconditioned and, outside MESP at $r{=}0.9$, measured on held-out splits. Transfer to larger scale is untested, as are Group DRO (Sagawa et al., 2020a), subsampling (Idrissi et al., 2022) and JTT (Liu et al., 2021).

7 Conclusion

A spurious correlation can raise the probability that a model finds the robust rule, and what matters is how far r is from the random-chance baseline. The null ratio is the worst two-layer setting on MESP, FESP and FE-Mod3 and, weakly, the lowest on ME-Mod3. Without an informative shortcut two-layer models memorize, but with $16\times$ the data the null ratio no longer fails. Gradient amplification finds no support in our measurements, and a transplant experiment tentatively favors an optimization scaffold. When data are limited, removing a shortcut's predictive power can remove the conditions under which the true rule is found.

AI Use Statement

In this work, we used AI Coding tools (Anthropic's Claude) for three categories of task that require disclosure. First, **writing**: during drafting and revision we used it to generate mock reviews and to edit the main text and appendix in response to them. AI is also used in most Appendices to generate natural language content from numerical results. Second, **research code**: authoring the analysis and reporting scripts `analysis/checkpoint_audit.py`, `analysis/trajectory_audit.py`, `scripts/paper_tables.py`, `scripts/seed_analyses.py`, `scripts/timing_analyses.py` and `scripts/make_paper_figures.py`, which produce the measurements reported in Appendices C, D, F, G, I, J, K and N from stored artifacts, together with corrections to pre-existing analysis code; and the seed-batched training script `scripts/run_owed_jax.py`, which ran the interventions of Appendix K with the original runner's model, initialization and optimizer. Third, **analysis and presentation of results**: generating the LaTeX for several result tables directly from the per-seed output files.

We have reviewed all AI-assisted work. All numerical results in this paper are computed by scripts that run over the stored experimental artifacts, and the commands that regenerate each reported table are named in the corresponding appendix, so every AI-assisted number is checkable

against the experiment results. AI-authored analysis code was reviewed by the authors and cross-checked against independent implementations where both exist—the linear-decodability result of Appendix J, for example, is produced by `checkpoint_audit.py` and reproduced independently by `analysis/residual_probe.py`.

We also report negative and disconfirming results where we have them, including a hypothesis of our own that the data rule out (Appendix D), a circuit-identification criterion that fails calibration in a specific cell (Appendix H), and a class-marginal drift in our data generator that we bound rather than correct (Appendix I).

Reproducibility Statement

The experimental setup—tasks, data generation, architectures, optimizer, schedules and the definition of every reported metric—is specified in Section 3 and Appendix H. Each analysis appendix names the exact command that regenerates its tables from the stored artifacts, and those commands are read-only and CPU-only: Appendix F for the gradient-conflict tables, Appendix J for linear decodability, deep feature reweighting and max-token attention, and Appendix D for the ME-Mod3 hypothesis tests, and Appendix N for the statistical tests and the remaining analyses. Source code for data generation, training and all analyses is included in the supplementary material.

Two caveats affect what can be reproduced. For MESP at $r{=}0.9$ the released artifacts include a checkpoint every 100 epochs, and the weight-based trajectory measurements of Appendices F, G and J use them. For every other configuration they include the best-adversarial-accuracy checkpoint, which breaks ties at 100% by the lowest loss, so for a generalizing run it is typically thousands of epochs after generalization, together with the statistics logged every 100 epochs (accuracies, losses, subset gradients, QK/OV and attention summaries), which carry our trajectory claims for those configurations. And the generalization rates we report are subject to high seed variance, so we report per-seed counts out of 30 rather than point estimates, and a replication should be compared against those counts rather than against a single summary number. Training is deterministic given the seed on a fixed backend but not bitwise identical across backends: the interventions, run on GPU, match the TPU runs in regime and rate but not seed by seed (Appendix K), so they are compared by rate and against a control on the same hardware.

References

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: Leap complexity and saddle-to-saddle dynamics. In *Proceedings of the Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 2552–2623, 2023.

Boaz Barak, Benjamin L Edelman, Surbhi Goel, Sham M Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Amit Daniely and Eran Malach. Learning parities with neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Katherine L. Hermann, Hossein Mobahi, Thomas Fel, and Michael C. Mozer. On the foundations of shortcut learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pp. 336–351. PMLR, 2022.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations*, 2023.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations*, 2023.

Lucas Mansilla, Rodrigo Echeveste, Diego H Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6630–6638, 2021.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Guanwen Qiu, Da Kuang, and Surbhi Goel. Complexity matters: Feature learning in the presence of spurious correlations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41658–41697, 2024.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020a.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020b.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-wise difficulty-balanced loss for solving class-imbalance. In *Asian Conference on Computer Vision*, pp. 549–565. Springer, 2020.

Damien Teney, Liangze Jiang, Florin Gogianu, and Ehsan Abbasnejad. Do we always need the simplicity bias? looking for optimal inductive biases in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 79–90, 2025.

Chenyu You, Haocheng Dai, Yifei Min, Jasjeet S. Sekhon, Sarang Joshi, and James S. Duncan. Uncovering memorization effect in the presence of spurious correlations. *Nature Communications*, 16(1):5424, 2025.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Table 3: Generalization rate by configuration, MESP (weight decay = 0.4). Columns as in Table 1. The main difference from weight decay 0.1 is between the two metrics: 2-layer models reach the robust solution about as often as at weight decay 0.1 but rarely hold it, and 2L, 1H holds it in at most one seed at any ratio.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.5	0/30 (0%)	0/30 (0%)	**1H,** r=0.5	**2/30 (7%)**	**0/30 (0%)**
1H, r=0.7	0/30 (0%)	0/30 (0%)	**1H,** r=0.7	**7/30 (23%)**	**0/30 (0%)**
1H, r=0.9	0/30 (0%)	0/30 (0%)	**1H,** r=0.9	**11/30 (37%)**	**1/30 (3%)**
2H, r=0.5	4/30 (13%)	4/30 (13%)	**2H,** r=0.5	**1/30 (3%)**	**0/30 (0%)**
2H, r=0.7	2/30 (7%)	1/30 (3%)	**2H,** r=0.7	**17/30 (57%)**	**2/30 (7%)**
2H, r=0.9	3/30 (10%)	2/30 (7%)	**2H,** r=0.9	**12/30 (40%)**	**7/30 (23%)**

A Extended Results: Weight Decay = 0.4

Table 1 reports weight decay 0.1. Table 3 reports the corresponding results for weight decay 0.4. The main imbalance $\times$ capacity interaction persists: 2-layer models improve with ratio while 1-layer models do not.

The 2L, 2H model at r=0.9 retains a 40% *anytime* rate under higher weight decay but a 23% terminal rate: as noted in Section 4.1, 5 of those 12 seeds hit the threshold only transiently before regressing to near-shortcut performance, and 7 consolidated. Reading the two columns together also shows the ratio effect under regression: the terminal column still rises with r at 2L, 2H (0%, 7%, 23%), so weight decay suppresses the magnitude of the effect without reversing its direction. The anytime column is higher at r=0.7 (57%) than at r=0.9 (40%), but the difference is not significant (17 vs. 12 of 30, Fisher p=0.30), so we draw no conclusion about where the optimum lies. Both imbalanced ratios remain far above the null (17 and 12 against 1), which is what we claim.

B First Equal Sum Parity

Section 4.3 summarizes the behavioral comparison between the magnitude-shortcut and position-shortcut binary tasks for 2-layer models; this appendix reports the full per-configuration results for all 12 configurations and provides additional mechanistic discussion. While the `max_parity_sum_parity` (MESP) task focuses on a magnitude-based shortcut, the `first_equal_sum_parity` (FESP) task employs a purely position-based shortcut. In the position-shortcut task, the parity of the first element in the sequence (seq[0]) is correlated with the total sum parity in shortcut-consistent examples. Because every input begins with `START`, seq[0] sits at token index 1; "position 0" below always means seq[0], not `START`.

B.1 Contrast in Task Difficulty

As shown in Tables 4 and 5, the position-shortcut task generalizes at least as readily as the magnitude-shortcut task once enough seeds are sampled. At high capacity and high imbalance ($r = 0.9$, 2L, 2H), the generalization rate reaches 77%, against 53% for the magnitude-shortcut task at the same setting. That difference is not significant at 30 seeds (Fisher p=0.10; Appendix N), so we read the two shortcuts as comparably displaceable rather than ranking them. Raising weight decay to 0.4 preserves the same ordering across configurations (Table 5).

The two shortcuts are implemented differently, but attention leaves the shortcut token on the same schedule on both (Section 5.3):

- **Magnitude shortcut (MESP):** Magnitude is often encoded in token embedding norms, so an attention head can implement this shortcut cheaply, and it is fit quickly at high r.
- **Position shortcut (FESP):** The positional shortcut requires absolute positional awareness. On both tasks the busiest head is still on the shortcut token when the model generalizes and leaves it afterwards (Appendix B.3).

Table 4: Generalization rate by configuration for the FESP task (weight decay = 0.1); columns as in Table 1. Two-layer models dominate and generalization rises with ratio (2L, 2H reaches 77% at r=0.9 on both metrics), while 1-layer models remain low—the same imbalance × depth interaction seen in the magnitude-shortcut task. At 2L, 2H, anytime and terminal rates are identical at r=0.7 and r=0.9. This count depends on the 100% threshold more than on MESP: 5 of the 7 non-generalizing 2L, 2H seeds at r=0.9 exceed 99% at some point, and the 17 non-generalizing two-layer seeds at that ratio end at 0.471 held-out adversarial accuracy, so they are near-generalizers that fell back (Appendix N).

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.5	1/30 (3%)	1/30 (3%)	**1H,** r=0.5	**0/30 (0%)**	**0/30 (0%)**
1H, r=0.7	7/30 (23%)	5/30 (17%)	**1H,** r=0.7	**16/30 (53%)**	**13/30 (43%)**
1H, r=0.9	2/30 (7%)	2/30 (7%)	**1H,** r=0.9	**20/30 (67%)**	**15/30 (50%)**
2H, r=0.5	1/30 (3%)	1/30 (3%)	**2H,** r=0.5	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.7	10/30 (33%)	7/30 (23%)	**2H,** r=0.7	**20/30 (67%)**	**20/30 (67%)**
2H, r=0.9	1/30 (3%)	1/30 (3%)	**2H,** r=0.9	**23/30 (77%)**	**23/30 (77%)**

Table 5: Generalization rate by configuration for the FESP task (weight decay = 0.4); columns as in Table 1. As on MESP, higher weight decay preserves the anytime rate and roughly halves the terminal rate for 2L, 2H, and leaves 2L, 1H with almost nothing consolidated.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.5	0/30 (0%)	0/30 (0%)	**1H,** r=0.5	**2/30 (7%)**	**0/30 (0%)**
1H, r=0.7	0/30 (0%)	0/30 (0%)	**1H,** r=0.7	**15/30 (50%)**	**4/30 (13%)**
1H, r=0.9	0/30 (0%)	0/30 (0%)	**1H,** r=0.9	**16/30 (53%)**	**2/30 (7%)**
2H, r=0.5	0/30 (0%)	0/30 (0%)	**2H,** r=0.5	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.7	2/30 (7%)	2/30 (7%)	**2H,** r=0.7	**22/30 (73%)**	**14/30 (47%)**
2H, r=0.9	0/30 (0%)	0/30 (0%)	**2H,** r=0.9	**20/30 (67%)**	**14/30 (47%)**

B.2 The 1-Layer Capacity Threshold and Shortcut-Trapping

As on MESP, whether imbalance helps depends on model capacity. For the position-shortcut task, 1-layer models exhibit a reverse trend compared to their 2-layer counterparts: generalization peaks at the intermediate ratio $r{=}0.7$ and collapses to near-zero at $r{=}0.9$, the opposite of the 2-layer trend where generalization rises with ratio.

Pattern in 1-Layer Models. In 1-layer models, the pattern inverts. Parameter count is not the constraint (Appendix I). For 1L, 1H the readout path is a candidate: the END position sees one attention-weighted average of value vectors per head before a single MLP, so once the only head is pinned to seq[0] at $r{=}0.9$ little else about the sequence reaches the readout. This does not extend to 1L, 2H, where the second head could carry the rest of the sequence; that model is also trapped at $r{=}0.9$ (1/30 on FESP; 3/30 on MESP), so the trap there is dynamical rather than representational (Section 4.1).

Adversarial Accuracy as a Measure of Capture. The near-zero adversarial accuracy observed in 1-layer, high-ratio models (e.g., 0.044 for 1L, 2H at $r = 0.9$) is the clearest evidence of this "shortcut-trapping". Because the adversarial set is constructed so that the shortcut contradicts the true label, a model that has perfectly learned the shortcut will consistently predict the wrong parity, resulting in near-0% accuracy. This is consistent with the following interpretation.

- There is a threshold—in this case, between one and two transformer layers—below which imbalance is harmful because training converges onto the easiest feature.
- Above this threshold, two escapes from the shortcut are open: reorganizing onto the true rule, or memorizing the anti-shortcut training examples (Appendix I).

B.3 Positional Shortcut Fingerprint

Section 5.3 summarizes the positional fingerprint; this subsection gives it in full. Because the shortcut is positional rather than magnitude-based, QK Spearman ρ is not the appropriate fingerprint: the shortcut circuit need not rank tokens by magnitude, only concentrate attention on seq[0]. We therefore use $p_{\text{seq[0]}}$, the attention the final (END) position places on seq[0], taken as the maximum over heads and layers on the logged probe sequence of the shortcut-consistent split; Table 6 reports it.

Every high-r configuration has a strong shortcut head, including the one-layer ones: peak $p_{\text{seq[0]}}$ is 0.982 at 1L, 1H and 0.991 at 2L, 2H ($r{=}0.9$). With no causal mask and readout from END alone, information at seq[0] can reach the readout of a one-layer model only through attention, so this head is how the one-layer models implement the shortcut. In generalizing 2L, 2H models the shortcut head is still at 0.887 at the generalization epoch and falls only afterwards, so the robust rule is in place before the head is withdrawn. MESP follows the same schedule (Section 5.3, Appendix G).

Strong seq[0] attention also appears at the null ratio, where no seed generalizes: 0.729 at peak and 0.571 at the end for 2L, 2H. It is not specific to seq[0]. At the end of training the same statistic for the most-attended *other* integer position is 0.976 ± 0.040, so at the null the busiest head pins some single position with nearly all its weight, and seq[0] is merely one of the positions it may choose. This is consistent with memorization keyed on individual tokens, though we have not tested that. Either way, attention to seq[0] is not a shortcut indicator on its own. OV parity is the final-checkpoint value, as the statistic was not logged at the generalization epoch for this task; we give it signed and as $|\rho|$ (Appendix H). The signed value's large spread in the 1L, 1H $r{=}0.9$ row (0.597 ± 0.618) is a sign artifact: $|\rho|$ is 0.847 ± 0.093. We do not use whole-model attention ablation here: zeroing all attention makes the END residual input-independent, so the resulting change in adversarial accuracy measures collapse to a constant prediction, not the role of attention.

Table 6: FESP positional shortcut fingerprint (wd = 0.1). $p_{\text{seq}[0]}$ is the maximum over heads of the attention from END to seq[0] on the logged shortcut-consistent probe sequence. "Peak" is the maximum over logged epochs; "at gen." is at the last logged epoch at or before the first 100% adversarial epoch, over generalizing seeds only (seed count in parentheses); "final" is the last logged epoch over all 30 seeds. † marks the one configuration with no generalizing seed. OV parity is at the final checkpoint, max over heads, signed and absolute. ‡ is MESP 2L, 2H at r=0.9 under the identical statistic, with attention to the max-valued token in place of seq[0].

Config	$p_{\text{seq}[0]}$ peak	at gen.	final	OV parity	|OV parity|
1L, 1H, r=0.5	0.135 ± 0.273	0.215 (1)	0.075 ± 0.190	-0.055 ± 0.259	0.184 ± 0.188
1L, 1H, r=0.9	0.982 ± 0.032	0.608 ± 0.054 (2)	0.661 ± 0.305	0.597 ± 0.618	0.847 ± 0.093
2L, 2H, r=0.5†	0.729 ± 0.315	—	0.571 ± 0.365	0.131 ± 0.142	0.282 ± 0.112
2L, 2H, r=0.9	$\mathbf{0.991 \pm 0.048}$	$\mathbf{0.887 \pm 0.219}$ (23)	0.391 ± 0.270	0.813 ± 0.317	0.873 ± 0.010
MESP‡	1.000 ± 0.001	0.813 ± 0.332 (16)	0.590 ± 0.400	—	—

C N11 VARIANT REPLICATION

The max_parity_sum_parity_n11 variant uses integers drawn from $[0, 11)$ instead of $[0, 20)$, producing a lower-variance max-element distribution. This makes the sum parity task structurally easier (fewer distinct values, shorter effective range). Tables 7 and 8 report generalization rates for weight decay 0.1 and 0.4 respectively; all per-seed counts are out of 10 seeds.

