UNDERSTANDING AND IMPROVING GROKKING THROUGH CROSS-SAMPLE GRADIENT INTERACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Grokking exposes a long delay between fitting training data and generalizing. Gradient-norm regularization can alter this delay, but the norm of the averaged gradient does not distinguish small per-sample gradients from cancellation between them. We decompose the squared batch-gradient norm into individual squared norms and positive and negative pairwise interactions. Negative interactions correspond to first-order increases in one sample's loss under another sample's update, yet enter norm penalties with a minus sign. We introduce Gradient Conflict-Aware Minimization (GCAM), which penalizes a coordinate-wise upper bound on these interactions using per-coordinate sign aggregation. In modular-addition Transformers, GCAM applied to attention value and output projections accelerates generalization, enables runs that fail under baseline training to generalize within the same budget, and nearly eliminates the memorization–generalization delay at narrow width. Beyond grokking, GCAM improves a vision Transformer's CIFAR-10 accuracy without augmentation from 66.6% to 72.0% under the evaluated learning-rate choices.

1 INTRODUCTION

In grokking, a network fits its training set early but reaches high test accuracy only much later (Power et al., 2022). Existing accounts describe the circuits the network forms and how its weight norm changes (Nanda et al., 2023; Liu et al., 2023); we ask how the per-sample gradients that the optimizer averages interact before the network generalizes, and whether acting on that interaction changes when it does.

Gradient-norm regularization provides a starting point: fixed-step gradient descent has an implicit squared-norm term (Barrett & Dherin, 2021), explicit gradient regularization (EGR) adds one directly, and SAM's local worst-case objective has a first-order gradient-norm term (Foret et al., 2021; Zhao et al., 2022). In our full-batch grokking experiments, EGR advances generalization, whereas raw-gradient Runge–Kutta probes (Dherin et al., 2025) do not (Section 3). These controls motivate examining the sample interactions hidden in the aggregate norm.

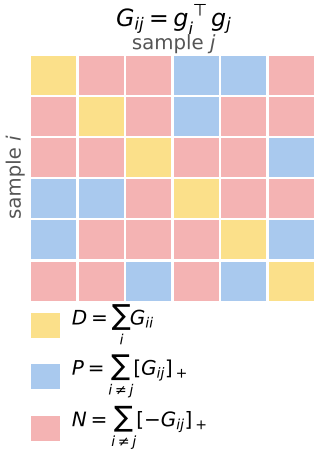


Figure 1: Per-sample Gram matrix split into D, P, N.

The batch gradient $\bar{g}$ is the average of the per-sample gradients g_i over a batch of B samples, and averaging hides two things: how large each g_i is, and whether pairs of samples push the parameters in the same or opposite directions. Expanding the square, as in gradient diversity (Yin et al., 2018), and splitting the pairwise inner products by sign separates them, $B^2\|\bar{g}\|^2 = D + P - N$, with summed squared per-sample gradient norms D and agreeing and conflicting pairs P and N (Figure 1), so the same small norm can come from small per-sample gradients or from large agreeing and conflicting pairs that cancel.

Each pair in N has a direct meaning: to first order, a small gradient step on one sample raises another sample's loss in proportion to their negative inner product (Fort et al., 2019), and within a batch the same pairs cancel part of the average descent. Norm regularization counts N with a minus sign: at fixed D and P, more conflict makes the penalty smaller (Section 4). We therefore introduce Gradient

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.

Conflict-Aware Minimization (GCAM; Section 5), which penalizes a coordinate-wise upper bound N_c on N with a plus sign. The attention value and output projections (V,O) account for most of N_c early in training; regularizing their gradient coordinates alone improves the measured efficiency over all-parameter regularization (Section 6).

Our contributions are:

- a decomposition connecting the batch-gradient norm to first-order interference between samples;
- GCAM, a selective interference penalty with exact coordinate aggregation, and an empirical study of its placement;
- paired-seed evidence of earlier generalization and recovery of baseline-failing runs, near-elimination of the delay at narrow width, and an extension to CIFAR-10 (Section 7).

2 Related work

Grokking. Delayed generalization on algorithmic tasks (Power et al., 2022) has been explained by structured representations and circuits (Liu et al., 2022; Nanda et al., 2023), a lazy-to-rich transition (Kumar et al., 2024) and weight-norm dynamics (Liu et al., 2023; Boursier et al., 2025); without explicit regularization, a floating-point collapse of the softmax can prevent it (Prieto et al., 2025), and adaptive optimizers show late-training instabilities (Thilak et al., 2022). Methods that shorten the delay amplify slow gradient components (Lee et al., 2024), remove the gradient's radial component (Prieto et al., 2025), equalize progress along principal directions (Saheb Pasand & Dohmatob, 2026), learn a gradient transformation (Zhou et al., 2025), align the input–output Jacobian with the data (Walker et al., 2025) or penalize curvature through weight noise (Zhang et al., 2024); GCAM instead penalizes interactions between per-sample gradients.

Gradient-norm regularization. Gradient-norm penalties have been made cheaper by finite differences (Karakida et al., 2023). SAM has been analysed through the largest Hessian eigenvalues along its trajectory (Agarwala & Dauphin, 2023) and the low-rank features it produces (Andriushchenko et al., 2023). For stochastic gradient descent the implicit penalty splits into the squared norm of the full-batch gradient and the trace of the per-sample gradient covariance (Smith et al., 2021), and penalizing per-sample gradient norms directly has been analysed (Meng et al., 2023). These works act on the norm or on per-sample gradient magnitudes; none separates agreeing from conflicting pairs of samples.