Table 7: n11 variant: generalization rate (weight decay = 0.1); columns as in Table 1, out of 10 seeds. Two-layer models are at or near ceiling on both metrics at every ratio, including the balanced one—which is the evidence that the capacity threshold sits below two layers on this easier task.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.5	5/10 (50%)	5/10 (50%)	**1H,** r=0.5	**7/10 (70%)**	**7/10 (70%)**
1H, r=0.7	8/10 (80%)	6/10 (60%)	**1H,** r=0.7	**9/10 (90%)**	**9/10 (90%)**
1H, r=0.9	1/10 (10%)	1/10 (10%)	**1H,** r=0.9	**10/10 (100%)**	**9/10 (90%)**
2H, r=0.5	7/10 (70%)	7/10 (70%)	**2H,** r=0.5	**10/10 (100%)**	**10/10 (100%)**
2H, r=0.7	7/10 (70%)	5/10 (50%)	**2H,** r=0.7	**10/10 (100%)**	**8/10 (80%)**
2H, r=0.9	4/10 (40%)	4/10 (40%)	**2H,** r=0.9	**10/10 (100%)**	**10/10 (100%)**

The same configurations under weight decay 0.4 are reported in Table 8.

Table 8: n11 variant: generalization rate (weight decay = 0.4); columns as in Table 1, out of 10 seeds. The same anytime/terminal divergence as on the base task appears here: 2L, 1H reaches 100% in 8/10 seeds at r=0.9 and holds it in 1.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.5	0/10 (0%)	0/10 (0%)	**1H,** r=0.5	**8/10 (80%)**	**5/10 (50%)**
1H, r=0.7	0/10 (0%)	0/10 (0%)	**1H,** r=0.7	**6/10 (60%)**	**2/10 (20%)**
1H, r=0.9	0/10 (0%)	0/10 (0%)	**1H,** r=0.9	**8/10 (80%)**	**1/10 (10%)**
2H, r=0.5	5/10 (50%)	5/10 (50%)	**2H,** r=0.5	**10/10 (100%)**	**5/10 (50%)**
2H, r=0.7	6/10 (60%)	5/10 (50%)	**2H,** r=0.7	**8/10 (80%)**	**4/10 (40%)**
2H, r=0.9	0/10 (0%)	0/10 (0%)	**2H,** r=0.9	**8/10 (80%)**	**4/10 (40%)**

One finding replicates in the n11 variant: the 1L, 1H model is trapped by high imbalance (r=0.9, 10% generalization rate) despite strong generalization at r=0.7 (80%), mirroring the base task's 1-layer shortcut trap. The two-layer ratio benefit does *not* replicate, because 2-layer models are at or near ceiling at every ratio; in particular 2L, 2H generalizes in 10/10 seeds at r=0.5, so on this task the null ratio is not the worst setting for two-layer models. n11 is therefore a counterexample to that

claim, not a replication of it. With 10 seeds per configuration this variant is a check rather than a second primary result.

The n11 task sits at a lower overall difficulty: the 2L, 2H model achieves 100% generalization even at balanced ratios (r=0.5), which does not occur in the base task. This suggests the capacity threshold for n11 falls below the 2L, 2H architecture. The location of the threshold thus appears to depend on the difficulty of the true rule relative to the shortcut.

Memorization on n11. If depth-enabled memorization were what blocks the rule at the null, n11 two-layer models should memorize too. They do not get the chance. The 10 generalizing 2L, 2H seeds at r=0.5 reach 0.99 training accuracy at a median of 250 epochs and 100% adversarial accuracy at a median of 300; training accuracy never runs more than 0.08 ± 0.14 ahead of held-out accuracy beforehand, so rule and fit arrive together. The three non-generalizing 2L, 1H seeds at the null fit 0.83 of their training set with held-out adversarial accuracy at 0.50, i.e. partial memorization, and the non-generalizing one-layer seeds fit 0.57–0.63. Two things distinguish n11 from the base task: the rule is easier, and the same 16,384 training examples cover 10.2% of the 11^5 possible sequences against 0.5% of 20^5. Coverage is sufficient on the base task: a training set covering 8% of sequence space rescues the null ratio in 10/10 seeds, again with rule and fit arriving together (Appendix K). This is the likely reason n11 is an exception to the null ratio being the worst setting. Regenerated by `python scripts/timing_analyses.py n11`.

D MOD 3 LABEL TASKS

The two mod 3 tasks use ternary labels (3 classes) instead of binary, so the random-chance shortcut correlation is $1/3$ rather than $1/2$. The ratio sweep is $r \in \{0.15, 0.33, 0.50\}$, where r=0.33 $\approx 1/3$ is the null point and r=0.15 and r=0.50 are imbalanced in opposite directions.

D.1 FIRST-EQUAL-SUM-MOD3 (FE-MOD3): POSITIONAL SHORTCUT, TERNARY LABEL

Table 9: Generalization rates for the FE-Mod3 task (weight decay = 0.1); columns as in Table 1. At the null ratio r=0.33 (random-chance shortcut correlation), **no 2-layer model generalizes**, on either metric. Both r=0.15 (anti-biased) and r=0.50 (biased) reach 80–93% anytime in 2-layer models, indicating that the relevant quantity is deviation from the random-chance baseline, not the direction of the imbalance. Terminal rates are markedly lower (27–67%), so the ternary tasks lose more seeds to post-generalization regression than the binary ones; the null ratio remains the worst setting under either metric.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.15	7/30 (23%)	4/30 (13%)	**1H,** r=0.15	**25/30 (83%)**	**8/30 (27%)**
1H, r=0.33	11/30 (37%)	2/30 (7%)	**1H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
1H, r=0.5	11/30 (37%)	10/30 (33%)	**1H,** r=0.5	**24/30 (80%)**	**20/30 (67%)**
2H, r=0.15	9/30 (30%)	8/30 (27%)	**2H,** r=0.15	**28/30 (93%)**	**20/30 (67%)**
2H, r=0.33	0/30 (0%)	0/30 (0%)	**2H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.5	2/30 (7%)	2/30 (7%)	**2H,** r=0.5	**27/30 (90%)**	**17/30 (57%)**

The corresponding results at weight decay 0.4 are given in Table 10.

Table 9 reveals a pronounced U-shaped dependence in 2-layer models: 0% of 2-layer seeds generalize at the null r=0.33, against 80–93% at both r=0.15 (where 85% of training examples are anti-shortcut) and r=0.50 (where 50% are shortcut-consistent). One-layer models show no such U-shape: 1L, 1H generalizes at similar rates across all three ratios, including the null (23–37%). At the null, failing two-layer models fit 0.996 of their training set while held-out accuracy sits at chance (Appendix I).

What saturates at r=0.15? Nothing, in the sense of the majority subset reaching near-zero loss. The informative question is what state models occupy before they generalize, and the logged subset losses answer it precisely. A predictor that uses only the shortcut and is calibrated to the training

Table 10: Generalization rates for the FE-Mod3 task (weight decay = 0.4); columns as in Table 1. The U-shaped pattern in 2-layer models persists on the anytime metric, though attenuated. Terminal rates nearly vanish (0–17%), matching the binary tasks: at this weight decay models reach the robust solution but do not keep it.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.15	1/30 (3%)	0/30 (0%)	**1H,** r=0.15	**11/30 (37%)**	**0/30 (0%)**
1H, r=0.33	3/30 (10%)	2/30 (7%)	**1H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
1H, r=0.5	9/30 (30%)	5/30 (17%)	**1H,** r=0.5	**12/30 (40%)**	**1/30 (3%)**
2H, r=0.15	3/30 (10%)	1/30 (3%)	**2H,** r=0.15	**17/30 (57%)**	**5/30 (17%)**
2H, r=0.33	0/30 (0%)	0/30 (0%)	**2H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.5	5/30 (17%)	4/30 (13%)	**2H,** r=0.5	**18/30 (60%)**	**2/30 (7%)**

ratio assigns probability r to the shortcut's class and $(1-r)/(k-1)$ to each of the other $k-1$ classes, so its loss is $-\ln r$ on shortcut-consistent examples and $-\ln \frac{1-r}{k-1}$ on anti-shortcut ones. Table 11 compares these floors with the measured held-out losses at the first logged epoch. Every configuration on every task sits on them, to within 0.02: at the first log every model has learned exactly the shortcut-only posterior, whatever it later does. At FE-Mod3 r=0.15 that posterior only *excludes* the shortcut class, so the anti-shortcut majority sits at 0.50 accuracy and loss 0.86, above $\log 2$, and the shortcut-consistent minority at loss 1.90. The best cell in the paper (2L, 2H, 28/30) generalizes from there, at a median of 400 epochs (first 100% adversarial epoch; 16 of the 28 by epoch 600, the rest between 3,900 and 14,600; Table 28). Two consequences follow. First, saturation of a majority to near-zero loss is not necessary for the effect, and even at binary r=0.9 the majority's floor is $-\ln 0.9 = 0.105$, not zero. Second, at the shortcut-only posterior the mean per-example loss gradient with respect to the logits of the minority exceeds the majority's by exactly the ratio of their counts, which is the $r/(1-r)$ force balance of Section 6 (and its inverse, 0.176, at r=0.15, where the shortcut-consistent examples are the minority). The identity is exact in logit space only; in parameter space it is approximate, because the Jacobian depends on the input and the two subsets differ in their input distribution. At the null ratios the calibrated shortcut-only posterior is the uniform predictor, so sitting on that floor means only that nothing has been learned yet. The gradient asymmetry we measured is therefore a property of having learned the shortcut, present from the first logged epoch in generalizing and trapped models alike. At that posterior the two subsets' aggregate contributions cancel along the shortcut direction, so the minority does not dominate the update there; what can move the model is the uncancelled part, the component of $(1-r)\bar{g}_{\text{anti}}$ orthogonal to $\bar{g}_{\text{SC}}$. This is the form of H1 we state in Section 6, and it applies in both directions of r: it covers the anti-correlated cell as well as the binary ones, so that cell does not discriminate between H1 and H2.

Across all four tasks, what matters appears to be the distance of r from the random-chance baseline ($1/2$ for binary tasks, $1/3$ for ternary tasks), not the absolute value of r. At the baseline the shortcut-only posterior is uninformative and supplies no partial fit, and it is there that two-layer models memorize; whether the memorization is a cause of the failure or a consequence is open (Section 6).

D.2 Max-Equal-Sum-Mod3 (ME-Mod3): Magnitude Shortcut, Ternary Label

The corresponding results at weight decay 0.4 are given in Table 13.

The magnitude-shortcut ternary task (ME-Mod3) shows a qualitatively similar but substantially weaker pattern (Tables 12 and 13). The r=0.33 null continues to suppress generalization in 2-layer models. However, the imbalanced ratios yield only 17–27% generalization for 2L, 2H—far below the 90–93% reached in the ternary position-shortcut task, and below the 53% of the binary magnitude-shortcut task. Two factors likely contribute. First, computing sum mod 3 requires distinguishing three residue classes across five elements, a harder aggregation than sum parity. Second, informative-ratio generalization on this task is late (median $12{,}850$ epochs at two layers, interquartile range $11{,}775$–$14{,}200$; Table 28) against a $15{,}000$-epoch horizon, so its counts are probably right-censored. A longer horizon could raise both the informative and the null counts, and the ME-Mod3 ratio comparison should be read with that in mind wherever it is counted among the four primary tasks.

Table 11: Held-out subset losses at the first logged epoch (100) against the loss of a shortcut-only predictor calibrated to the training ratio: $-\ln r$ on shortcut-consistent (SC) and $-\ln \frac{1-r}{k-1}$ on anti-shortcut (anti) examples, k the number of classes. "Measured" values are medians over 30 seeds of 2L, 2H at weight decay 0.1; the last column is the largest deviation from either floor over all four depth/head configurations (medians per configuration). At the null ratios the floor is the uniform predictor.

Task	r	SC floor	SC measured	anti floor	anti measured	Max dev. (all)
MESP	0.5	0.693	0.695	0.693	0.699	0.006
MESP	0.7	0.357	0.357	1.204	1.219	0.015
MESP	0.9	0.105	0.107	2.303	2.308	0.011
FESP	0.5	0.693	0.700	0.693	0.692	0.007
FESP	0.7	0.357	0.357	1.204	1.211	0.007
FESP	0.9	0.105	0.107	2.303	2.322	0.019
FE-Mod3	0.15	1.897	1.907	0.856	0.861	0.014
FE-Mod3	0.33	1.109	1.119	1.094	1.104	0.010
FE-Mod3	0.50	0.693	0.698	1.386	1.389	0.007
ME-Mod3	0.15	1.897	1.908	0.856	0.873	0.018
ME-Mod3	0.33	1.109	1.106	1.094	1.106	0.012
ME-Mod3	0.50	0.693	0.708	1.386	1.401	0.015

Table 12: Generalization rates for the ME-Mod3 task (weight decay = 0.1); columns as in Table 1. The effect is present but weaker than on the ternary position-shortcut and binary magnitude-shortcut tasks, and the depth ordering inverts: 1L, 1H at r=0.50 is the best cell in the table on both metrics (50% anytime, 47% terminal), above every 2-layer cell.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.15	8/30 (27%)	6/30 (20%)	**1H,** r=0.15	**3/30 (10%)**	**3/30 (10%)**
1H, r=0.33	2/30 (7%)	1/30 (3%)	**1H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
1H, r=0.5	15/30 (50%)	14/30 (47%)	**1H,** r=0.5	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.15	0/30 (0%)	0/30 (0%)	**2H,** r=0.15	**8/30 (27%)**	**8/30 (27%)**
2H, r=0.33	0/30 (0%)	0/30 (0%)	**2H,** r=0.33	**1/30 (3%)**	**0/30 (0%)**
2H, r=0.5	2/30 (7%)	1/30 (3%)	**2H,** r=0.5	**5/30 (17%)**	**1/30 (3%)**

Table 13: Generalization rates for the ME-Mod3 task (weight decay = 0.4). The effect remains weak; the r=0.33 null continues to suppress generalization.

1-layer models			2-layer models		
Config	Anytime	Terminal	Config	Anytime	Terminal
1H, r=0.15	0/30 (0%)	0/30 (0%)	**1H,** r=0.15	**0/30 (0%)**	**0/30 (0%)**
1H, r=0.33	1/30 (3%)	0/30 (0%)	**1H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
1H, r=0.5	11/30 (37%)	5/30 (17%)	**1H,** r=0.5	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.15	0/30 (0%)	0/30 (0%)	**2H,** r=0.15	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.33	0/30 (0%)	0/30 (0%)	**2H,** r=0.33	**0/30 (0%)**	**0/30 (0%)**
2H, r=0.5	1/30 (3%)	0/30 (0%)	**2H,** r=0.5	**1/30 (3%)**	**0/30 (0%)**

The capacity ordering inverts on this task. ME-Mod3 is the one task in our set where depth does not help, which limits the claim in Section 4.1. The 1L, 1H model at r=0.50 generalizes in 15/30 seeds (50%)—the highest rate anywhere in Table 12, and above *every* 2-layer cell, the best of which reaches 8/30 (27%). The 2L, 1H model generalizes in 0/30 seeds at the null ratio (r=0.33) and 0/30 at the shortcut-biased ratio (r=0.50), reaching only 3/30 (10%) at the anti-biased ratio (r=0.15), while its 1-layer counterpart reaches 50% at r=0.50. The same ordering holds at weight decay 0.4 (Table 13), where 1L, 1H at r=0.50 reaches 11/30 and no 2-layer cell exceeds 1/30, so this is not a single-hyperparameter artifact.

What survives on this task is the ratio effect rather than the capacity effect: the 1L, 1H row is 27%, 7%, 50% across $r \in \{0.15, 0.33, 0.50\}$, still dipping at the null. We do not have a mechanistic account of why depth reverses here, and we flag it as the clearest limit on the capacity threshold reported in Section 4.1. Since this bears on the capacity argument of Section 6, we consider three hypotheses, each with a measurement that would test it.

(i) The shortcut posterior is learned poorly on this task. Every account we consider starts from the shortcut-only posterior, and ranking tokens by magnitude does not cleanly determine max $\bmod 3$. If the models reach that posterior poorly here, the gradient norm ratio at the imbalanced ratios should stay far below the $r/(1-r)$ value the posterior implies, in contrast to the 8.65–8.96× against a predicted 9× that the binary task shows (Appendix F).