Per-sample agreement and conflict. Gradient diversity (Yin et al., 2018) and coherence (Chatterjee & Zielinski, 2022) compare D with $D + P - N$, in which agreeing and conflicting pairs cancel. Gradient confusion bounds the most negative pairwise inner product (Sankararaman et al., 2020), and stiffness is the average sign or cosine of pairwise inner products (Fort et al., 2019). Groups of outlier samples with opposing gradients drive sharpening and loss spikes, with connections to grokking and SAM (Rosenfeld & Risteski, 2024). MSQ is an Adam variant whose preconditioner keeps the squared mean of per-sample gradients and drops their variance (Roulet & Agarwala, 2025); Section 5 writes it in terms of P_c and N_c. Coordinate-wise treatments of disagreement clip extreme per-sample values (Chatterjee, 2020), mask coordinates whose signs disagree across environments or examples (Parascandolo et al., 2021), or drop gradient coordinates by sign purity across tasks (Chen et al., 2020); conflict between tasks is also handled by projection (Yu et al., 2020), by constraints against earlier tasks (Lopez-Paz & Ranzato, 2017), and appears as a conflict term in the implicit regularizer of multitask learning (Dherin, 2023). GCAM instead adds a differentiable penalty on the coordinate-wise negative interactions between samples of one task.

3 Gradient-norm regularization in a grokking setting

Setting and notation. As in Nanda et al. (2023) but with modulus 97 and pre-layer normalization, a one-layer Transformer (width 128) learns $(a + b) \bmod 97$ from a fixed random 30% of all pairs with cross-entropy and AdamW (Loshchilov & Hutter, 2019) (learning rate $\eta = 10^{-3}$, weight decay 1), with minibatches of 256 samples unless full-batch training is stated; we call this the *reference*

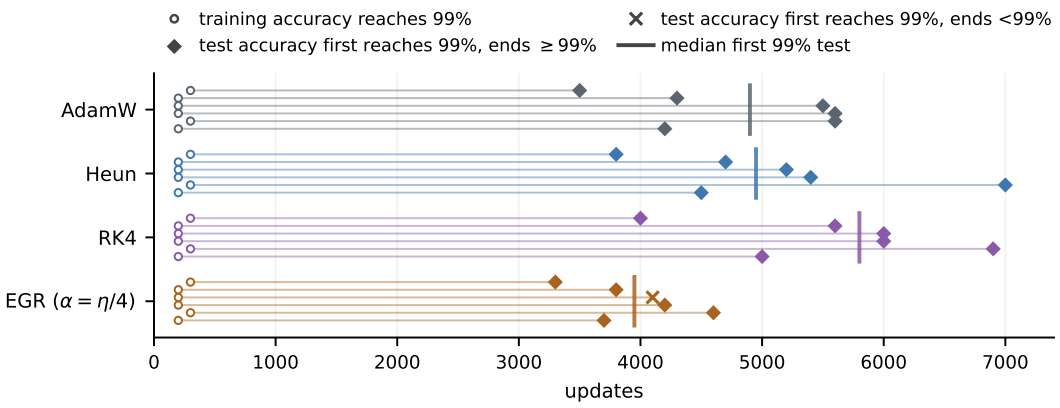


Figure 2: Adding the gradient of the norm penalty before AdamW (EGR, $\alpha = \eta/4$) makes generalization earlier on every seed, and subtracting it with RK4 makes it later; Heun stays close to AdamW. Reference setting with full-batch training, one row per seed.

setting. Accuracy is evaluated every 100 updates; t_{mem} and t_{99} are the first evaluations at which training and test accuracy reach 99%, marking when the network fits the training set and when it generalizes. Seeds are 42–47 unless stated; Appendix B gives budgets and the full protocol.

Gradient-norm regularization. For a batch of B samples with losses ℓ_i, let $L = B^{-1}\sum_i \ell_i$, $g_i = \nabla_\theta \ell_i$, $\bar{g} = B^{-1}\sum_i g_i$, $Q = \|\bar{g}\|^2$ and $H = \nabla^2_\theta L$. To first order in η, gradient descent follows the gradient flow of the modified loss $L + \frac{\eta}{4}Q$ (Barrett & Dherin, 2021). Explicit gradient regularization (EGR) penalizes the norm directly, minimizing $L + \alpha Q$ with gradient $\bar{g} + 2\alpha H\bar{g}$ (Barrett & Dherin, 2021; Karakida et al., 2023), and SAM's worst-case loss increase within radius ρ is $\rho\sqrt{Q} + O(\rho^2)$ (Foret et al., 2021; Appendix A). Heun's method and the classical fourth-order Runge–Kutta method (RK4) (Hairer et al., 2006; Dherin et al., 2025), using step parameter η and the raw gradient of the same batch at every stage, pass AdamW the gradient $\bar{g} - \frac{\eta}{2}H\bar{g} = \nabla_\theta(L - \frac{\eta}{4}Q)$ to first order, while EGR with $\alpha = \eta/4$ passes $\nabla_\theta(L + \frac{\eta}{4}Q)$: Heun and RK4 subtract the norm term from the gradient AdamW receives, and EGR adds it.

Effect on generalization. All four methods fit the training set within 200–300 updates. EGR advances generalization on all six seeds, reducing median t_{99} from 4,900 to 3,950 updates; RK4 delays it, whereas Heun remains close to AdamW (Figure 2). We next decompose the aggregate norm into individual gradient scale and sample interactions.