The measurement does not support the hypothesis. Table 14 gives all twelve ME-Mod3 configurations at weight decay 0.1. Every cell sits *above* its force-balance value rather than below it—by 1.10–1.96× on the mean, with medians spanning 0.96–1.95× and only one cell (1L, 1H at r=0.50) marginally under its equilibrium. The shortcut-trapped one-layer cells of the binary task, by contrast, sit essentially *on* theirs (0.96–1.02×; Table 17). Measured against its own equilibrium, ME-Mod3 therefore carries *more* gradient asymmetry than MESP, not less, and we consider (i) ruled out.

The end-of-training values in the table are consequences of where each run ended rather than predictors of its outcome: on MESP a generalized model ends with low ratio and positive cosine because it is correct on both subsets (Section 5.1). On ME-Mod3 the two-layer cells end above their equilibrium by the largest margins in the table (1.66–1.96× at $r \in \{0.15, 0.33\}$) and near-orthogonal (-0.10 to $+0.17$) while generalizing in only 0–8 of 30 seeds, consistent with the failing two-layer models having memorized their training set (Table 29), which also moves both statistics. The inversion is left without a gradient-level explanation.

Table 14: Gradient norm ratio against the force-balance prediction, ME-Mod3 at weight decay 0.1, 30 seeds per cell. "Predicted" is the $r/(1-r)$ equilibrium of Section 6; "obs./pred." divides the observed mean by it. Hypothesis (i) predicts this column far *below* 1.00; every cell instead lies above it. "Final $\cos\theta$" is the end-of-training cosine similarity between the two subset gradients, and "gen." the number of seeds that generalize.

Config	r	Predicted	Observed mean	Median	obs./pred.	Final $\cos\theta$	Gen.
1L, 1H	0.15	0.176	0.234 ± 0.065	0.223	1.33×	-0.632 ± 0.250	8/30
1L, 1H	0.33	0.493	0.773 ± 0.135	0.777	1.57×	-0.129 ± 0.334	2/30
1L, 1H	0.50	1.000	1.136 ± 1.055	0.959	1.14×	-0.483 ± 0.568	15/30
1L, 2H	0.15	0.176	0.257 ± 0.024	0.252	1.46×	-0.490 ± 0.124	0/30
1L, 2H	0.33	0.493	0.802 ± 0.129	0.789	1.63×	-0.235 ± 0.151	0/30
1L, 2H	0.50	1.000	1.100 ± 0.109	1.084	1.10×	-0.717 ± 0.271	2/30
2L, 1H	0.15	0.176	0.346 ± 0.067	0.345	1.96×	-0.052 ± 0.133	3/30
2L, 1H	0.33	0.493	0.818 ± 0.077	0.806	1.66×	$+0.092 \pm 0.122$	0/30
2L, 1H	0.50	1.000	1.249 ± 0.078	1.260	1.25×	-0.468 ± 0.078	0/30
2L, 2H	0.15	0.176	0.330 ± 0.118	0.316	1.87×	-0.097 ± 0.270	8/30
2L, 2H	0.33	0.493	0.886 ± 0.120	0.863	1.80×	$+0.172 \pm 0.261$	1/30
2L, 2H	0.50	1.000	1.321 ± 0.159	1.287	1.32×	-0.415 ± 0.240	5/30

(ii) Depth costs more than it supplies when the true rule needs three-way composition. Sum mod 3 requires carrying a three-state residue across five positions. Composing that through two attention-plus-MLP blocks admits more ways to arrive at a partially correct intermediate representation that the second block cannot repair, so the extra routing flexibility that helps on the binary tasks may instead expand the set of saddles and rank-deficient intermediates. If so, the two-layer failures on this task

should look different from the two-layer failures at $r{=}0.33$: trapped rather than never-started, with a strong QK fingerprint that is never displaced.

The logged QK trajectories test this directly. Non-generalizing two-layer seeds at the informative ratios form a strong max-selecting circuit and keep it (peak QK ρ 0.970–0.975 at $r{=}0.15$ and 0.831–0.844 at $r{=}0.50$, final 0.956–0.961 and 0.816–0.824), whereas at the null none forms (peak 0.30–0.41). So the prediction holds, but it does not discriminate: non-generalizing *one*-layer seeds show the same trapped-versus-never-started split (peak 0.972 and 0.848 against 0.158 at 1L, 1H), so it cannot by itself explain why depth hurts here. What does separate the depths is that failing two-layer models fit their training set at every ratio on this task (final training accuracy 0.971, 0.992 and 0.994 at $r{=}0.15$, 0.33, 0.50, pooled over head counts), while failing one-layer models do not (0.579, 0.560, 0.617).

(iii) The one-layer advantage is about the readout, not the circuit. With three classes and—on this task—$d_{\text{model}}{=}128$, the one-layer model's residual stream feeds the output projection directly, and sum mod 3 may be more nearly linearly decodable from a single aggregation step than from a composed one. This predicts that a linear probe for the true label should be *more* accurate in the one-layer model than in the two-layer model on this task, reversing the ordering we would expect on the binary tasks.

This hypothesis is partly supported (Table 15). We compare non-generalizing seeds only, since generalizing seeds probe at 1.000 by construction and would turn the column into a restatement of the generalization rate. At one head the one-layer models decode the true label better than the two-layer models, by $+0.114$ at $r{=}0.50$ ($p < 0.001$) and $+0.035$ at $r{=}0.15$ ($p{=}0.011$), with no difference at the $r{=}0.33$ null. At two heads it is small: absent at $r{=}0.15$ and 0.33, but $+0.013$ ($p < 0.001$) at $r{=}0.50$, a cell where the generalization ordering is *not* inverted. This restriction matters because the depth inversion itself only occurs at one head: at one head the one-layer model generalizes 15/30 against 0/30 at $r{=}0.50$ and 8/30 against 3/30 at $r{=}0.15$, whereas at two heads the ordering is the ordinary one (2/30 against 5/30, and 0/30 against 8/30). The probe difference is largest where the generalization rates invert ($+0.114$ and $+0.035$), but it is also present, much smaller, in one cell where they do not ($+0.013$ at 2H, $r{=}0.50$), so it does not isolate the inversion as cleanly as (iii) would require; we read it as weak support.

Two caveats on that comparison. It conditions on failure, so at $r{=}0.50$ the surviving one-layer seeds are the worst 15 of 30 measured against all 30 two-layer seeds; selection therefore works against the effect we report rather than for it. The predicted reversal relative to the binary task holds only weakly: on MESP at $r{=}0.9$ the two-layer models decode better at one head (0.518 against 0.484), and the two depths are level at two heads (0.497 against 0.496; Table 30).

The failures on this task also differ in kind from those on MESP. On MESP the true label is essentially absent from failing models: every non-generalizing cell at $r \leq 0.7$ probes within 0.010 of chance. On ME-Mod3 the failing models carry noticeably more of it. Normalizing as $(\text{acc} - \text{chance})/(1 - \text{chance})$ so that a three-way probe can be set beside a two-way one, the ME-Mod3 non-generalizing cells span 0.06–0.38 against -0.03–0.04 for MESP, and every one of the twelve ME-Mod3 cells clears the MESP median of 0.001. The ternary task's failures look less like "the rule never formed" and more like "the rule formed in part and went unused." This is the failure mode (iii) describes, and the only one under which a readout-limited account makes sense.

All three hypotheses were testable from the stored artifacts: (i) is ruled out, (ii)'s prediction holds but does not separate the depths, and (iii) receives weak support, strongest at the head count where the inversion occurs. Both are single commands against the released artifacts—`python scripts/paper_tables.py gradient -task max_mod3_sum_mod3` and `python -m analysis.checkpoint_audit probe -task max_mod3_sum_mod3`. The (ii) test reads the logged QK trajectories. The depth inversion on this task remains without a mechanistic explanation. Memorization separates the depths here, but whether it causes failure or accompanies it is open on every task (Section 6).

Table 15: True-label linear decodability by depth, ME-Mod3 at weight decay 0.1, *non-generalizing seeds only*; generalizing seeds probe at 1.000 by construction, so including them would make the column a restatement of the generalization rate. Chance is 0.333. p is a two-sided Mann–Whitney U test between the one- and two-layer seeds at the same head count and ratio. Hypothesis (iii) predicts 1L $>$ 2L: it holds at one head, where the depth inversion is, and at two heads is absent except for a small effect at r=0.50, where the inversion is absent. The final column gives the generalization counts that define the inversion.

Heads	r	1L probe	2L probe	Difference	p	Gen. 1L vs 2L
1H	0.15	0.528 ± 0.088 (22)	0.494 ± 0.003 (27)	$+\mathbf{0.035}$	0.011	8 vs 3
1H	0.33	0.377 ± 0.020 (28)	0.381 ± 0.018 (30)	-0.004	0.234	2 vs 0
1H	0.50	0.584 ± 0.199 (15)	0.470 ± 0.010 (30)	$+\mathbf{0.114}$	< 0.001	15 vs 0
2H	0.15	0.491 ± 0.006 (30)	0.491 ± 0.005 (22)	-0.000	0.926	0 vs 8
2H	0.33	0.374 ± 0.013 (30)	0.382 ± 0.015 (29)	-0.008	0.097	0 vs 1
2H	0.50	0.479 ± 0.007 (28)	0.466 ± 0.026 (25)	$+\mathbf{0.013}$	< 0.001	2 vs 5

E Circuit Formation Timing

Section 5.2 quotes the first-robust-head epochs; Table 16 gives all twelve configurations.

Table 16: Epoch of first robust head formation, wd=0.1. Each mean $\pm$ std is over only the seeds that ever form a robust head; the trailing fraction gives that count out of 30, so cells with small counts are conditional on a rare event. One-layer, one-head models form one rarely and very late (12850 ± 919, 2/30 at r=0.7) or never (r=0.9). The two-layer r=0.9 row compares 1H (10/30) against 2H (20/30), and neither the raw criterion (4420 against 4785 epochs) nor the event gated on above-chance adversarial accuracy (5212 against 5300) separates the two head counts given the spread (Appendix H). At 1L, 2H, r=0.9 formation and outcome coincide: a robust head forms in 3/30 seeds, the same three that generalize. Appendix H calibrates the criterion, which is not specific at balanced ratios.

1-layer models		2-layer models	
Config	First robust head n/30	Config	First robust head n/30
1H, r=0.5	4990 ± 4060 10/30	1H, r=0.5	4100 ± 4068 9/30
1H, r=0.7	12850 ± 919 2/30	1H, r=0.7	2988 ± 3969 8/30
1H, r=0.9	— (never) 0/30	1H, r=0.9	4420 ± 5140 10/30
2H, r=0.5	4775 ± 6186 4/30	2H, r=0.5	4291 ± 4154 23/30
2H, r=0.7	567 ± 321 3/30	2H, r=0.7	1464 ± 3384 14/30
2H, r=0.9	2033 ± 2501 3/30	2H, r=0.9	4785 ± 4939 20/30

F Gradient Dynamics: All Configurations

Reading the median-epoch markers in Figure 1. The dotted vertical lines are medians over generalizing seeds only. Because fewer than half the runs generalize in every one of the six depth $\times$ ratio groups, the unconditional median is undefined throughout. Even the largest group, 2-layer r=0.9, reaches only 23/60; the others rest on 13, 5, 3, 3 and 21 of 60 seeds (1-layer r=0.5, 0.7, 0.9; 2-layer r=0.5, 0.7), so every marker locates the median of a minority, and the three smallest should not be read as group timings.

Section 5.1 quotes gradient statistics per seed and per configuration without tabulating them; Table 17 gives all twelve configurations. Medians are included alongside means because the gradient ratio is heavy-tailed at r=0.9: the 2L, 1H mean of 30.083 carries a standard deviation of 125, and its median of 7.900 is the more representative summary. Pooled over outcomes the two-layer r=0.9 cells are therefore uninformative about the force-balance prediction; Table 18 splits them by outcome.

Split by outcome, and before escape. Table 18 splits every configuration that contains both outcomes. In all ten, generalizing seeds end with higher cosine similarity than non-generalizing

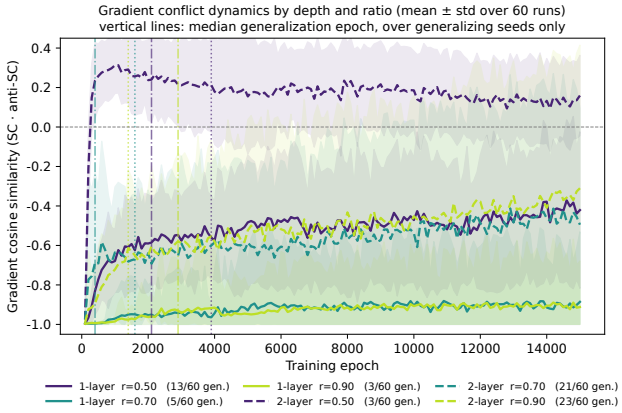


Figure 1: Gradient cosine similarity between shortcut-consistent and anti-shortcut gradient vectors over training. Each curve is one depth $\times$ ratio group at weight decay 0.1, pooling both head counts (mean $\pm$ std over 60 runs, band clipped to $[-1, 1]$). Dotted lines mark each group's median generalization epoch over *generalizing seeds only*, a minority in every group (Appendix F). Conflict persists in 1-layer models at $r \geq 0.7$ (both curves end near -0.9) while 2-layer models move toward zero (-0.31 at r=0.9, pooling a trapped subpopulation near -0.8 with a generalizing one above zero). Proximity to zero is not sufficient: the 2-layer r=0.5 curve ends highest of the six (0.16) with a 0% generalization rate at 2H (Table 1).

Table 17: Gradient conflict dynamics, all 12 configurations, MESP at weight decay 0.1, 30 seeds each. "Peak conflict epoch" is the epoch of minimum cosine similarity. "Gradient ratio" is the anti-shortcut to shortcut-consistent gradient norm ratio at the end of training. Entries without a standard deviation have zero variance across seeds at the reported precision. "Resolved ep." is the conflict-resolution epoch defined in Appendix H—the first epoch at which cosine similarity exceeds -0.1—with the count of seeds that ever reach it. It separates the two failure modes sharply: at r=0.5 the two-layer models resolve conflict in 30/30 seeds within the first few hundred epochs and still never generalize, while the shortcut-trapped 1-layer models at r=0.9 resolve it in 0/30. The force-balance prediction $r/(1-r)$ of Section 6 is 1.00, 2.33 and 9.00 at r=0.5, 0.7 and 0.9; the 1-layer cells at $r \geq 0.7$ sit within 6% of it on the median, while 1L, 1H at r=0.5 is 14% below (0.857) and the pooled two-layer cells at r=0.9 are 12% and 3% below (7.90 and 8.77). The pooled two-layer figures average a trapped subpopulation sitting on the equilibrium with an escaped one well below it, so they should be read against the outcome-conditioned values in Table 18 rather than on their own.

Config	Peak conflict ep.	Min cos.	Final cos.	Grad. ratio	Median	Resolved ep. n/30
1L, 1H, r=0.5	1070 ± 3149	-0.998 ± 0.002	-0.458 ± 0.422	0.996 ± 0.522	0.857	5412 ± 3978 16/30
1L, 1H, r=0.7	287 ± 168	-0.999	-0.953 ± 0.137	2.366 ± 0.429	2.327	14700 1/30
1L, 1H, r=0.9	263 ± 188	-0.998 ± 0.001	-0.977 ± 0.009	8.957 ± 0.616	8.995	— 0/30
1L, 2H, r=0.5	510 ± 2246	-0.995 ± 0.003	-0.384 ± 0.373	1.014 ± 0.266	0.994	4452 ± 2891 21/30
1L, 2H, r=0.7	163 ± 93	-0.998 ± 0.001	-0.818 ± 0.477	2.264 ± 0.276	2.273	2033 ± 1457 3/30
1L, 2H, r=0.9	213 ± 174	-0.998 ± 0.001	-0.847 ± 0.319	8.647 ± 1.388	8.462	3967 ± 3247 3/30
2L, 1H, r=0.5	253 ± 840	-0.974 ± 0.023	0.197 ± 0.255	1.043 ± 0.390	0.963	303 ± 107 30/30
2L, 1H, r=0.7	157 ± 256	-0.994 ± 0.005	-0.491 ± 0.604	2.647 ± 1.461	2.466	3800 ± 4033 8/30
2L, 1H, r=0.9	147 ± 86	-0.994 ± 0.006	-0.456 ± 0.684	30.083 ± 124.929	7.900	4025 ± 3863 8/30
2L, 2H, r=0.5	100	-0.978 ± 0.022	0.122 ± 0.128	0.987 ± 0.202	0.930	250 ± 51 30/30
2L, 2H, r=0.7	1257 ± 2305	-0.991 ± 0.023	-0.498 ± 0.587	2.568 ± 1.326	2.281	1631 ± 3452 13/30
2L, 2H, r=0.9	1020 ± 2786	-0.995 ± 0.005	-0.164 ± 0.745	6.988 ± 4.410	8.773	5107 ± 5355 15/30

ones. Because end-of-training gradients of a generalized model are ill-determined, the last two columns read the logged trajectory at the last epoch before a generalizing seed's adversarial accuracy first exceeds 0.25 (undefined at $r{=}0.5$, where adversarial accuracy starts near 0.5; at 2L, 2H $r{=}0.7$ two seeds exceed 0.25 at the first log and are omitted). Before escape, generalizing seeds are as anti-aligned as trapped ones, and in two-layer models at $r{=}0.9$ their ratio sits at or above the $9\times$ force-balance value (the three one-layer escapers at 1L, 2H are below it, at $6.8\times$); the time-matched comparison below shows that this does not distinguish them from trapped seeds at the same epoch.