4 What the Gradient Norm Hides

Decomposition. Expanding $B^2Q = \|\sum_i g_i\|^2$ (Yin et al., 2018) and splitting the pairwise inner products by sign gives, with sums over ordered pairs $i \neq j$,

$$B^2Q = D + P - N, \qquad D = \sum_i \|g_i\|^2, \quad P = \sum_{i\neq j} \left[g_i^\top g_j\right]_+, \quad N = \sum_{i\neq j} \left[-g_i^\top g_j\right]_+, \tag{1}$$

which splits the Gram matrix $G_{ij} = g_i^\top g_j$ into its diagonal (D) and its off-diagonal entries by sign (P, N; Figure 1). A small norm therefore implies neither small per-sample gradients nor little conflict: if $B = 2$ and $g_2 = -g_1 \neq 0$, the batch gradient vanishes while $D = N = 2\|g_1\|^2$.

Two readings of the negative term. For a single pair, a gradient step of size τ on sample j changes the loss of sample i by

$$\ell_i(\theta - \tau g_j) - \ell_i(\theta) = -\tau\, g_i^\top g_j + O(\tau^2) \tag{2}$$

(Fort et al., 2019), so $\tau\left[-g_i^\top g_j\right]_+$ is, to first order, the increase in sample i's loss caused by a step on sample j, and N sums this local interference over all pairs. Across tasks, constraints on this

interference limit forgetting (Lopez-Paz & Ranzato, 2017). For the batch, a gradient step changes the batch loss by

$$L(\theta - \eta\bar{g}) - L(\theta) = -\eta Q + O(\eta^2) = -\frac{\eta}{B^2}\big(D + P - N\big) + O(\eta^2), \tag{3}$$

so the same pairs cancel part of the descent that D and P would otherwise provide, the pairwise cancellation that gradient confusion bounds (Sankararaman et al., 2020).

The sign of the negative term. Substituting (1) into $L + \frac{\eta}{4}Q$ (Section 3) gives

$$L + \frac{\eta}{4}Q = L + \frac{\eta}{4B^2}\big(D + P - N\big), \tag{4}$$

and likewise for EGR with α in place of $\eta/4$; the first-order term of SAM, $\rho\sqrt{Q}$, also increases with Q. A regularizer aimed at the interference of (2) would count N with a plus sign. In the objective, norm regularizers count it with a minus sign, whatever a training step does to N: at fixed D and P, more conflict lowers the penalty. Section 5 reverses the sign of N alone, penalizing a coordinate-wise upper bound on it. The conflict also carries early information about the outcome: on 128 new seeds of the reference setting (24 of which AdamW does not bring to 99%), a frozen predictor that adds module-wise N_c at update 500 to the training loss and accuracy raises ROC-AUC from .871 to .952, although it does not transfer to width 48 (Appendix F).

5 Gradient Conflict-Aware Minimization

Gradient Conflict-Aware Minimization (GCAM) directly regularizes cross-sample gradient interference through a coordinate-wise upper bound on negative pairwise gradient inner products. It adds the gradient of this penalty to the gradient that the base optimizer receives. Direct Gram-matrix evaluation forms all $B(B-1)/2$ pairwise inner products.

Dot and coordinate forms. To allow the penalty to act on part of the network, let $\mathcal{S}$ be a set of parameter coordinates and h_i equal to g_i with every coordinate outside $\mathcal{S}$ set to zero, $\bar{h} = B^{-1}\sum_i h_i$ and $D_{\mathcal{S}} = \sum_i \|h_i\|^2$. The *dot forms* $P_{\mathcal{S}}$ and $N_{\mathcal{S}}$ are the terms of (1) computed from the h_i (P and N when $\mathcal{S}$ holds all parameters); they split each pair's inner product $h_i^\top h_j$ by sign. The *coordinate forms* split each coordinate's product by sign before summing over coordinates,

$$P_{c,\mathcal{S}} = \sum_{i\neq j}\sum_{k\in\mathcal{S}} [h_{ik}h_{jk}]_+, \qquad N_{c,\mathcal{S}} = \sum_{i\neq j}\sum_{k\in\mathcal{S}} [-h_{ik}h_{jk}]_+, \tag{5}$$

and the *sign sums* collect the positive parts and the negative magnitudes of each coordinate over samples, $a_k = \sum_i [h_{ik}]_+$ and $b_k = \sum_i [-h_{ik}]_+$.

Proposition 1 (Appendix A) shows that

$$B^2\|\bar{h}\|^2 = D_{\mathcal{S}} + P_{c,\mathcal{S}} - N_{c,\mathcal{S}}, \qquad N_{c,\mathcal{S}} - N_{\mathcal{S}} = P_{c,\mathcal{S}} - P_{\mathcal{S}} \geq 0, \qquad N_{c,\mathcal{S}} = 2a^\top b. \tag{6}$$

The coordinate form thus splits the same norm, bounds the local interference $N_{\mathcal{S}}$ of (2) from above (it also counts the conflict that a pair's agreement on other coordinates cancels, and equals $N_{\mathcal{S}}$ exactly when no pair has both an agreeing and a conflicting coordinate), and needs only the two sign-sum vectors: $O(B|\mathcal{S}|)$ operations instead of $O(B^2|\mathcal{S}|)$ for the Gram matrix (Figure 3). The per-sample gradients and the differentiation through them remain, and unlike N, N_c depends on the coordinate basis. MSQ (Roulet & Agarwala, 2025) uses signed cross-sample products, after positive and negative contributions cancel, to estimate a preconditioning statistic. GCAM penalizes negative coordinate products before cancellation (Appendix B).