Table 18: Gradient statistics by outcome, MESP at weight decay 0.1, for every configuration containing both outcomes. "Final" columns are at the end of training: ratio is the median anti-shortcut/shortcut-consistent gradient norm ratio, $\cos\theta$ the mean cosine similarity, given as generalizing / non-generalizing. "Pre-escape" columns are medians over generalizing seeds at the last logged epoch before adversarial accuracy first exceeds 0.25. All are computed on the held-out splits.

Config	r	Gen./non	Final ratio	Final $\cos\theta$	Pre-escape ratio	Pre-escape $\cos\theta$
1L, 1H	0.5	10/20	0.94 / 0.86	-0.14 / -0.62	—	—
1L, 1H	0.7	2/28	3.56 / 2.29	-0.59 / -0.98	2.26	-0.97
1L, 2H	0.5	3/27	1.12 / 0.98	$+0.64$ / -0.50	—	—
1L, 2H	0.7	3/27	1.96 / 2.32	$+0.47$ / -0.96	2.27	-0.99
1L, 2H	0.9	3/27	6.12 / 8.70	$+0.03$ / -0.95	6.81	-0.96
2L, 1H	0.5	3/27	1.38 / 0.95	$+0.56$ / $+0.16$	—	—
2L, 1H	0.7	8/22	2.37 / 2.47	$+0.32$ / -0.79	2.60	-0.95
2L, 1H	0.9	7/23	2.85 / 8.38	$+0.63$ / -0.79	9.55	-0.88
2L, 2H	0.7	13/17	2.19 / 2.42	-0.09 / -0.81	2.47	-0.95
2L, 2H	0.9	16/14	2.70 / 9.24	$+0.39$ / -0.80	10.26	-0.88

Time-matched comparison. Comparing a seed's pre-escape ratio with where trapped seeds *end* mixes epochs. Table 19 instead compares each late escaper at MESP $r{=}0.9$ (escape epoch at or after 1,900, so the epoch before escape is at or after 1,800) with the trapped seeds of the same cell at the same logged epoch, on the held-out splits and on the training subsets, computed from the per-100-epoch checkpoints. Over epochs 1,900–10,000, trapped 2L, 2H seeds sit at a per-epoch median of 6.9–$8.9\times$ on the held-out splits. On the training subsets they stay near $9\times$ throughout (median $9.5\times$ at epoch 100 and $8.6\times$ at 15,000), even though they have memorized the minority; their training-subset cosine rises from -1.00 to -0.22 over the same period, so they are not at the anti-aligned stationary point either.

The epoch just before escape often lies inside the transition: over epochs 1,900–10,000 trapped two-layer seeds sit at a median adversarial accuracy of 0.08 (0.06–0.10 between the 10th and 90th percentiles of seeds), while 10 of the 12 late escapers are already at 0.12–0.25 at the epoch before escape. We therefore also compare at fixed leads of 500 and 1,000 epochs before escape. The rise is gradual enough that even these leads are partly inside it: 8 of the 9 late 2L, 2H escapers at 500 epochs, and 7 of 9 at 1,000, are above 0.096, the 90th percentile of the trapped seeds. Slowly rising adversarial accuracy is thus the earliest sign of escape among the gradient and accuracy statistics; the layer-placement marker of Appendix G appears earlier. On the training subsets, which is what the force-balance argument concerns, no comparison reaches significance: 6 of 9 at the epoch before escape ($p{=}0.51$), 7 of 9 at 500 epochs ($p{=}0.18$) and 4 of 9 at 1,000 epochs, where the escapers' median ($7.2\times$) is below the trapped seeds' ($9.1\times$). The only nominally significant result, 8 of 9 on the held-out splits at the epoch before escape, is on the quantity a rising held-out anti-shortcut loss can inflate by itself.

The uncancelled component. At the shortcut posterior the two subsets' contributions cancel along $\bar{g}_{\text{SC}}$, so the part of the training gradient that can move the model is the component of $(1-r)\bar{g}_{\text{anti}}$ orthogonal to $\bar{g}_{\text{SC}}$ (Section 6). Table 19 reports its norm and the norm of the full training gradient at the same leads. Neither is larger in escapers: at 2L, 2H the full gradient is 1.26 against 1.98 at 1,000 epochs (3 of 9 above) and 1.10 against 1.98 at 500 (4 of 9), and the orthogonal component 0.75 against 1.14 (2 of 9) and 1.08 against 1.51 (5 of 9). We also measured whether the negative training gradient points where the weights actually go, as the cosine with the displacement from the current checkpoint to the generalization-epoch checkpoint (escapers) or to the checkpoint 1,000 epochs later (trapped seeds). It does not, in either group: every such cosine lies within ± 0.16, the group medians

within ± 0.007. Under AdamW the raw gradient is a poor guide to the update, which is consistent with this.

Table 19: Late escapers (escape epoch $\geq$ 1,900) against trapped seeds at the same epoch, MESP r=0.9, weight decay 0.1. The escape epoch is the first logged epoch with adversarial accuracy above 0.25; "before escape" is the logged epoch just before it, and the fixed leads are 500 and 1,000 epochs before it. "Escaper" is the median over late escapers; "trapped" the median over those epochs of the trapped seeds' median at the same epoch; "above" counts escapers above the trapped median, with a two-sided sign-test p. Ratio is $\|\bar{g}_{\text{anti}}\|/\|\bar{g}_{\text{SC}}\|$; "full" is $\|r\bar{g}_{\text{SC}} + (1-r)\bar{g}_{\text{anti}}\|$ and "orth." the norm of the component of $(1-r)\bar{g}_{\text{anti}}$ orthogonal to $\bar{g}_{\text{SC}}$. Regenerated by `python scripts/timing_analyses.py time-matched` (held-out) and `python -m analysis.trajectory_audit train-grad` and `lead-grad` (training).

Config	Statistic, gradients on	Epoch	Escaper	Trapped	Above	p
2L, 2H	ratio, held-out	before escape	12.2×	8.5×	8/9	0.04
2L, 2H	ratio, training	before escape	13.7×	9.7×	6/9	0.51
2L, 2H	ratio, training	−500	12.1×	8.6×	7/9	0.18
2L, 2H	ratio, training	−1,000	7.2×	9.1×	4/9	1.00
2L, 2H	full, training	−500 / −1,000	1.10 / 1.26	1.98 / 1.98	4/9, 3/9	1.00, 0.51
2L, 2H	orth., training	−500 / −1,000	1.08 / 0.75	1.51 / 1.14	5/9, 2/9	1.00, 0.18
2L, 1H	ratio, held-out	before escape	12.8×	8.3×	3/3	0.25
2L, 1H	ratio, training	before escape	12.5×	9.2×	3/3	0.25
2L, 1H	ratio, training	−500 / −1,000	4.9× / 12.2×	8.7× / 8.4×	1/3, 3/3	1.00, 0.25
2L, 1H	full, training	−500 / −1,000	2.74 / 4.57	3.20 / 3.72	1/3, 2/3	1.00, 1.00

G ATTENTION TO THE MAX-VALUED TOKEN

Section 5.3 reports that attention is withdrawn from the maximum-valued token in generalizing models, after generalization. This appendix gives the values behind that statement: at the end of training (Tables 20–22), and at and before the generalization epoch (Tables 23 and 24). Only the latter bear on how generalization happens; end-of-training values describe where a run ended. $p_{\max}$ is defined in Appendix H.

Table 20 is computed from per-epoch attention logs on one probe sequence and covers $r \in \{0.5, 0.7\}$; Table 21 measures all ratios split-wide.

Table 20: Max-token attention $p_{\max}$ at the final checkpoint (wd = 0.1). The first two columns average all 30 seeds on each split; the last two split the shortcut-consistent column by outcome, with seed counts in parentheses. Two patterns stand out. One-layer models at $r{=}0.7$ end with most attention on the max-valued token (0.886 at 1H). Wherever a configuration contains both outcomes and a shortcut circuit has formed ($r{=}0.7$), generalizing seeds end far below non-generalizing ones; this is after generalization and is not a predictor of it (Table 23). At $r{=}0.5$, where the QK fingerprint shows no shortcut circuit forms (Table 2), both groups sit low and the separation vanishes.

		All seeds		Shortcut-consistent split, by outcome	
Config	r	SC split	Adv split	Generalizing	Non-generalizing
1L, 1H	0.5	0.114 ± 0.109	0.180 ± 0.134	0.199 ± 0.132 (10)	0.072 ± 0.066 (20)
1L, 1H	0.7	0.886 ± 0.161	0.951 ± 0.177	0.308 ± 0.034 (2)	0.927 ± 0.035 (28)
1L, 2H	0.5	0.102 ± 0.160	0.195 ± 0.234	0.140 ± 0.038 (3)	0.098 ± 0.168 (27)
1L, 2H	0.7	0.624 ± 0.245	0.678 ± 0.281	0.147 ± 0.028 (3)	0.677 ± 0.195 (27)
2L, 1H	0.5	0.113 ± 0.148	0.120 ± 0.141	0.108 ± 0.035 (3)	0.113 ± 0.156 (27)
2L, 1H	0.7	0.485 ± 0.270	0.542 ± 0.279	0.178 ± 0.132 (8)	0.597 ± 0.215 (22)
2L, 2H	0.5	0.078 ± 0.103	0.114 ± 0.114	— (0)	0.078 ± 0.103 (30)
2L, 2H	0.7	0.384 ± 0.255	0.387 ± 0.255	0.141 ± 0.109 (13)	0.570 ± 0.155 (17)

This measurement is an association, not an intervention, and it is taken at the final checkpoint, so it describes where training ended rather than the state at the generalization epoch.

Split-wide measurement. Table 20 is computed on one fixed probe sequence per split (Appendix H), which leaves open whether it reflects circuit behavior or an idiosyncrasy of that sequence. We therefore recomputed $p_{\max}$ as an expectation over *all* 2,048 examples of each split, using the stored checkpoints, and report it in Table 21. The two measurements agree where a shortcut circuit exists ($r{=}0.7$, within 0.07 in every cell) but not at $r{=}0.5$, where they differ by up to $2\times$ (0.114 against 0.206 at 1L, 1H; 0.078 against 0.195 at 2L, 2H): with unconcentrated attention, one probe sequence is a noisy estimate. Neither measurement renormalizes over the integer positions; both report the raw weight in the seven-way softmax, for which uniform attention gives $1/7 \approx 0.143$. Every conclusion drawn from the single probe still holds on the split-wide values, which we treat as primary.

We note three things. First, the *shortcut trap* is clear: one-layer models at $r{=}0.9$ put 0.943 of the final position's attention on the max-valued token across the whole split, about 6.6 times the uniform level of $1/7$, with a seed-to-seed standard deviation of 0.013. Second, at the stored checkpoint the *outcome separation* is large at every ratio where a shortcut circuit forms: at $r{=}0.9$, generalizing versus non-generalizing seeds sit at 0.155 versus 0.594 (2L, 1H) and 0.196 versus 0.542 (2L, 2H). This separation arises after generalization; at the generalization epoch the two groups are indistinguishable (Table 23). Third, there is little separation at $r{=}0.5$: every group lies between 0.125 and 0.215, and the two outcomes differ by at most 0.07 within a cell, so no max-selecting circuit forms there. The val and adv columns agree to within 0.004 everywhere, which is what one expects if $p_{\max}$ is a property of the circuit rather than of the input distribution.

Per-head attention. Appendix H averages $p_{\max}$ over all heads and layers, which can hide head specialization: averaging a head pinned to the max token with one that has left it gives a value that describes neither head. Table 22 disaggregates, giving the largest and smallest per-head value in each model and their spread.

Table 21: Max-token attention as a split-wide expectation over all 2,048 examples per split (wd = 0.1, 30 seeds per cell), to be read against the single-probe values in Table 20. The stored checkpoint is the best-adversarial-accuracy checkpoint, with ties at 100% broken by the lowest loss. For generalizing seeds it is therefore *not* the state at generalization: at r=0.9 it lies a median of 6,600 epochs later (Table 23). Values are raw weights in the seven-way softmax; uniform attention gives $1/7 \approx 0.143$.

		All seeds		SC split, by outcome	
Config	r	SC split	Adv split	Generalizing	Non-generalizing
1L, 1H	0.5	0.206 ± 0.052	0.203	0.187 (10)	0.215 (20)
1L, 1H	0.7	0.831 ± 0.137	0.825	0.330 (2)	0.867 (28)
1L, 1H	0.9	$\mathbf{0.943 \pm 0.013}$	0.939	— (0)	0.943 (30)
1L, 2H	0.5	0.198 ± 0.036	0.195	0.178 (3)	0.200 (27)
1L, 2H	0.7	0.636 ± 0.187	0.631	0.178 (3)	0.687 (27)
1L, 2H	0.9	0.707 ± 0.164	0.707	0.319 (3)	0.750 (27)
2L, 1H	0.5	0.187 ± 0.031	0.186	0.125 (3)	0.194 (27)
2L, 1H	0.7	0.501 ± 0.224	0.498	0.171 (8)	0.621 (22)
2L, 1H	0.9	0.492 ± 0.213	0.491	**0.155** (7)	**0.594** (23)
2L, 2H	0.5	0.195 ± 0.019	0.192	— (0)	0.195 (30)
2L, 2H	0.7	0.362 ± 0.193	0.359	0.157 (13)	0.519 (17)
2L, 2H	0.9	0.357 ± 0.207	0.358	**0.196** (16)	**0.542** (14)

Table 22: Max-token attention disaggregated by head, split-wide on the shortcut-consistent split at the stored checkpoint (wd = 0.1). "Max head" and "Min head" are the largest and smallest p_{max} over the model's $n_{\text{layers}} \times n_{\text{heads}}$ heads, averaged across seeds; "Spread" is their difference. The 1L, 1H rows have a single head, so their spread is exactly 0, as expected. Elsewhere the spread separates outcomes as sharply as the mean does, and in the opposite direction from what head-averaging suggests.

		Non-generalizing		Generalizing	
Config	r	Max head / Min head	Spread	Max head / Min head	Spread
1L, 1H	0.5	0.215 / 0.215	0.000	0.187 / 0.187	0.000
1L, 1H	0.7	0.867 / 0.867	0.000	0.330 / 0.330	0.000
1L, 1H	0.9	0.943 / 0.943	0.000	—	—
1L, 2H	0.5	0.230 / 0.169	0.061	0.186 / 0.170	0.016
1L, 2H	0.7	0.901 / 0.473	0.427	0.201 / 0.156	0.044
1L, 2H	0.9	0.946 / 0.555	0.391	0.470 / 0.168	0.302
2L, 1H	0.5	0.217 / 0.171	0.046	0.148 / 0.102	0.046
2L, 1H	0.7	0.846 / 0.395	0.451	0.190 / 0.152	0.038
2L, 1H	0.9	0.818 / 0.371	0.447	0.161 / 0.149	**0.012**
2L, 2H	0.5	0.255 / 0.137	0.118	—	—
2L, 2H	0.7	0.903 / 0.223	**0.680**	0.210 / 0.104	0.105
2L, 2H	0.9	0.875 / 0.292	**0.583**	0.280 / 0.128	0.152

Three things follow. First, the division of labor is real and large: in non-generalizing 2L, 2H models at $r{=}0.7$ one head holds 0.903 of the final position's attention on the max-valued token while another holds 0.223, a spread of 0.680 that the all-head mean of 0.519 represents badly. Second, at the stored checkpoint, which for generalizing seeds lies well after generalization, it is not just the average that falls; *no* head stays strongly on the max token: in the two-layer cells the maximum over heads never exceeds 0.280 and spreads run 0.012–0.152, against 0.447–0.680 for their non-generalizing counterparts. The exception is the one-layer, two-head cell at $r{=}0.9$, where the three generalizing seeds still reach 0.470 on their busiest head. This describes where generalizing models end up, not the state in which they generalize (Table 23). Third, at the end of training it is the failing seeds that keep one head pinned while the others move, consistent with the absence of any head-count acceleration in first-robust-head formation (Section 5.2).

At the generalization epoch. For MESP at $r{=}0.9$ a checkpoint is kept every 100 epochs, so for that ratio we measure split-wide $p_{\max}$ at the generalization epoch itself (Table 23). At generalization, the busiest head still places 0.80 (2H) and 0.88 (1H) of the final position's attention on the max token, at least 0.5 in 21 of 23 seeds, which is as high as in non-generalizing seeds at the end of training. It falls to near uniform only later. On MESP, as on FESP, attention leaves the shortcut token after the rule has formed, not before. Attention at the generalization epoch therefore does not separate the outcomes, and the stored-checkpoint separation of Tables 21 and 22 arises afterwards.

Table 23: Split-wide max-token attention on the shortcut-consistent split, MESP $r{=}0.9$, weight decay 0.1, from the per-100-epoch checkpoints; busiest head / mean over heads, each averaged (mean) over seeds. "At gen." is the last checkpoint at or before the first 100% adversarial epoch. "Stored" is the best-adversarial checkpoint used in Tables 21 and 22; the last column gives its median distance from the generalization epoch. Regenerated by `python -m analysis.trajectory_audit pmax`.

Config	Outcome (n)	At gen.	Stored	Epoch 15,000	Stored − gen. epoch
2L, 1H	generalizing (7)	0.876 / 0.583	0.161 / 0.155	0.156 / 0.152	6,600
2L, 1H	non-gen. (23)	—	0.818 / 0.594	0.810 / 0.586	—
2L, 2H	generalizing (16)	0.804 / 0.477	0.280 / 0.196	0.257 / 0.184	6,600
2L, 2H	non-gen. (14)	—	0.875 / 0.542	0.870 / 0.533	—

This is still an association. Because the max element is a summand of the rule, a head attending to it in a model that is already perfect on the adversarial split is part of a correct computation, not evidence of shortcut reliance. The later withdrawal is most simply read as weight decay removing a redundant head.

Which layer holds the max token. The busiest head says nothing about the others, so Table 24 tracks max-token attention per layer (mean over a layer's heads) from the first logged epoch. In 2L, 1H the busiest head of a generalizing model is the layer-1 head, which stays on the max token through generalization (median 0.95 at the generalization epoch; the mean, 0.88, matches the busiest-head mean in Table 23, which is at the same checkpoint), while the layer-0 head, the non-pinned one, has moved off it (0.77 at epoch 100, 0.27 at generalization). The split is visible long before generalization and separates outcomes. Already at epoch 100, generalizing seeds put more max-token attention in layer 1 than in layer 0, and non-generalizing seeds the reverse: the difference $p^{(1)}_{\max} - p^{(0)}_{\max}$ has median $+0.21$ against -0.27 at 2L, 1H (Mann–Whitney AUC 0.80, $p{=}0.019$) and $+0.19$ against -0.26 at 2L, 2H (AUC 0.82, $p{=}0.003$). At 2L, 2H, 9 of the 10 seeds whose layer-1 lead exceeds 0.1 at epoch 100 generalize, against 7 of 20 otherwise. Among seeds that have not yet generalized, the separation persists at epoch 1,000 (AUC 0.77 and 0.90), and in the time-matched comparison of late 2L, 2H escapers all 9 sit below the trapped seeds on layer-0 attention 1,000 epochs before escape (0.53 against 0.62). It is the one measurement in the paper that separates outcomes well before generalization, and we treat it as exploratory. It is one of 14 pre-generalization outcome comparisons we report (gradient ratio on held-out and training subsets, full gradient, orthogonal component, cosine, displacement alignment, two probes, busiest-head and mean attention, QK, OV, the circuit criterion, and per-layer attention), all on MESP at $r{=}0.9$. Under a Bonferroni correction over these 14 ($\alpha{=}0.0036$), the 2L, 1H result ($p{=}0.019$) does not survive, and the 2L, 2H result ($p{=}0.003$) only just does.

The transplant of Appendix K allows a check on a new outcome for some of the same seeds. It started from the epoch-500 checkpoints of ten of these 2L, 2H seeds and trained them at a different ratio. At epoch 500 the layer-1 lead is higher in the six seeds that generalize after the transplant than in the four that do not (median $+0.16$ against -0.21, AUC 0.79, p=0.17). The marker values are in-sample, and the new outcome is tied to the old one, since four of the six transplant generalizers also generalized at r=0.9. Agreement is therefore expected even if the marker carries no information beyond whether a seed was going to generalize anyway, and ten seeds cannot settle the question. The interpretation also needs care. After layer 0, residual streams are mixed by bidirectional attention, so a layer-1 head attending to the max-token position reads that position's layer-0 output, which may already aggregate the sequence. Attention weights alone therefore do not establish that the shortcut is routed through the last layer, for the same reason that attention to a summand is not evidence of shortcut reliance (Section 5.3).

Table 24: Max-token attention by layer, MESP r=0.9, weight decay 0.1, split-wide on the shortcut-consistent split from the per-100-epoch checkpoints; medians over seeds of the layer mean. Seeds that have already generalized are excluded at epochs 100 and 1,000 (seed counts in parentheses, generalizing / non-generalizing). AUC is the Mann–Whitney probability that a generalizing seed has the larger $p^{(1)}_{\max} - p^{(0)}_{\max}$, with two-sided p. Regenerated by `python -m analysis.trajectory_audit within-run`.

Config	Epoch	Layer 0, gen. / non	Layer 1, gen. / non	AUC	p
2L, 1H	100 (7/23)	0.77 / 0.86	0.99 / 0.68	0.80	0.019
2L, 1H	1,000 (6/23)	0.55 / 0.79	0.98 / 0.58	0.77	0.048
2L, 1H	at gen. / 15,000	0.27 / 0.60	0.95 / 0.50	—	—
2L, 2H	100 (16/14)	0.63 / 0.77	0.71 / 0.47	0.82	0.003
2L, 2H	1,000 (9/14)	0.53 / 0.69	0.80 / 0.45	0.90	0.002
2L, 2H	at gen. / 15,000	0.41 / 0.60	0.58 / 0.41	—	—

Memorization before generalization, within a run. At r=0.9 training accuracy is uninformative, since the shortcut alone gives 0.9, so we measure accuracy on the anti-shortcut training examples directly from the checkpoints. The 11 seeds that escape by epoch 1,100 fit the minority and generalize together. The 12 late escapers (escape at or after epoch 1,900) memorize first: every one fits at least half its anti-shortcut training examples before escaping (median epoch 1,350, a median of 6,450 epochs before escape), 9 of the 12 fit at least 90% of them before escaping, and at the epoch before escape they fit 0.75–1.00 of them while held-out adversarial accuracy is at most 0.25. Trapped two-layer seeds memorize on a similar schedule (median $5{,}950$ epochs to 90% of the minority at 2H). Within a run, then, memorizing the minority does not prevent later generalization: late escapers generalize from a memorized state.

H Analysis Methodology

Circuit identification: scores. All four scores are defined on the *logit margin* rather than on accuracy, which makes them continuous and unbounded above. For an example with true label y, let

$$\Delta(x) \;=\; z_y(x) \;-\; \max_{y' \neq y} z_{y'}(x) \tag{1}$$

be the margin between the correct logit and the strongest competitor. The **ablation scores** are the mean drop in this margin when component c's output is zeroed, taken separately on the two splits:

$$s_{\text{val}}(c) = \mathbb{E}_{x \sim \text{val}}[\Delta(x) - \Delta_{c=0}(x)]\,, \qquad s_{\text{adv}}(c) = \mathbb{E}_{x \sim \text{adv}}[\Delta(x) - \Delta_{c=0}(x)]\,, \tag{2}$$

so positive means the component helps on that split. The **cross-split patching scores** are measured on matched pairs. A pair $(x^{\text{sc}}, x^{\text{anti}})$ is drawn so that the two examples share the *same shortcut feature* — identical `(max_val, max_pos)` for the magnitude-shortcut tasks, identical `first_num` for the positional ones — while carrying opposite true labels, so that everything differing between them lies in the non-shortcut tokens. We take up to 512 such pairs. Writing Δ^{anti} for the margin on the anti-shortcut member and $\Delta^{\text{anti}}_{c \leftarrow \text{sc}}$ for the same margin after component c's activation is replaced by its value on the shortcut-consistent member,

$$g_{\text{fwd}}(c) = \mathbb{E}_{\text{pairs}}\left[\Delta^{\text{anti}}_{c \leftarrow \text{sc}} - \Delta^{\text{anti}}\right], \tag{3}$$

and g_{rev} is the same quantity with the roles of the two members exchanged. Both are raw restoration scores, not normalized recoveries, so their scale depends on the logit scale of the checkpoint.

Circuit identification: labels. A head carries the `shared` label at checkpoint t when all four scores are positive: $s_{\text{val}} > 0, s_{\text{adv}} > 0$ (it contributes to both splits) and $g_{\text{fwd}} > 0, g_{\text{rev}} > 0$ (patching transfers in both directions). Split selectivity, $\sigma = |s_{\text{val}} - s_{\text{adv}}|/(|s_{\text{val}}| + |s_{\text{adv}}| + \epsilon)$ with $\epsilon = 10^{-8}$ so that a head with no effect on either split gives $\sigma = 0$ rather than $0/0$, is what separates the two *shortcut* labels from each other: when $\sigma \geq 0.2$ the sign pattern $s_{\text{val}} > 0 > s_{\text{adv}}$ gives `shortcut_val` and the reverse gives `shortcut_adv`, each taking priority over `shared`; when $\sigma < 0.2$ the head is either `shared` or `uninvolved`. Selectivity therefore never excludes `shared`, but the shortcut labels pre-empt it.

Two metrics are built on this label and they are not interchangeable. The *epoch of first robust head* is the earliest checkpoint at which any head across all layers is labeled `shared` *and* has mean patching score $\bar{g} = (g_{\text{fwd}} + g_{\text{rev}})/2 > 0.5$, the extra threshold keeping weakly transferring heads from triggering the event. The *robust circuit fraction at generalization* is the share of all heads labeled `shared` — without the $\bar{g}$ threshold — at the last checkpoint at or before the generalization epoch. The *final robust circuit fraction* is the same share at the last training checkpoint, and is lower in runs that regress after generalizing.

Circuit identification: calibration and where it fails. The criterion is a sign test on four scores, so near chance performance it can be satisfied by small fluctuations of either sign. Table 25 calibrates it directly: for each configuration we compare the peak adversarial accuracy of the seeds in which the criterion fires against the seeds in which it does not, and report the share of firing seeds that ever reach 100%.

Table 25: Calibration of the first-robust-head criterion, MESP at weight decay 0.1, 30 seeds per cell. "Fires" counts seeds where the criterion is met at some checkpoint; "Gated" counts them under the accuracy gate described above (margin 0.05 over chance), whose count matches the generalization rate of Table 1 in eleven of the twelve cells. The next two columns give mean *peak* anti-shortcut accuracy among firing and non-firing seeds; a genuine detector should separate them. The last column is the share of firing seeds that ever reach 100%. The criterion separates cleanly at $r \geq 0.7$ and in one-layer models at r=0.5, and fails completely in exactly one cell—2L, 2H at r=0.5, where it fires in 23/30 seeds with *no* separation (0.529 against 0.523, both at chance) and never predicts generalization.

Config	r	Fires	Gated	Peak adv. (fires)	Peak adv. (no fire)	Share reach 100%
1L, 1H	0.5	10/30	10/30	1.000	0.541	1.00
1L, 1H	0.7	2/30	2/30	1.000	0.038	1.00
1L, 1H	0.9	0/30	0/30	—	0.003	—
1L, 2H	0.5	4/30	3/30	0.885	0.537	0.75
1L, 2H	0.7	3/30	3/30	1.000	0.152	1.00
1L, 2H	0.9	3/30	3/30	1.000	0.029	1.00
2L, 1H	0.5	9/30	3/30	0.693	0.529	0.33
2L, 1H	0.7	8/30	8/30	1.000	0.308	1.00
2L, 1H	0.9	10/30	8/30	0.827	0.120	0.70
2L, 2H	0.5	**23/30**	**0/30**	**0.529**	**0.523**	**0.00**
2L, 2H	0.7	14/30	13/30	0.957	0.352	0.93
2L, 2H	0.9	20/30	16/30	0.825	0.128	0.80

The 2L, 2H r=0.5 row, set in bold, shows the failure one would expect from how the criterion is built: 23 of 30 seeds at 2L, 2H, r=0.5 satisfy the criterion in a configuration that generalizes in 0% of seeds, whose OV parity Spearman sits at 0.284 (Table 2), and whose firing and non-firing seeds are indistinguishable in peak adversarial accuracy. Tightening the patching thresholds does not rescue it—requiring $g_{\text{fwd}} > 0.5$ and $g_{\text{rev}} > 0.5$ individually, rather than their mean, only reduces 23 to 20—because the problem is not threshold height but the absence of any underlying circuit to detect. We therefore read the r=0.5 column of Table 16 as uninformative rather than as evidence of early circuit formation.

Gated criterion. The repair is to require that the model be above chance on the adversarial split at the checkpoint where the event fires. A bare "above chance" test is not enough—chance-level accuracy fluctuates over 0.5, and it only takes the 2L, 2H $r{=}0.5$ cell from 23/30 to 21/30—but adding a margin works, and the result is the same for every margin from 0.05 to 0.30. At a margin of 0.05 the gated event fires in 0/30 seeds at 2L, 2H $r{=}0.5$, and its count matches the generalization count exactly in eleven of the twelve configurations, the one exception being 2L, 1H at $r{=}0.9$, off by one (8 gated against 7 generalizing). At 2L, 2H $r{=}0.9$ the gate removes four of the twenty raw firings and leaves exactly the 16 generalizing seeds. Table 25 reports the gated counts alongside the raw ones.

The gate improves specificity at a cost, so we report both: by construction the gated event cannot fire before adversarial accuracy has already moved, so it cannot serve as evidence that circuit formation *precedes* the behavioral change. The raw criterion can fire early, which is what makes it interesting and also what makes it unreliable at chance. On head count in Section 5.2 the two agree: 1H is slightly earlier under both (4420 against 4785 raw, 5212 against 5300 gated), and neither difference is meaningful given the spread.

The same calibration at weight decay 0.4. Section 5 scopes the mechanistic analysis to weight decay 0.1, but the regression story of Appendix A lives at 0.4, so we ran the identical calibration there (Table 26). The criterion is *better* behaved in that regime, which confirms that the failure above is specific to that cell. The pathological cell is not pathological at 0.4: 2L, 2H at $r{=}0.5$ fires in 3/30 seeds rather than 23/30, and its firing and non-firing seeds do separate (0.684 against 0.532). Across all twelve cells the gated count matches the generalization count exactly in nine, the three exceptions each being off by one: 1L, 1H $r{=}0.5$, 2L, 1H $r{=}0.9$ and 1L, 2H $r{=}0.9$. The first is a single seed that peaked at $2{,}047$ of $2{,}048$ adversarial examples, which is why its row shows a peak of 1.000 at three decimals but a 0.00 share reaching 100%. The conclusion we draw in Section 5.2 is therefore that the raw criterion degrades specifically where no circuit exists to detect, not that it is unreliable in general—at weight decay 0.4 the shortcut circuit is present in more cells and the criterion tracks it.

Table 26: Calibration of the first-robust-head criterion at weight decay 0.4, MESP, 30 seeds per cell; columns as in Table 25, with the generalization count added for comparison against the gated count. Recomputed with `python scripts/paper_tables.py calibration -task max_parity_sum_parity -weight-decay 0.4`.