Objective. GCAM normalizes $N_{c,\mathcal{S}}$ by the number of ordered pairs and adds it to the loss,

$$R_{\mathcal{S}} = \frac{2a^\top b}{B(B-1)}, \qquad L_{\mathrm{GCAM}} = L + \lambda R_{\mathcal{S}}, \quad \lambda \geq 0. \tag{7}$$

By (2) and (6), $\tau R_{\mathcal{S}}$ bounds from above, to first order in the step size τ, the average positive loss increase that a step along one sample's gradient on $\mathcal{S}$ causes in another; Appendix A bounds the

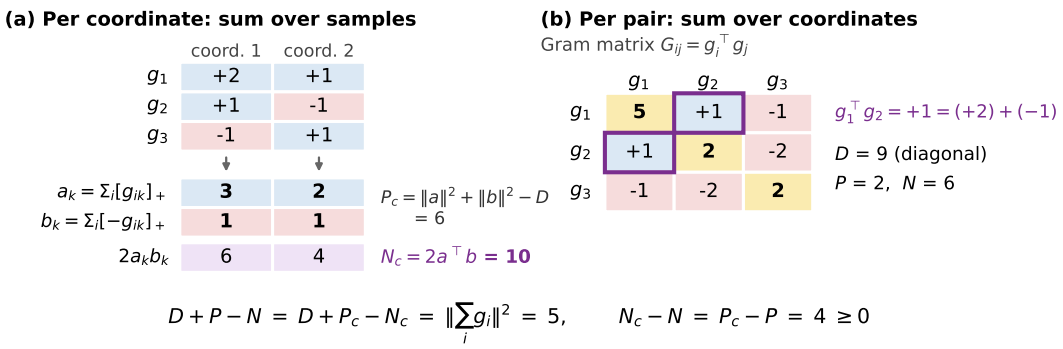


Figure 3: Proposition 1 on three samples with two coordinates, $g_1 = (2, 1)$, $g_2 = (1, -1)$, $g_3 = (-1, 1)$. (a) For each coordinate, the positive parts and the negative magnitudes of the gradients are summed over samples; this gives N_c and P_c without forming any pair. (b) For each pair, the inner product sums over coordinates (Gram matrix), giving the dot forms P and N. Both give the same norm (bottom line). Pairs with both an agreeing and a conflicting coordinate (boxed) count more in N_c than in N; here $N_c - N = P_c - P = 4$.

second-order remainder. Over all parameters with batch 256, penalties on D, P_c, N_c or the whole norm each rescue the eight of 14 seeds on which AdamW does not generalize within 20k updates, while a penalty on $P_c - N_c$ rescues none (Table 3, Appendix C; each term divided by B^2); we penalize N_c because it bounds the interference N from above. We write GCAM_{all} when $\mathcal{S}$ contains all parameters and GCAM_{VO} when it contains the value and output projection weights of the attention layers (V,O); Section 6 explains this choice. Each step forms per-sample gradients, computes the sign sums and $R_{\mathcal{S}}$, and passes $\bar{g} + \lambda \nabla_\theta R_{\mathcal{S}}$ to AdamW, differentiating with respect to all parameters (Algorithm 1). The penalty couples sample gradients to other samples' curvature through masked Hessian–gradient products; Appendix A derives this correction and its local relation to interference.

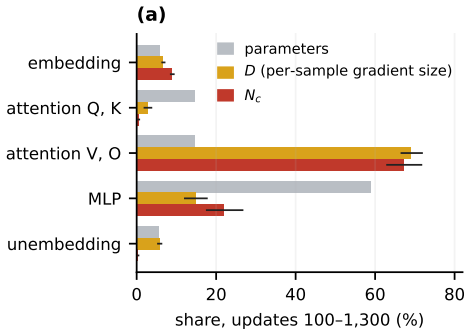
6 Where to apply the penalty

Module-wise interference. Because N_c is a sum over coordinates, it splits exactly across groups of parameters, which the dot form N does not. We track the split under AdamW in the reference setting of Section 3 with batch 256 over fourteen seeds (Appendix B), eight of which AdamW does not bring to 99% test accuracy within 20k updates. Over updates 100–1,300, before any of the fourteen seeds fits its training set under AdamW, the attention value and output projections (V,O) hold 63–72% of N_c (range over the seeds) with 14.7% of the parameters (Figure 4a). The squared per-sample gradient norms D are similarly concentrated, so this localization is not unique to conflict.

Placement. Regularizing V,O gradient coordinates yields earlier generalization than regularizing all parameters under both batching conditions (Figure 4b). A VO step also costs less, although it remains 3.7 times as slow as AdamW in our implementation (Table 5); Appendix C reports per-seed comparisons and full costs. We use VO for modular arithmetic. On CIFAR-10, the tested VO configuration underperforms AdamW, motivating all-parameter and classifier-only comparisons (Section 7).

7 Results

Comparison. Each rival looks past a different part of the averaged gradient: its norm (EGR; SAM, Foret et al., 2021), its signed per-coordinate interaction (MSQ, Roulet & Agarwala, 2025, Section 5; adaptation in Appendix B) or its history (Grokfast, $\lambda = 2$, $\alpha = .98$, Lee et al., 2024). All methods run on AdamW with weight decay 1. Table 1 varies the reference setting, using learning rate 10^{-3} except for the selected-rate 15% row. Each method uses its best tested coefficient; the rescue and timing figures use GCAM_{VO} with $\lambda = .02$.



(b)

Setting	coefficient all / VO	median t_{99} all / VO	ends $\geq 99\%$ all / VO
Batch 256	.005 / .05	3,650 / **3,050**	6 / 5
Full batch	.01 / .1	3,250 / **2,700**	5 / 6

Figure 4: Module-wise interference and GCAM placement. (a) Share of the parameter count, of the per-sample gradient magnitude D and of N_c for each parameter group, averaged over updates 100–1,300, before any seed fits its training set (AdamW, reference setting, batch 256; measured on a fixed set of 256 training samples; bar: median over 14 seeds, line: range; layer norms, below 1%, omitted). (b) GCAM_{all} against GCAM_{VO}, seeds 42–47: coefficient (earliest median t_{99} among those tested), median t_{99} in updates, and number of the six seeds that end at $\geq 99\%$ test accuracy. GCAM_{VO} reaches 99% earlier on 10 of the 12 seed pairs and at the same update on the other two.