Config	r	Gen.	Fires	Gated	Peak adv. (fires)	Peak adv. (no fire)	Share reach 100%
1L, 1H	0.5	0/30	1/30	1/30	1.000	0.546	0.00
1L, 1H	0.7	0/30	0/30	0/30	—	0.006	—
1L, 1H	0.9	0/30	0/30	0/30	—	0.002	—
1L, 2H	0.5	4/30	4/30	4/30	1.000	0.538	1.00
1L, 2H	0.7	2/30	2/30	2/30	1.000	0.050	1.00
1L, 2H	0.9	3/30	4/30	4/30	1.000	0.004	0.75
2L, 1H	0.5	2/30	2/30	2/30	1.000	0.540	1.00
2L, 1H	0.7	7/30	7/30	7/30	1.000	0.209	1.00
2L, 1H	0.9	11/30	12/30	12/30	1.000	0.065	0.92
2L, 2H	0.5	1/30	**3/30**	1/30	**0.684**	**0.532**	**0.33**
2L, 2H	0.7	17/30	17/30	17/30	1.000	0.286	1.00
2L, 2H	0.9	12/30	14/30	12/30	0.880	0.103	0.86

Max-valued-token attention. At each logged checkpoint we record the attention the final position places on every input position, for one fixed probe sequence per split, and average over all heads and layers. The *max-token attention* $p_{\max}$ is the weight on the position holding the largest integer in that sequence, read directly from the seven-way softmax without renormalizing over the integer positions, so uniform attention gives $1/7$. Integer positions are identified by token value; the `START` and `END` tokens carry the two values above the integer range and are excluded, since `START` would otherwise be the largest value in every sequence. Because a single probe sequence is used, $p_{\max}$ reports what one representative input elicits rather than a dataset average; its seed-to-seed standard deviation is nonetheless small in the trapped configurations (0.013 for 1L, 1H at $r{=}0.9$ on the split-wide measurement of Table 21), and the probe is identical across all seeds and configurations, so the comparisons in Table 20 are like-for-like.

Gradient attribution. At each logged checkpoint, gradients are computed separately on the shortcut-consistent and adversarial splits using the same cross-entropy loss. Gradient cosine similarity is the cosine of the angle between the two flattened gradient vectors. The gradient ratio is the adversarial gradient norm divided by the shortcut-consistent gradient norm. The conflict-resolution epoch is the first epoch at which cosine similarity exceeds -0.1, i.e. the first epoch at which the two gradients are no longer strongly opposed.

OV parity ceiling. The OV parity statistic is Spearman ρ between a head's per-token-value OV logit differences (20 values, untied in general) and the parity of those token values (a binary vector with 10 ties in each group). A binary variable with a $10/10$ tie structure cannot attain $\rho = 1$ against an untied one. Ranking the parity vector gives 5.5 for the ten even values and 15.5 for the ten odd ones, so with x perfectly sorted by parity, $\sum(x_i - \bar{x})(y_i - \bar{y}) = 500$, $\sum(x_i - \bar{x})^2 = n(n^2-1)/12 = 665$ and $\sum(y_i - \bar{y})^2 = 500$, giving

$$\rho_{\max} = \frac{500}{\sqrt{665 \times 500}} = 0.8671. \tag{4}$$

Values near 0.86 in Tables 2 and 6 are therefore at the ceiling. The bound assumes the margins $d(t)$ are untied; when a head's margins themselves collapse onto few values, $|\rho|$ can slightly exceed it, which is why one FESP entry reads 0.873.

QK/OV by seed outcome. Table 2 reports each row under one convention, but two of its rows contain both generalizing and non-generalizing seeds, so we give the split here. At 1L, 1H r=0.5 the 10 generalizing seeds reach at-generalization OV parity -0.062 ± 0.568 against a final-checkpoint -0.023 ± 0.136 for the other 20. The signed values are misleading: a head can map odd tokens to either logit direction, and the signed statistic averages the two. In absolute value the generalizing seeds reach $|\rho| = 0.468 \pm 0.289$ against 0.115 ± 0.072 for the rest, so parity structure does form in the seeds that solve the task. In the r=0.9 rows of Table 2 signed and absolute values coincide. The informative split is 2L, 2H at r=0.9, the one cell where both outcomes occur *and* a shortcut circuit forms. There the 16 generalizing seeds reach OV parity 0.855 ± 0.035 at generalization while the 14 failures sit at 0.739 ± 0.076 at the final checkpoint, so parity structure develops in both groups. Their final QK ρ differs (0.953 ± 0.018 against 0.580 ± 0.212 for the generalizing seeds), but that difference arises after generalization, when the generalizing models withdraw attention from the max token (Section 5.3); neither statistic separates the outcomes before generalization.

QK/OV circuit analysis. For each attention head, the QK circuit is characterized by Spearman ρ between query-key dot products and the magnitude rank of the element at each position; a value near 1 indicates the head preferentially attends to higher-valued tokens, i.e. the magnitude-sorting shortcut circuit.

The OV statistic involves *no trained probe*; it is a purely weight-space quantity. For head h of layer ℓ we push each integer token's embedding through that head's value and output projections and then through the unembedding, giving one effective logit vector per token identity, with positional encodings excluded so that only token identity contributes:

$$z(t) \;=\; W_{\text{out}}\, W_O^{(h)}\, W_V^{(h)}\, E[t] \;\in\; \mathbb{R}^C, \qquad t \in \{0, \ldots, 19\}, \tag{5}$$

where $W_V^{(h)}$ and $W_O^{(h)}$ are the head's slices of the value and output matrices, $E[t]$ is the token embedding, and C is the number of classes. This yields one scalar per token: for the binary tasks the class margin $d(t) = z_1(t) - z_0(t)$, and for the ternary tasks the top-1 minus top-2 logit margin, since token values there are not class indices. *OV parity Spearman* is then ρ between the 20 values $d(t)$ and the 20 parities $t \bmod 2$ (or $t \bmod 3$). A high value means the head's output-value path maps token parity monotonically onto the class margin. Because the parity target is tied in two groups of ten, this statistic is capped at 0.867 as derived above. Three limits apply. It is signed, so we report $|\rho|$ alongside it wherever the sign varies across seeds. It measures only the head's *direct* path to the logits, bypassing later layers and MLPs, which for a first-layer head in a two-layer model is not the operative path. And for the ternary tasks it is not well defined: Spearman against $t \bmod 3$ treats residues as ordinal, the unsigned top-1 minus top-2 margin discards which class wins, and the 0.867 ceiling does not apply to the $7/7/6$ tie structure. We therefore report OV parity for the binary tasks only.

Ablation drops are the change in the logit margin of Equation 1 when the head's output is zeroed, not a change in accuracy.

I Dataset Construction Audit

Section 3.2 states the sampling procedure; this appendix checks what that procedure actually produced: whether skewing the spurious ratio also skews the label marginal, and whether the evaluation splits are disjoint from training.

Label marginals. Because we condition acceptance on shortcut status but not on the label, the marginal $P(y)$ is free to drift with r. Table 27 measures it on the generated training files. On the two *position*-shortcut tasks the marginal is essentially flat: FESP stays within 0.3 percentage points of 0.5 at every ratio, and FE-Mod3 within 0.9 points of $1/3$. On the two *magnitude*-shortcut tasks it drifts, most visibly at MESP $r{=}0.9$, where $P(y{=}1)$ reaches 0.550. The reason is structural rather than a sampler defect: the maximum of five draws from $[0, 20)$ is not parity-balanced, so conditioning on "max parity equals sum parity" tilts the sum parity in the same direction, and the tilt grows with the fraction of examples subject to that condition.

Table 27: Measured label marginals of the generated training splits. Binary tasks report $P(y{=}0)/P(y{=}1)$; ternary tasks report $P(y{=}0)/P(y{=}1)/P(y{=}2)$. Uniform is 0.500 and 0.333 respectively. The two position-shortcut tasks stay flat across the sweep; the two magnitude-shortcut tasks drift with r.

Task	Low r	Middle r	High r
Binary sweep, $r \in \{0.5, 0.7, 0.9\}$			
MESP (magnitude)	$.505/.495$	$.479/.521$	$.450/.550$
FESP (position)	$.502/.498$	$.497/.503$	$.498/.502$
Ternary sweep, $r \in \{0.15, 0.33, 0.50\}$			
FE-Mod3 (position)	$.325/.334/.342$	$.330/.336/.334$	$.342/.334/.325$
ME-Mod3 (magnitude)	$.340/.307/.353$	$.337/.332/.330$	$.335/.355/.311$

This drift is a possible confound, because a skewed $P(y)$ could in principle produce a gradient asymmetry by ordinary class imbalance rather than by the shortcut. Three observations bound it. First, the drift is absent exactly where the effect is *strongest*: FESP, whose marginal moves by at most 0.3 points, reaches a 77% peak against MESP's 53%, and FE-Mod3, which moves by at most 0.9 points, reaches 93%. A mechanism driven by class imbalance would have to explain why the two tasks with no class imbalance show the effect most strongly, and why the one task with the largest drift shows it least. Second, the ternary sweeps straddle the null ratio, and generalization is suppressed at $r{=}0.33$ in both directions even though the marginal there is the closest to uniform of the three—if class skew were the driver, the null ratio would be the *best* setting, not the worst. Third, the binary sweep runs $0.50 \to 0.90$ while $P(y{=}1)$ moves only $0.495 \to 0.550$; a 5-point class skew is a weak lever relative to the $0\% \to 53\%$ change in outcome.

Train/evaluation overlap. Rejection sampling excludes duplicates within a split and against every split already generated in the same process. Because the splits of a task were not always generated in one process, we measure overlap directly on the generated files. No split contains a duplicate. The val and adversarial splits are disjoint from each other and from training in every MESP, FESP and FE-Mod3 cell. ME-Mod3 has 6 and 3 adversarial sequences in training at $r{=}0.15$ and $r{=}0.33$ (0.3% and 0.1%). On n11, the $r{=}0.5$ training file is disjoint from both evaluation splits, but the $r{=}0.7$ and $r{=}0.9$ files contain all $2{,}048$ val sequences and 10 and 30 adversarial ones (0.5% and 1.5%). Since the generalization criterion requires perfection on all $2{,}048$ adversarial sequences, a few seen ones cannot produce a generalizing run; overlap can only raise adversarial accuracy, so it does not weaken the n11 one-layer trap at $r{=}0.9$; and the n11 null-ratio result is on the disjoint $r{=}0.5$ file. The n11 val accuracy at $r \geq 0.7$ is training accuracy, and we do not use it. The only other overlap is $8{,}195$ sequences shared between the MESP `train_r0.50` and `train_r0.70` files, which is train-to-train between independent runs and not leakage.

Shortcut-trapping on the training set. The claim of Section 4.2—that one-layer models are wrong even on anti-shortcut sequences they were trained on—is measurable directly on the training set. At $r{=}0.9$ exactly 90% of the training file is shortcut-consistent, so a model perfect on that portion and wrong on all $1{,}639$ anti-shortcut training examples scores exactly 0.900 overall. The measured final

training accuracies at weight decay 0.1 are 0.9012 (1L, 1H) and 0.9213 (1L, 2H), which bounds their accuracy on the anti-shortcut training examples at 1.2% and 21.3%. One-layer models therefore fail on anti-shortcut sequences with the labels in hand and gradients taken on them, which is a sharper statement of the trap than the pooled adversarial numbers convey, and it rules out memorization as *their* escape route.

Two-layer failures memorize. The same measurement, split by outcome, shows that it does not rule memorization out for two-layer models. Two-layer models at $r{=}0.9$ reach 0.9896 (1H) and 0.9985 (2H) training accuracy pooled over seeds, and generalizing seeds are at 1.000, so the pooled figure hides the failures. Taken alone, the 14 non-generalizing 2L, 2H seeds score 0.997 on the training set, which requires at least 0.969 on their anti-shortcut training examples (at least 0.905 in every seed), while scoring 0.097 on the held-out adversarial split and 0.910 on held-out shortcut-consistent examples. The 23 at 2L, 1H reach at least 0.864 on average. These models have fit the minority by memorizing it. Table 29 shows this is general. At the null ratio, failing two-layer models fit 0.95–0.996 of the entire 16,384-example training set on all four tasks, with held-out accuracy at chance; only the ternary tasks reach 0.99. Failing one-layer models fit 0.56–0.68, which is partial memorization but substantially less. The binary-task models have $67{,}072$ (two-layer) and $34{,}304$ (one-layer) trainable parameters, and either count is ample to store $16{,}384$ binary labels, so parameter count does not explain the difference. We have not shown that depth as such does. A more specific candidate is the one-layer readout path, which sees the `END` embedding plus a single attention-weighted average of value vectors before one MLP; a one-layer model with a wider MLP or a much larger d_{model} would separate the two (Appendix L). There are therefore two failure modes: one-layer models are shortcut-trapped, and two-layer models memorize.

When memorization happens, and at weight decay 0.4. Table 28 gives, per two-layer configuration, the epoch at which failing null-ratio models first reach 0.9 training accuracy, and the generalization epoch (first logged epoch at 100% adversarial accuracy) of the same configuration at each informative ratio. The 2H configurations memorize at the null quickly (median 450–900 epochs on every task) and the 1H configurations more slowly ($1{,}450$–$9{,}300$). Generalization at informative ratios spans the same range and is often bimodal: FE-Mod3 2L, 2H at $r{=}0.15$ has a median of 400, with 16 of 28 seeds by epoch 600 and the rest between 3,900 and 14,600. Across conditions there is thus no clear ordering in which one route closes the other; within a run, late escapers at MESP $r{=}0.9$ memorize the minority before they generalize (Appendix G).

At weight decay 0.4, memorization at the null is reduced on the binary tasks but not removed: failing two-layer models end at 0.75–0.88 training accuracy against 0.50 held out, more than the one-layer models' partial memorization at weight decay 0.1, and 2L, 2H still generalizes at the null in only 1/30 (MESP) and 0/30 (FESP) seeds. The ternary tasks still memorize at 2L, 2H (0.98). Informative ratios still generalize at this weight decay, so the null neither memorizes fully nor finds the rule, which counts against memorization as the only blocker; but stronger regularization also slows memorization itself (12 of 29 MESP 2L, 2H seeds reach 0.9, at a median of 11,300 epochs), so this is not a clean isolation.

J Linear Decodability of the True Label

Section 5 focuses on attention circuits, but a more basic question remains: in a model that fails, does the true rule fail to *form*, or does it form and go *unused*? The distinction matters because last-layer retraining recovers robust accuracy precisely when the core feature is already linearly present in the representation and the classifier is merely weighting it wrongly (Kirichenko et al., 2023). If sum parity were linearly decodable in our shortcut-trapped models, "the circuit never formed" would be the wrong description of them, and a single retrained readout would fix the problem.

Method. We probe the exact representation the output head consumes: `resid_post` of the final layer at the final (`END`) token position, which the bias-free linear readout maps to logits. For each of the 360 stored MESP checkpoints at weight decay 0.1 we pool the val and adv splits (4,096 examples), fit an ℓ_2-regularized logistic regression on a stratified 70% split, and report accuracy on the held-out 30%. We fit two probes: one for the **true label** (sum parity) and one for the **shortcut label** (max-element parity). Pooling val with adv deliberately gives the probe an easier job than the

Table 28: Memorization at the null against generalization at informative ratios, per two-layer configuration. "Final train" is the mean final training accuracy of failing null-ratio seeds; "≥ 0.9 train" the first logged epoch with training accuracy at least 0.9, median [interquartile range] over the seeds that reach it (count of failing seeds in parentheses). "Gen. epoch" is the first logged epoch at 100% adversarial accuracy, median [interquartile range] over generalizing seeds (count in parentheses), at the lower and higher informative ratio (r=0.7, 0.9 binary; 0.15, 0.50 ternary). Regenerated by `python scripts/timing_analyses.py memo-timing`.

			Null ratio, failing seeds		Gen. epoch, informative ratios	
Task	wd	Config	Final train	≥ 0.9 train	lower r	higher r
MESP	0.1	2L, 1H	0.932	6,400 [4,175–10,900] (24/27)	800 [475–5,125] (8)	1,400 [1,150–6,650] (7)
MESP	0.1	2L, 2H	0.992	900 [800–1,075] (30/30)	300 [200–400] (13)	3,150 [300–10,700] (16)
FESP	0.1	2L, 1H	0.915	9,300 [6,300–11,200] (21/30)	3,500 [800–7,000] (16)	3,700 [1,175–7,150] (20)
FESP	0.1	2L, 2H	0.991	800 [725–1,100] (30/30)	300 [275–325] (20)	1,400 [500–10,000] (23)
FE-Mod3	0.1	2L, 1H	0.995	1,450 [1,025–2,175] (30/30)	6,100 [400–9,300] (25)	10,900 [5,325–12,800] (24)
FE-Mod3	0.1	2L, 2H	0.996	450 [400–500] (30/30)	400 [275–9,400] (28)	400 [300–10,900] (27)
ME-Mod3	0.1	2L, 1H	0.995	2,250 [1,625–3,175] (30/30)	12,500 [12,400–12,550] (3)	— (0)
ME-Mod3	0.1	2L, 2H	0.989	500 [500–600] (29/29)	13,800 [13,250–14,375] (8)	11,700 [10,300–11,800] (5)
MESP	0.4	2L, 1H	0.763	— (0/28)	3,400 [500–4,200] (7)	6,100 [2,250–9,300] (11)
MESP	0.4	2L, 2H	0.875	11,300 [8,400–13,275] (12/29)	300 [200–400] (17)	650 [400–825] (12)
FESP	0.4	2L, 1H	0.747	— (0/28)	3,000 [1,400–5,650] (15)	3,650 [1,200–6,000] (16)
FESP	0.4	2L, 2H	0.875	10,300 [9,375–11,775] (10/30)	300 [300–400] (22)	1,600 [750–4,125] (20)
FE-Mod3	0.4	2L, 2H	0.982	950 [800–1,600] (30/30)	300 [300–600] (17)	300 [300–375] (18)
ME-Mod3	0.4	2L, 2H	0.981	2,000 [800–2,700] (30/30)	— (0)	300 (1)

Table 29: Final training accuracy of *non-generalizing* seeds against held-out accuracy (wd = 0.1), pooled over head counts. "Anti-train $\geq$" is the lower bound on accuracy over the anti-shortcut training examples implied by training accuracy, $(\text{train} - r)/(1 - r)$, assuming perfect accuracy on the shortcut-consistent ones. Two-layer failures fit the training set, including its anti-shortcut examples, while held-out accuracy stays at chance (null) or low (informative ratio). One-layer failures do not.