Table 1: Methods across settings. Modular addition, one-layer Transformer, AdamW learning rate 10^{-3} (except *), weight decay 1, seeds 42–47; each row changes one item of the reference setting (width 128, 30% training data, batch 256), the last row two. Each cell: median t_{99} in thousands of updates (lower is better; – if a seed does not reach 99%; bold: the earliest in the row), then in parentheses the seeds reaching 99% / the seeds with test accuracy $\geq$99% at the last evaluation (higher is better). Budgets: 20k updates (40k with 20%, 120k with 15% data). Each method at its best tested coefficient (earliest median among those run on all six seeds; values in Appendix B). *With 15% of the data (1,411 training pairs) the best learning rate differs strongly between methods (at 10^{-3} AdamW reaches 99% on no seed), so this row gives each method its best of 10^{-3}, $2 \cdot 10^{-3}$ and $3 \cdot 10^{-3}$ (earliest median among configurations ending $\geq$99% on at least five seeds): AdamW and SAM $3 \cdot 10^{-3}$, GCAM_{VO} ($\lambda = .1$) and Grokfast $2 \cdot 10^{-3}$, MSQ 10^{-3}; EGR was run at 10^{-3} only.

Setting	AdamW	EGR	GCAM_{VO}	SAM	MSQ	Grokfast
Full batch	4.9 (6/6)	3.95 (6/5)	**2.7** (6/6)	6.1 (6/6)	4.45 (6/6)	5.65 (6/4)
Reference	– (5/5)	3.5 (6/6)	3.05 (6/5)	3.65 (6/6)	**2.7** (6/6)	4.25 (6/6)
Width 64	– (5/2)	– (4/3)	**2.85** (6/6)	3.95 (6/6)	3.9 (6/5)	6.2 (6/6)
Width 48	– (3/1)	– (2/1)	**3.1** (6/6)	– (3/3)	4.65 (6/6)	6.45 (6/6)
20% training data	– (5/5)	7.7 (6/6)	**6.75** (6/5)	10.4 (6/6)	7.3 (6/3)	10.15 (6/6)
15% training data*	10.85 (6/5)	45.4 (6/6)	**6.5** (6/5)	8.1 (6/5)	13.35 (6/5)	12.05 (6/6)
Width 48, 20% data	– (1/0)	– (0/0)	**6.9** (6/6)	– (3/3)	13.6 (6/5)	– (5/5)

7.1 Reference setting: earlier generalization and rescued seeds

With full-batch training, GCAM_{VO} reaches 99% test accuracy earlier than AdamW on five seeds and at the same update on the sixth. With batch 256, it advances generalization on all six successful AdamW seeds and enables all eight baseline-failing seeds in the fourteen-seed cohort to reach 99% within 20k updates (Figure 5). These paired runs fit the training data earlier and show markedly smaller absolute N_c and Q; their joint decrease alone does not identify selective conflict reduction. All eight end above 99%, although two undergo severe intervening dips.

7.2 Harder settings: rescue

Width 48. GCAM_{VO} reaches 99% on all six seeds, whereas AdamW and SAM do so on three and EGR on at most two (Table 1). With $\lambda = .02$, GCAM nearly eliminates the memorization–generalization delay: the median gap is 100 updates and no seed exceeds 200, compared with 1,100 for MSQ and 4,600 for Grokfast (Figure 6). The gap also shrinks at width 64 and at width 48 with

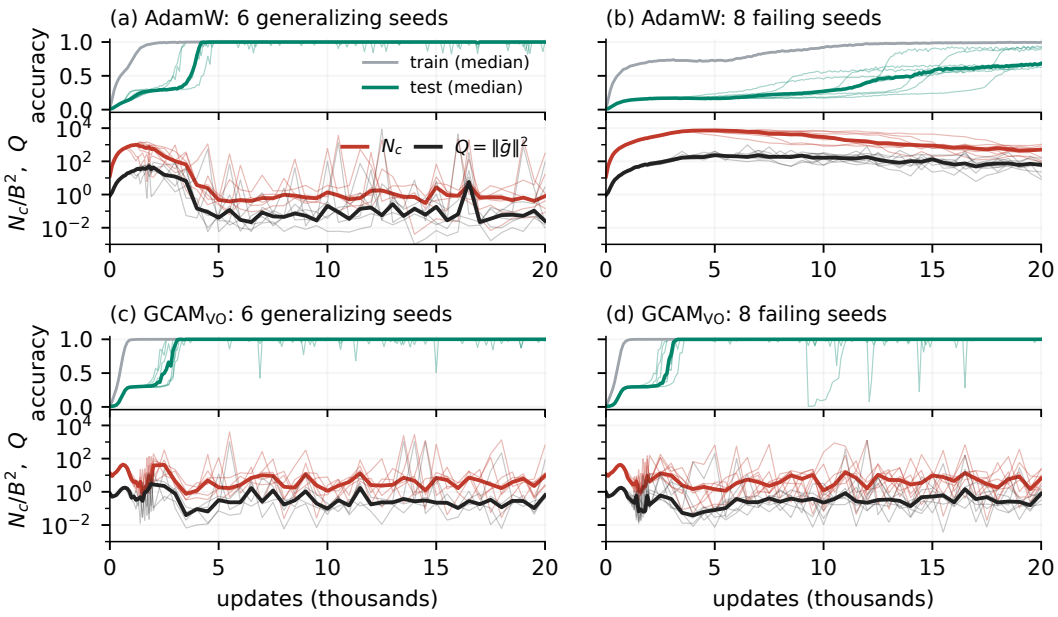


Figure 5: The fourteen seeds of the batch-256 set, grouped by whether AdamW generalizes, without (top) and with GCAM$_{VO}$ ($\lambda = .02$, bottom). Upper panels: test accuracy of each seed (thin) and median training and test accuracy (thick). Lower panels: N_c per B^2 (red) and the squared batch-gradient norm Q (black) over all parameters, each seed thin and the median thick, on 256 fixed training samples (Q for 11 of the 14 GCAM$_{VO}$ runs, whose D was logged). Under GCAM$_{VO}$ all eight failing seeds reach 99% and end above it; two (1 and 43) dip to 7.7% and 0.9% on the way.

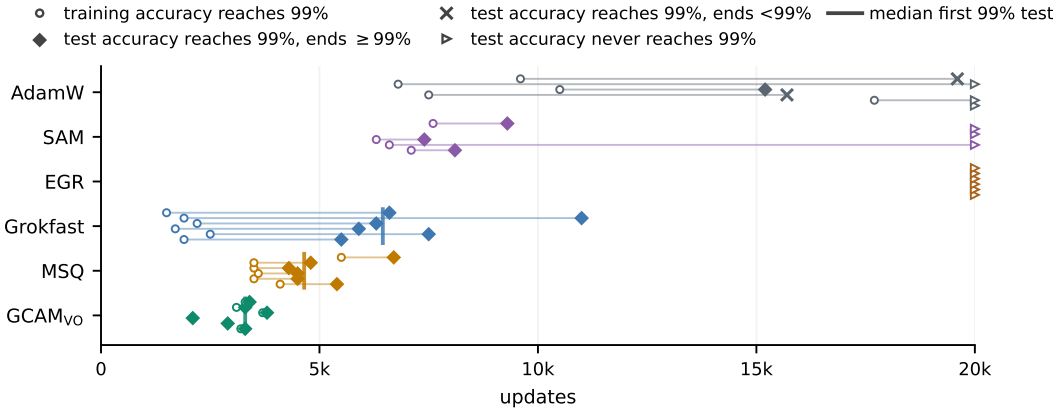


Figure 6: Width 48 (batch 256, 30% data, learning rate 10^{-3}, seeds 42–47), one row per seed as in Figure 2; no circle: training accuracy does not reach 99% within the budget. GCAM$_{VO}$ $\lambda = .02$, SAM $\rho = .05$, Grokfast $\lambda = 2$, EGR $\alpha = .003$; retention after t_{99}: Appendix Table 7.

20% training data, but not at width 128. Table 1 reports best-tested coefficients, which differ from this fixed-coefficient illustration.

Less training data. The 15% condition is sensitive to learning rate: AdamW reaches 99% on no seed at 10^{-3} but on all six at its selected higher rate. Under the reported selection rule, GCAM has the earliest median t_{99} while ending at or above 99% on five of six seeds (Table 1). Appendix Table 8 reports rate sensitivity, and Table 10 extends the task to multiplication.

Table 2: CIFAR-10, vision Transformer (three layers, width 128), batch 128, AdamW with weight decay .01 and constant learning rate, 100 epochs. Mean ± standard deviation over seeds 41–43 of the final evaluation on 2,000 test images, in percent; each method at the better of learning rates 10^{-3} and $3 \cdot 10^{-4}$ (LR). Coefficient: EGR α, SAM ρ, Grokfast λ, GCAM λ on N_c/B^2 over all parameters or over the classifier weights; 0: none. GCAM: best tested coefficient (†: .1; tested .01–1 on all parameters, .1–2 on the classifier, which was run at 10^{-3} only); each rival: one coefficient. MSQ* is our adaptation (Appendix B). Bold: best in the column.

		No augmentation		Flip + crop	
Method	Coefficient	Accuracy	LR	Accuracy	LR
AdamW	0	66.60 ± 0.60	$3 \cdot 10^{-4}$	78.12 ± 0.74	10^{-3}
EGR	0.02	67.93 ± 1.10	$3 \cdot 10^{-4}$	78.78 ± 0.38	$3 \cdot 10^{-4}$
SAM	0.05	67.68 ± 1.44	$3 \cdot 10^{-4}$	78.60 ± 0.40	$3 \cdot 10^{-4}$
Grokfast	2	68.35 ± 0.68	$3 \cdot 10^{-4}$	79.32 ± 0.91	$3 \cdot 10^{-4}$
MSQ*	0	69.88 ± 0.84	$3 \cdot 10^{-4}$	**80.53** ± 0.81	$3 \cdot 10^{-4}$
GCAM, all	0.05	**72.00** $\pm 0.13^\dagger$	10^{-3}	78.92 ± 0.68	10^{-3}
GCAM, classifier	0.5	66.33 ± 2.01	10^{-3}	79.05 ± 0.48	10^{-3}

7.3 Image classification

We evaluate the penalty beyond grokking using a vision Transformer (Dosovitskiy et al., 2021) on CIFAR-10 (Krizhevsky, 2009). Without augmentation, GCAM_{all} achieves the highest mean accuracy among the evaluated configurations, including when rivals use their better tested learning rate (Table 2). With augmentation, MSQ leads and the GCAM gain is smaller. Appendix E reports the protocol. Coefficient-search budgets differ across methods, as disclosed in the table.