Task	r	Depth	n	Train	Anti-train $\geq$	Held-out SC	Held-out adv
MESP	0.5	1L	47	0.677	—	0.500	0.506
MESP	0.5	2L	57	0.964	—	0.500	0.503
MESP	0.9	1L	57	0.907	0.066	0.989	0.011
MESP	0.9	2L	37	0.990	0.904	0.906	0.127
FESP	0.5	1L	58	0.635	—	0.492	0.502
FESP	0.5	2L	60	0.953	—	0.496	0.502
FESP	0.9	1L	57	0.902	0.021	0.995	0.007
FESP	0.9	2L	17	0.984	0.836	0.953	0.471
FE-Mod3	0.33	1L	49	0.588	—	0.340	0.348
FE-Mod3	0.33	2L	60	0.996	—	0.325	0.333
ME-Mod3	0.33	1L	58	0.560	—	0.333	0.336
ME-Mod3	0.33	2L	59	0.992	—	0.333	0.336

model had—it sees anti-shortcut examples during fitting—so probe accuracy near chance is a strong negative result.

Table 30: Linear decodability of the true label and the shortcut label from the readout representation, MESP at weight decay 0.1, 30 seeds per cell, probes fitted on val+adv pooled and scored on held-out data. Chance is 0.500 for the true label (the pooled split is exactly balanced) and 0.563 for the shortcut label. Generalizing seeds are separated out because their probe accuracy is 1.000 by construction, since the model itself is perfect on both splits.

			Non-generalizing seeds		
Config	r	n gen.	True label	Shortcut label	True (gen. seeds)
1L, 1H	0.5	10	0.510 ± 0.017	0.678	1.000
1L, 1H	0.7	2	0.494 ± 0.014	0.992	1.000
1L, 1H	0.9	0	$\mathbf{0.484 \pm 0.008}$	**0.999**	—
1L, 2H	0.5	3	0.507 ± 0.016	0.728	1.000
1L, 2H	0.7	3	0.502 ± 0.012	0.993	1.000
1L, 2H	0.9	3	0.496 ± 0.013	0.999	1.000
2L, 1H	0.5	3	0.499 ± 0.015	0.732	1.000
2L, 1H	0.7	8	0.505 ± 0.012	0.983	1.000
2L, 1H	0.9	7	0.518 ± 0.106	0.990	1.000
2L, 2H	0.5	0	$\mathbf{0.503 \pm 0.010}$	**0.747**	—
2L, 2H	0.7	13	0.497 ± 0.018	0.990	1.000
2L, 2H	0.9	16	0.497 ± 0.011	0.996	1.000

The true rule does not form. In every non-generalizing cell the true label probes at 0.484–0.518 against a chance of 0.500, while the shortcut label probes at 0.983–0.999 wherever a shortcut circuit exists ($r \geq 0.7$). The shortcut-trapped 1L, 1H model at r=0.9 is the extreme case: max-element parity is decodable at 0.999 and sum parity at 0.484. Sum parity is therefore not present-but-unused in these models; it is absent from the representation the readout sees, to within the resolution of a linear probe fitted with the adversarial split in hand. Last-layer retraining should therefore not be able to recover it, since there is no direction to reweight, and the next paragraph confirms this. This matches the circuit result: the failing models never build a parity feature in the first place.

Two further points. First, the result is uniform across the sweep: no non-generalizing cell has a mean probe more than 0.018 from chance, including the failing two-layer models at r=0.9 (2L, 2H at 0.497). The 2L, 1H cell still carries a large spread (0.518 ± 0.106), driven by a few seeds, but its mean is within 0.02 of chance. Second, at r=0.5 the shortcut label itself probes at only 0.678–0.747, so the balanced condition fails to build *either* feature cleanly, consistent with the QK fingerprint showing no strong shortcut circuit there (Section 5.3).

Last-layer retraining does not recover robustness. A probe measures decodability; the intervention that matters practically is deep feature reweighting, which freezes the featurizer and refits only the readout on a group-balanced set (Kirichenko et al., 2023). We ran it on the same checkpoints: the two shortcut-consistency groups play the role of the (label, attribute) groups, half of each split is used for reweighting and half held back, and the two reweighting halves are subsampled to equal size so the refit sees both groups equally. Only `output_proj` is refit.

DFR does not recover adversarial accuracy in any non-generalizing cell: it lands between 0.452 and 0.526 against a chance of 0.500, and worst-group accuracy between 0.446 and 0.508. Every cell mean is within 0.05 of chance. Across all 292 non-generalizing seeds only 1 exceeds 0.9 after reweighting, a seed whose stored checkpoint is already within 0.1% of 100% adversarial accuracy, at an epoch the 100-epoch logging grid missed—the best-checkpoint tracker runs at a finer interval than the logged CSV, so our anytime rate is a very slight undercount rather than an overcount.

The shortcut-trapped row is the clearest case. At 1L, 1H r=0.9 the model starts at 0.997 on the shortcut-consistent split and 0.002 on the adversarial one, and reweighting moves it to 0.492 and 0.526. Last-layer retraining removes the *systematic* error—a model that was reliably wrong becomes merely uninformative—but cannot supply a feature the representation does not contain, so both splits collapse toward chance together rather than the adversarial split rising. This is what the probe

Table 31: Deep feature reweighting on the stored MESP checkpoints (wd = 0.1), restricted to the *non-generalizing* seeds of each cell, which are the ones where recovery would matter. "pre" is the model's own readout on the held-back halves; "DFR" is after refitting the readout on the group-balanced set. Worst-group is min over the two splits. Generalizing seeds are omitted because they are already at 1.000 before and after, which serves as the positive control that the procedure itself works.

Config	r	n	pre val	pre adv	DFR val	DFR adv	DFR worst
1L, 1H	0.5	20	0.473	0.543	0.521	0.492 ± 0.039	0.475
1L, 1H	0.7	28	0.958	0.037	0.569	0.452 ± 0.050	0.446
1L, 1H	0.9	30	**0.997**	**0.002**	**0.492**	$\mathbf{0.526 \pm 0.046}$	**0.471**
1L, 2H	0.5	27	0.484	0.538	0.515	0.495 ± 0.033	0.481
1L, 2H	0.7	27	0.836	0.154	0.537	0.485 ± 0.030	0.479
1L, 2H	0.9	27	0.974	0.029	0.505	0.522 ± 0.036	0.485
2L, 1H	0.5	27	0.487	0.533	0.512	0.499 ± 0.029	0.484
2L, 1H	0.7	22	0.692	0.311	0.539	0.487 ± 0.018	0.486
2L, 1H	0.9	23	0.892	0.160	0.548	0.515 ± 0.110	0.508
2L, 2H	0.5	30	0.500	0.527	0.523	0.488 ± 0.026	0.482
2L, 2H	0.7	17	0.652	0.357	0.536	0.485 ± 0.029	0.482
2L, 2H	0.9	14	0.889	0.125	0.530	0.483 ± 0.021	0.482

predicts, and the opposite of what DFR does on benchmarks where the core feature is present but underweighted.

The same intervention at weight decay 0.4. Because weight decay 0.4 is where the transient-generalization and regression behavior of Appendix A occurs, we repeated DFR on those 360 checkpoints as well (Table 32). The conclusion is unchanged and slightly stronger. Across all 301 non-generalizing seeds, reweighting lands adversarial accuracy between 0.445 and 0.556 and worst-group accuracy between 0.444 and 0.489, against a chance of 0.500; only 3 seeds exceed 0.9, the same best-checkpoint artifact described above. The shortcut-trapped row is starker than at weight decay 0.1: 1L, 1H at r=0.9 begins at 0.999 on the shortcut-consistent split and 0.001 on the adversarial one, i.e. almost perfectly anti-correlated with the true label, and reweighting moves it to 0.468 and 0.536, i.e. both splits to chance rather than the adversarial split upward. A readout refit removes the systematic error and supplies nothing in its place, at both weight decay settings.

Table 32: Deep feature reweighting at weight decay 0.4, MESP, restricted to the non-generalizing seeds of each cell; columns as in Table 31. Recomputed with `python -m analysis.checkpoint_audit dfr -task max_parity_sum_parity -weight-decay 0.4`.

Config	r	n	pre val	pre adv	DFR val	DFR adv	DFR worst
1L, 1H	0.5	30	0.485	0.566	0.526	0.518 ± 0.100	0.488
1L, 1H	0.7	30	0.995	0.006	0.548	0.472 ± 0.077	0.447
1L, 1H	0.9	30	**0.999**	**0.001**	**0.468**	$\mathbf{0.536 \pm 0.041}$	**0.456**
1L, 2H	0.5	26	0.482	0.539	0.511	0.495 ± 0.043	0.469
1L, 2H	0.7	28	0.948	0.049	0.564	0.460 ± 0.054	0.453
1L, 2H	0.9	27	0.997	0.040	0.491	0.556 ± 0.097	0.485
2L, 1H	0.5	28	0.479	0.545	0.509	0.499 ± 0.036	0.479
2L, 1H	0.7	23	0.782	0.211	0.571	0.445 ± 0.049	0.444
2L, 1H	0.9	19	0.943	0.116	0.571	0.496 ± 0.128	0.489
2L, 2H	0.5	29	0.488	0.533	0.520	0.489 ± 0.027	0.479
2L, 2H	0.7	13	0.703	0.288	0.559	0.453 ± 0.019	0.453
2L, 2H	0.9	18	0.896	0.112	0.549	0.467 ± 0.034	0.464

Before generalization. For MESP at r=0.9 the per-100-epoch checkpoints let us ask whether sum parity becomes decodable before a seed generalizes. We fit the same probe on the `END` residual after each layer at 500 and 1,000 epochs before each late escaper's escape epoch (Appendix F), for every seed of the cell at the same epoch. After the first layer nothing separates escapers from trapped seeds (at 2L, 2H: 4 of 9 and 5 of 9 escapers above the trapped median). After the final layer the escapers

are slightly ahead, 0.516 against 0.494 at 1,000 epochs (7 of 9, p=0.18) and 0.531 against 0.494 at 500 (8 of 9, p=0.04), with no escaper above 0.61. That is at most a faint signal a few points above chance, and at 500 epochs 8 of those 9 escapers' adversarial accuracy has already risen above that of trapped seeds (Appendix F). Regenerated by `python -m analysis.trajectory_audit lead-probe`.

K Interventions: Training-Set Size, Transplant and Minority Upweighting

This appendix reports three interventions on MESP 2L, 2H at weight decay 0.1, together with a null-ratio control, all at 10 seeds (Table 33). Each targets a question the observational analyses leave open: whether memorization is what blocks the rule at the null ratio, and whether the benefit of an informative shortcut needs the minority gradient asymmetry (H1) or only structure the shortcut leaves behind (H2).

Setup, and comparison across hardware. These runs were trained on one GPU (NVIDIA RTX 5070 Ti Laptop, JAX 0.11.2) rather than on the TPUs used for every other run in the paper. The model, initialization (`PRNGKey(seed)`), optimizer (gradient clipping at 1.0, AdamW with lr=10^{-3} and ϵ=10^{-6}), float64 loss, per-epoch shuffle (`PRNGKey(seed+epoch)`), batch size 4,096 and evaluation every 400 optimizer steps are those of the original runner; the only change is that the seeds of one cell are trained together in a single vectorized program (`scripts/run_owed_jax.py`). Training is therefore deterministic given the seed on a fixed backend, but not bitwise identical across backends. Retraining the MESP 2L, 2H r=0.9 seeds on GPU for the full 15,000 epochs reproduces the TPU regime: every seed sits at adversarial accuracy near 0.001 through Phase 1, and the two seeds that escape by epoch 300 on TPU also escape by epoch 300 on GPU. Individual seeds nonetheless diverge later (seed 1008 generalizes at epoch 800 on GPU and 11,300 on TPU). Generalization rates, not per-seed outcomes, are the reproducible quantity across hardware, and we compare them both with the TPU cells of Table 1 and with a null-ratio control trained on the same GPU.

Table 33: Interventions on MESP 2L, 2H, weight decay 0.1, 10 seeds each, trained on GPU. "Gen." counts seeds reaching 100% adversarial accuracy; in every row the anytime and terminal counts coincide and every generalizing seed is also at 100% on the shortcut-consistent split. The N=16,384 control uses the paper's training file and runs for 16,000 optimizer steps (4,000 epochs), four times the median generalization time of the N=262,144 row. Fisher p is two-sided against the comparison cell named; the larger training set is also p=1.2×10^{-9} against the TPU null (0/30). The transplant starts from epoch-500 checkpoints and trains for the remaining 14,500 epochs.

Condition	r	N	Steps	Gen.	Comparison (Fisher p)
Null control	0.5	16,384	16k	0/10	TPU null: 0/30
Larger training set	0.5	262,144	60k	**10/10**	GPU control 0/10 (1.1×10^{-5})
Transplant from r=0.9	0.5	16,384	58k	**6/10**	TPU null 0/30 (5.5×10^{-5}), cross-hardware
Minority upweighting ($\times$9)	0.9	16,384	60k	**1/10**	TPU r=0.9: 16/30 (0.026), cross-hardware

Training-set size at the null ratio. We retrained MESP 2L, 2H at r=0.5 with N=262,144 training examples (16$\times$ the default, 8% of sequence space) at the same 60,000 optimizer steps, 10 seeds, drawn by the same rule as the default training file (unique maximum, exactly half shortcut-consistent, disjoint from the evaluation splits). All 10 seeds reach 100% adversarial accuracy and are still there at the end of training, against 0/30 at N=16,384 on TPU (Fisher $p \approx 10^{-9}$) and 0/10 in the same-hardware control (p=1.1×10^{-5}), whose seeds instead memorize their training set (0.976 training accuracy, 0.50 held out). They do so within a median of about 4,000 optimizer steps (2,700–6,100), and training accuracy never exceeds held-out accuracy by more than 0.03 beforehand, so rule and fit arrive together, as on n11 (Appendix C). The null-ratio failure is therefore data-limited. Whether memorization causes it remains open: more data also makes the rule better determined by the training set, and within a run at r=0.9 late escapers memorize the minority before generalizing (Appendix G). We trained only the null ratio at N=262,144, so whether informative ratios still help at that size is

unmeasured, and this is one larger N, not a sweep, so where between the two sizes the null stops failing is also unmeasured.

Transplant. We initialized $r{=}0.5$ training from the epoch-500 checkpoints of ten MESP 2L, 2H $r{=}0.9$ seeds that were still at the shortcut posterior (adversarial accuracy < 0.25 through epoch 500), with a fresh AdamW state, and trained for the remaining 14,500 epochs. Six of ten reach 100% adversarial accuracy and hold it, against 0/30 from scratch at $r{=}0.5$ (Fisher $p \approx 5{\times}10^{-5}$). They generalize 100–12,300 epochs after the transplant (median 2,400). Median shortcut-consistent held-out accuracy falls to chance (0.49) within 100 epochs of the transplant, so the rescue does not come from continued use of the shortcut. The four failures memorize their training set (0.94–0.97), as from-scratch null runs do.