7.4 Gradient and trajectory geometry

At width 48, GCAM_{VO} has a higher gradient participation ratio at update 1,000 than AdamW (181–188 versus 61–67; six seeds). CIFAR-10 shows a different organization: within-class gradients align more, between-class mean cosines become more negative, and participation ratio falls. Appendix E reports full-gradient statistics at $\lambda = .1$ with the classifier excluded; Figure 8 illustrates class structure at $\lambda = .2$. These task-dependent changes complement the Hessian–gradient interpretation but do not themselves establish a decrease in dot-product conflict N.

A separate matched-seed width-48 cohort illustrates the trajectories on common train and test loss slices (Figure 7; GCAM_{all}, $\lambda = .005$). GCAM reaches 99% test accuracy first in this cohort. The actual learning curves distinguish this timing from the loss of models reconstructed on the visualization plane; projected locations alone do not establish the loss of the trained models or distinct attraction basins.

8 Discussion

The squared batch-gradient norm $Q = (D + P - N)/B^2$ couples gradient magnitude, agreement and conflict, so a change in Q does not determine a change in N. GCAM penalizes the conflicting part selectively, while training still moves all three (Appendix A: the cross term between the two has no fixed sign). Its effects appear in timing, where the delay between fitting and generalizing nearly vanishes at narrow width, and in geometry, where CIFAR-10 gradients organize by class; neither shows by itself that conflict caused the original delay. How the task gradient, the penalty and AdamW's preconditioning jointly change absolute and normalized conflict and the timing of the transition remains to be measured, for instance by intervening at matched checkpoints. *Limits.* A GCAM_{VO} step costs about 3.7 times the time and 5 times the memory of AdamW's (Table 5); its gains depend on the learning rate and coefficient, and in wider models its test accuracy dips after first reaching 99%, far more than SAM's.

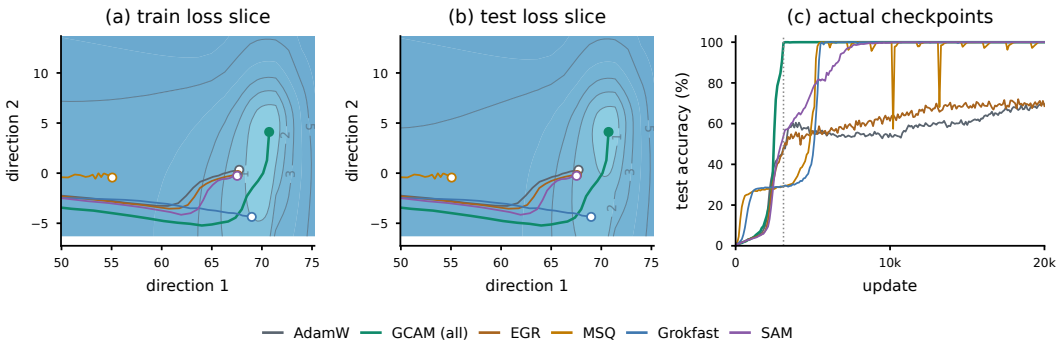


Figure 7: Shared loss slices and training paths: width 48, batch 256, 30% data, seed 42, GCAM_{all} $\lambda = .005$. (a,b) Train/test cross-entropy on one fixed plane through GCAM's initialization, memorization and generalization checkpoints. All six paths use the same projection and are shown through step 3,100; filled markers are on-plane checkpoints, hollow markers are projections. Contours give plane-model loss, not necessarily actual checkpoint loss. (c) Actual test accuracy over the full budget; dotted line marks 3,100. Appendix E.1 gives the protocol.

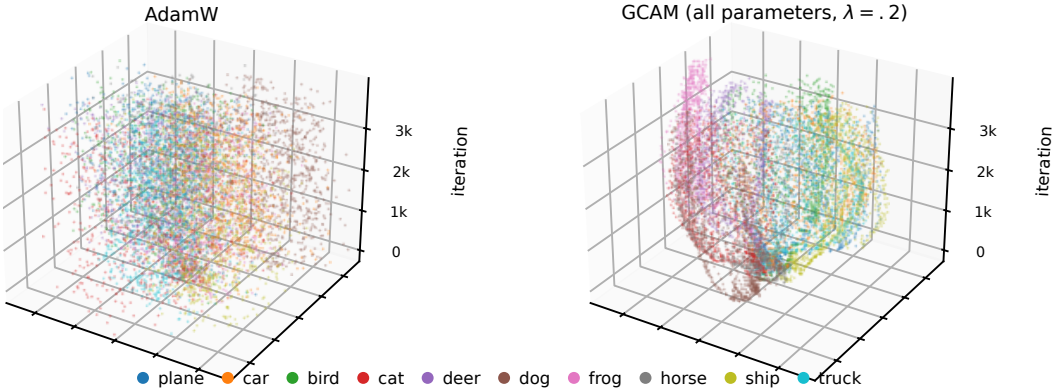


Figure 8: CIFAR-10 gradient class structure (200 training images, 20 per class; seed 43, no augmentation, all parameters). Fixed 512-dimensional gradient sketches are unit-normalized and projected onto two uncentered singular-vector directions fitted separately for each run; height gives updates, sampled every 100 through 3,900. Panels illustrate within-run structure, not shared coordinates. GCAM uses $\lambda = .2$; Table 2 uses the stated accuracy-selected coefficients.