The transplant removes a *persisting* minority asymmetry, not the asymmetry itself. At the moment of transfer the model is at the $r{=}0.9$ posterior, with per-example losses of about 2.30 on anti-shortcut and 0.11 on shortcut-consistent examples (Table 11), and at $r{=}0.5$ the two groups have equal counts. Until the shortcut is abandoned, about 100 epochs later, the aggregate update is therefore dominated by the anti-shortcut examples more strongly than at the $r{=}0.9$ steady state. Three further caveats apply. The optimizer state was not saved and so was reset, and a fresh AdamW state makes the first steps close to sign-gradient steps at the full learning rate. The significant comparison is across hardware: the same-hardware control covers 4,000 epochs (counted from the transplant for transplanted seeds, from initialization for controls), and within that window 4 of the 10 transplanted seeds have generalized against 0 of 10 controls ($p{=}0.087$). And five of the same ten seeds generalize when left at $r{=}0.9$ on TPU, against six after the transplant; four seeds generalize under both. Their state at epoch 500 may already determine much of the outcome, consistent with the layer-placement marker (Appendix G), whose epoch-500 value points the same way on these seeds ($p{=}0.17$), although that agreement is largely expected from the overlap with the $r{=}0.9$ outcomes. With these caveats the transplant favors the scaffold account (H2): a briefly learned informative shortcut can leave structure that makes the rule reachable after the shortcut stops being informative. Two controls would separate installed structure from the reset and the transient pulse: from-scratch $r{=}0.5$ runs restarted at epoch 500 with a fresh optimizer (the null's median time to 0.9 training accuracy is 900 epochs, so these have not yet memorized), and a transplant whose `output_proj` is first refit at $r{=}0.5$, so that training starts with balanced losses.

Minority upweighting. Weighting the anti-shortcut examples of MESP $r{=}0.9$ by $r/(1-r){=}9$, so that both groups carry equal loss mass, reduces 2L, 2H generalization from 16/30 on TPU to 1/10 on GPU (Fisher $p{=}0.026$; the GPU retrain of $r{=}0.9$ described above was used to validate the trainer, and its per-seed outcomes were not retained, so the comparison is with the TPU rate); the one generalizing seed does so at epoch 13,400. The failing seeds reach 1.000 training accuracy on both subsets, with held-out adversarial accuracy at 0.16 and held-out shortcut-consistent accuracy at 0.84, i.e. they memorize the upweighted minority. This is the outcome H1 predicts, but it does not discriminate: upweighting to balance also makes the shortcut uninformative in the weighted objective, so H2 predicts it as well. Together with the transplant, the evidence at $N{=}16{,}384$ is that generalization requires the shortcut to have been informative at some point in training, not a persisting gradient asymmetry. At $N{=}262{,}144$ the null generalizes with a shortcut that is never informative.

Regenerating these results. Each row of Table 33 is one command, `python scripts/run_owed_jax.py` with the subcommand `nsweep` (`-n 16384 -steps 16000` and `-n 262144`), `transplant` (`-ratio 0.5 -src-epoch 500 -seed-list 1001 ... 1010`) or `upweight` (`-ratio 0.9`), each with `-seeds 10` where seeds are not listed; the queue that ran them is `experiments/owed/queue.sh`, and each run writes its per-seed log and summary under `experiments/owed/`. The epoch-500 layer-lead check is `python -m analysis.trajectory_audit transplant-lead`.

L Scope: Comparisons and Controls We Did Not Run

The questions below require new training runs. For each we state what the competing accounts predict.

Extending the interventions. Each intervention is a single configuration at 10 seeds. Three controls bear most directly on the claims. The first is $r{=}0.9$ at $N{=}262{,}144$: if informative ratios still help there, the effect is not confined to data-limited training; if they are neutral, the practical claim holds only in that regime. The second is a reset-only control for the transplant: from-scratch $r{=}0.5$ runs restarted at epoch 500 with a fresh optimizer. If these also generalize, the transplant says nothing about the $r{=}0.9$ history. The third is a transplant whose readout is first refit at $r{=}0.5$, which removes the transient gradient imbalance at the moment of transfer. Further extensions are an N sweep between $16{,}384$ and $262{,}144$ to locate where the null stops failing, and whether the informative-ratio benefit shrinks with it; transplants from several source epochs, including one before the shortcut posterior is reached; a transplant that carries the optimizer state; and all three on FESP and FE-Mod3.

Binary ratios below the null. "Either direction" is shown only for ternary labels. Binary $r \in \{0.1, 0.3\}$ would test whether the effect is symmetric about $1/2$, or specific to a binary shortcut-consistent majority.

Group-robustness baselines. Minority upweighting at $r{=}0.9$ removes most of the benefit of high r (Appendix K), but it also makes the shortcut uninformative in the weighted objective and so does not separate H1 from H2. Subsampling the majority to $r{=}0.5$, at the reduced sample size subsampling actually produces, tests whether the null-ratio failure survives with fewer examples to memorize. Group DRO (Sagawa et al., 2020a) and JTT (Liu et al., 2021) are cheap at this scale and belong alongside both.

Architecture and optimization controls at the null ratio. The 0/30 at 2L, 2H, $r{=}0.5$ may be partly an optimization pathology of our architecture. The controls are: pre-LayerNorm, biases, a learning-rate sweep around 10^{-3}, and a $3\times$ longer horizon. 15,000 epochs at $r{=}0.5$ show no upward drift in adversarial accuracy for this cell, but its models have already memorized the training set by then (Appendix I); with $16\times$ the data the same cell generalizes in 10/10 seeds (Appendix K), so the architecture can find the rule at the null, and these controls bear on the data-limited regime only. A width sweep at fixed depth, and a three-layer point, would say whether the "capacity threshold" is about depth or about capacity. The sharpest version is a one-layer model with a much wider MLP, or two heads at a much larger d_{model}: if it memorizes at the null and the ratio benefit appears, the threshold is the expressivity of the one-layer readout path rather than depth.

Preconditioned updates. Our subset gradients are raw gradients, on held-out splits except for MESP at $r{=}0.9$, where the per-100-epoch checkpoints let us compute them on the training subsets (Table 19). The optimizer state was not saved, so the per-subset Adam-preconditioned update $m/\sqrt{v}$, which is what actually moves the weights, needs new runs.

Transfer to natural benchmarks. Our tasks admit an exact finite rule, but the two-layer failures show that memorization is available at this scale (Appendix I). On Waterbirds, CelebA or MultiNLI, overparameterized models fit the minority by memorizing atypical examples (Sagawa et al., 2020b), and that memorization localizes to small groups of neurons (You et al., 2025). Whether an informative shortcut tips training toward reorganization rather than memorization at scale is the key question for transfer. The two are distinguishable in principle: reorganization predicts that the readout's reliance on the shortcut feature *decreases* as minority loss falls, while memorization predicts that it is preserved and the minority is fit around it. Reliance has to be measured through the readout, for instance by patching; attention to the shortcut token does not measure it (Section 5.3).

M Per-Seed Subset Trajectories

Section 5.4 summarizes the two-phase dynamic; this appendix gives the per-seed trajectories behind it.

Figure 2 plots accuracy separately on the shortcut-consistent (SC, val split at $r{=}1$) and anti-shortcut (adv split at $r{=}0$) subsets for all 30 seeds at $r{=}0.9$, one panel per architecture. In both 1-layer models (panels a–b), SC accuracy rises to near-perfect within the first logged epoch while anti-shortcut accuracy stays near zero throughout—well below the 0.5 chance line, and training remains in Phase 1 for the full 15,000 epochs in all but three seeds, all at 1L, 2H.

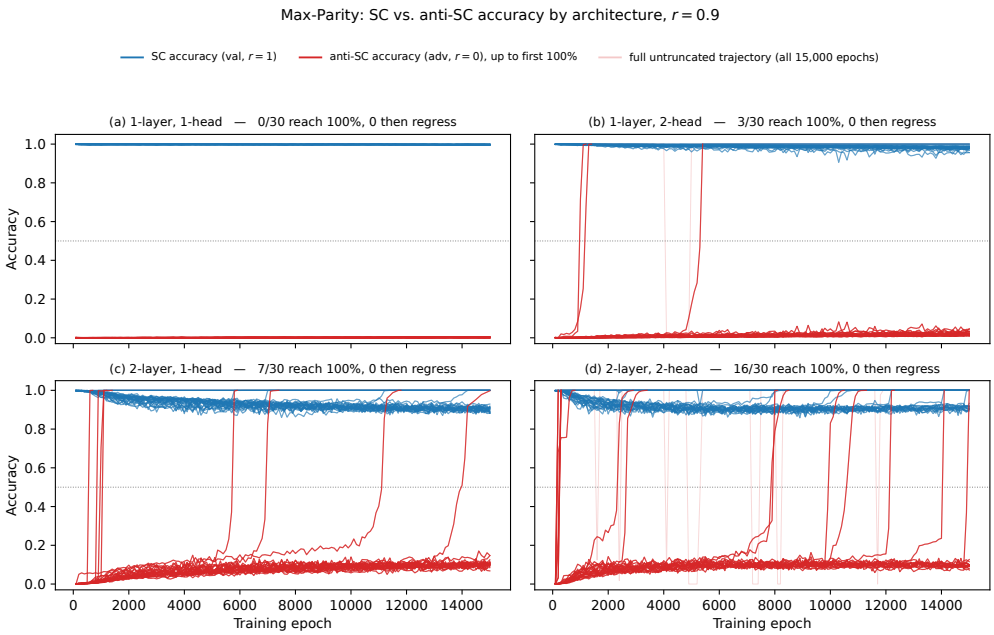


Figure 2: Per-epoch SC accuracy (blue, val split $r{=}1$) and anti-SC accuracy (red, adv split $r{=}0$) for all 30 seeds at $r{=}0.9$, weight decay 0.1. Solid anti-SC curves are drawn up to the first epoch reaching 100%, so a solid curve ending at the top of a panel marks a seed that generalized; *the faint traces behind them are the full untruncated 15,000-epoch trajectories*, so post-generalization regression is visible rather than hidden. Each panel title gives how many seeds reach 100% and how many of those fall back before the end of training—none of 7 in panel (c) and none of 16 in panel (d), matching the identical anytime and terminal columns of Table 1 at $r{=}0.9$. **(a)** 1L, 1H: all seeds trapped in Phase 1 (best anti-SC accuracy any seed reaches is 0.005). **(b)** 1L, 2H: similarly trapped; three seeds reach 100% and are there at the end. No non-generalizing seed in this panel exceeds 0.083. The faint vertical excursions in (b) and (d) are *generalizing* seeds that collapse briefly after reaching 100% and recover: one seed in (b), for nine logged epochs, and four in (d), for one to eight. **(c)** 2L, 1H: 7/30 show Phase 2 recovery. **(d)** 2L, 2H: 16/30, with seed-dependent onset.

In 2-layer models (panels c–d), the same Phase 1 pattern appears initially: SC accuracy reaches near-perfect while anti-shortcut accuracy collapses toward zero. Phase 2 then begins at a seed-dependent epoch, ranging from a few hundred to over 10,000 epochs: anti-shortcut accuracy rises toward 1.0 while SC accuracy remains near-perfect. Shortcut-consistent accuracy stays high through Phase 2, but this does not show the shortcut circuit is retained, since a model implementing the true rule is also correct on those examples; the measurements of Section 5.3 show attention leaving the max token only after generalization. A few seeds show brief SC dips during Phase 2, visible as downward spikes in panels (c–d); these recover in every generalizing seed, all of which end at SC accuracy 1.000. Mean final SC accuracy over all 30 seeds is 0.927 (2L 1H) and 0.958 (2L 2H), lowered by the non-generalizing seeds (0.904 and 0.910). The Phase 2 onset is abrupt for some seeds and gradual for others, consistent with the wide seed-to-seed variance in generalization epochs. Head count modulates the fraction of seeds that enter Phase 2: 7/30 for 2L 1H (panel c) and 16/30 for 2L 2H (panel d).

N Statistical Tests and Threshold Robustness

Table 34 gives 95% Wilson intervals and two-sided Fisher exact tests for every comparison of generalization rates that the text relies on (anytime metric, weight decay 0.1 unless noted, 30 seeds per cell). The informative-versus-null comparisons that carry the paper's claims are significant at $p \leq 10^{-3}$ or better on MESP, FESP and FE-Mod3, and the one- versus two-layer comparison at the null is significant on MESP and FE-Mod3. On FESP one-layer models are not better at the null (1/30

against 0/30). On ME-Mod3 the ratio comparison is weak (p=0.026) and probably right-censored by the training horizon (Appendix D). Three comparisons are not significant, and we draw no conclusion from them: FESP against MESP at r=0.9, r=0.9 against 0.7 on MESP, and the weight-decay-0.4 ordering of r=0.7 and 0.9.

Table 34: Generalization-rate comparisons with 95% Wilson intervals and two-sided Fisher exact p (anytime metric, wd = 0.1 unless noted).

Comparison	A	B	p
MESP 2L, 2H: r=0.9 vs. 0.5	16/30 [0.36, 0.70]	0/30 [0.00, 0.11]	1.9×10^{-6}
MESP 2L, 2H: r=0.7 vs. 0.5	13/30 [0.27, 0.61]	0/30 [0.00, 0.11]	4.6×10^{-5}
MESP 2L, 2H: r=0.9 vs. 0.7	16/30 [0.36, 0.70]	13/30 [0.27, 0.61]	0.61
MESP 1L, 1H: r=0.5 vs. 0.9	10/30 [0.19, 0.51]	0/30 [0.00, 0.11]	8×10^{-4}
MESP r=0.5: 1L, 1H vs. 2L, 2H	10/30 [0.19, 0.51]	0/30 [0.00, 0.11]	8×10^{-4}
FESP 2L, 2H: r=0.9 vs. 0.5	23/30 [0.59, 0.88]	0/30 [0.00, 0.11]	1.7×10^{-10}
2L, 2H r=0.9: FESP vs. MESP	23/30 [0.59, 0.88]	16/30 [0.36, 0.70]	0.10
2L, 1H r=0.7: FESP vs. MESP	16/30 [0.36, 0.70]	8/30 [0.14, 0.44]	0.064
FE-Mod3 2L, 2H: r=0.15 vs. 0.33	28/30 [0.79, 0.98]	0/30 [0.00, 0.11]	8.4×10^{-15}
FE-Mod3 2L, 2H: r=0.50 vs. 0.33	27/30 [0.74, 0.97]	0/30 [0.00, 0.11]	9.2×10^{-14}
FE-Mod3 r=0.33: 1L, 1H vs. 2L, 1H	11/30 [0.22, 0.54]	0/30 [0.00, 0.11]	3.2×10^{-4}
ME-Mod3 r=0.50: 1L, 1H vs. 2L, 1H	15/30 [0.33, 0.67]	0/30 [0.00, 0.11]	5.8×10^{-6}
ME-Mod3 2L, 2H: r=0.15 vs. 0.33	8/30 [0.14, 0.44]	1/30 [0.01, 0.17]	0.026
MESP wd 0.4, 2L, 2H: r=0.7 vs. 0.9	17/30 [0.39, 0.73]	12/30 [0.25, 0.58]	0.30

Is the 100% threshold brittle? Outcomes are strongly bimodal. On MESP, of the 292 non-generalizing seeds at weight decay 0.1 exactly one ever exceeds 0.75 peak adversarial accuracy (a 2L, 1H r=0.9 seed at 0.998), and the median non-generalizing peak is 0.23. Lowering the threshold to 99%, 95% or 90% changes no MESP count by more than one seed. The threshold matters most on FESP, where 22 non-generalizing seeds exceed 0.9 without reaching 100%. Relaxing it there only strengthens the effect (2L, 2H at r=0.9: 23, 28, 28, 28 at 100, 99, 95, 90%; at r=0.5: 0 at every threshold). On the ternary tasks no null-ratio two-layer count changes at any threshold. Every two-layer null cell stays at its reported count under all four thresholds; the only null cell that moves is FE-Mod3 1L, 1H (11 to 12).

This appendix, Tables 29, 11, 18 and 6, the $|\rho|$ values and the ME-Mod3 QK test are regenerated from logged per-seed files by `python scripts/seed_analyses.py` with the subcommands `stats`, `thresholds`, `memorization`, `shortcut-floor`, `gradient-outcome`, `fesp-seq0`, `ov-abs` and `memod3-qk`. Tables 19 (held-out rows) and 28, the n11 memorization result and the attention statistic of Table 6 are regenerated by `python scripts/timing_analyses.py` (subcommands `memo-timing`, `n11`, `time-matched`, `attn-unified`, `excursions`, `nongen-losses`). The measurements that need the per-100-epoch MESP r=0.9 checkpoints (Tables 19 training rows, 23 and 24, and the fixed-lead probe) are regenerated by `python -m analysis.trajectory_audit` (`stored-epoch`, `pmax`, `train-grad`, `lead-grad`, `lead-probe`, `within-run`).

O Computational Resources

All experiments were conducted using TPU v5-e8 accelerators provided via Kaggle. A single experimental configuration required approximately 7 hours of wall-clock time per 10 seeds to complete the 15,000-epoch training schedule; the two primary binary tasks and the two mod 3 tasks were each extended to 30 seeds per configuration. The interventions of Appendix K were run on one NVIDIA RTX 5070 Ti Laptop GPU, training all 10 seeds of a condition together; each 60,000-step condition took about 23 minutes.