AI USE STATEMENT

The research direction, the hypotheses and arguments, the GCAM method and its original implementation, the structure of the manuscript and its original drafts, and the references on which the analysis builds are the authors'. Within that scope we used generative AI assistants (Claude, Anthropic; Codex, OpenAI) in five ways. (i) Execution: implementing baselines and additional configurations on the authors' code, writing analysis and plotting code, and running and managing experiments; the authors designed the experimental framework and corrected and redirected the assistants' implementations and analyses where needed. (ii) Writing: revising and expanding the authors' drafts, with content and structure set by the authors. (iii) Retrieval: locating related literature. (iv) Proofs: for the derivations in Appendix A, the authors set the approach and supplied the key references, and the assistants helped derive, write out and check them. (v) Analysis: computing the statistics and checks the authors specified; the interpretation is the authors'. The authors reviewed every sentence and every derivation step, and checked AI-run experiments against their configurations and logs. We take full responsibility for all content of this paper.

REFERENCES

Atish Agarwala and Yann N. Dauphin. SAM operates far from home: eigenvalue regularization as a dynamical phenomenon. In *International Conference on Machine Learning*, 2023.

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. In *Advances in Neural Information Processing Systems*, 2023.

Imen Ayadi and Gabriel Turinici. Stochastic Runge–Kutta methods and adaptive SGD-G2 stochastic gradient descent. *arXiv preprint arXiv:2002.09304*, 2020. URL https://arxiv.org/abs/2002.09304.

David G. T. Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.

Etienne Boursier, Scott Pesme, and Radu-Alexandru Dragomir. A theoretical framework for grokking: Interpolation followed by riemannian norm minimisation. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.20172.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2002.10657.

Satrajit Chatterjee and Piotr Zielinski. On the generalization mystery in deep learning. *arXiv preprint arXiv:2203.10036*, 2022. URL https://arxiv.org/abs/2203.10036.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *Advances in Neural Information Processing Systems*, 2020.

Benoit Dherin. Implicit biases in multitask and continual learning from a backward error analysis perspective. *arXiv preprint arXiv:2311.00235*, 2023.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, Sourabh Medapati, and Javier Gonzalvo. Learning by solving differential equations. *arXiv preprint arXiv:2505.13397*, 2025. URL https://arxiv.org/abs/2505.13397.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.11929.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Stanislav Fort, Pawel Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer, 2nd edition, 2006.

Ryo Karakida, Tomoumi Takase, Tomohiro Hayase, and Kazuki Osawa. Understanding gradient regularization in deep learning: Efficient finite-difference computation and implicit bias. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 15809–15827. PMLR, 2023. URL https://proceedings.mlr.press/v202/karakida23a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06110v3.

Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2024.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1712.09913.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.10343.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.01117v2.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.08840.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Andreas Maurer. A note on the pac bayesian theorem. *arXiv preprint arXiv:cs/0411099*, 2004.

Xuran Meng, Yuan Cao, and Difan Zou. Per-example gradient regularization improves learning signals from noisy data. *arXiv preprint arXiv:2303.17940*, 2023.

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.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. In *International Conference on Learning Representations*, 2021.

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.

Lucas Prieto, Melih Barsbey, Pedro A. M. Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. In *International Conference on Learning Representations*, 2025.

Elan Rosenfeld and Andrej Risteski. Outliers with opposing signals have an outsized effect on neural network optimization. In *International Conference on Learning Representations*, 2024.

Vincent Roulet and Atish Agarwala. Per-example gradients: a new frontier for understanding and improving optimizers. *arXiv preprint arXiv:2510.00236*, 2025.

Ali Saheb Pasand and Elvis Dohmatob. Egalitarian gradient descent: A simple approach to accelerated grokking. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.04930v3.

Karthik Abinav Sankararaman, Soham De, Zheng Xu, W. Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 8469–8479. PMLR, 2020. URL https://proceedings.mlr.press/v119/sankararaman20a.html.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2101.12176.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022. URL https://arxiv.org/abs/2206.04817.

Thomas Walker, Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. GrokAlign: Geometric characterisation and acceleration of grokking. *arXiv preprint arXiv:2506.12284*, 2025. URL https://arxiv.org/abs/2506.12284.

Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: A key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, 2018. URL https://arxiv.org/abs/1706.05699v3.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

Zhenshuo Zhang, Jerry W. Liu, Christopher Ré, and Hongyang R. Zhang. A hessian view of grokking in mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS*, 2024. URL https://openreview.net/forum?id=BfLVCoov0b.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, volume 162 of *PMLR*, 2022.

Xinyu Zhou, Simin Fan, Martin Jaggi, and Jie Fu. NeuralGrok: Accelerate grokking by neural gradient transformation. *arXiv preprint arXiv:2504.17243*, 2025. URL https://arxiv.org/abs/2504.17243.

A Derivations and mathematical scope

Proposition 1. *For any $h_1, \dots, h_B$, with $A_{ij} = \sum_{k\in\mathcal{S}} [h_{ik}h_{jk}]_+$ and $C_{ij} = \sum_{k\in\mathcal{S}} [-h_{ik}h_{jk}]_+$:*

$$\text{(i)} \quad B^2\|\bar{h}\|^2 = D_{\mathcal{S}} + P_{\mathcal{S}} - N_{\mathcal{S}} = D_{\mathcal{S}} + P_{c,\mathcal{S}} - N_{c,\mathcal{S}}; \tag{8}$$

$$\text{(ii)} \quad N_{c,\mathcal{S}} - N_{\mathcal{S}} = P_{c,\mathcal{S}} - P_{\mathcal{S}} = \sum_{i\neq j} \min(A_{ij}, C_{ij}) \ \geq 0; \tag{9}$$

$$\text{(iii)} \quad N_{c,\mathcal{S}} = 2a^\top b, \qquad P_{c,\mathcal{S}} = \|a\|^2 + \|b\|^2 - D_{\mathcal{S}}. \tag{10}$$

The coordinate forms, sign sums and dot forms are defined in Section 5; Figure 3 gives an example.

Proof of Proposition 1. (i) $B^2\|\bar{h}\|^2 = \|\sum_i h_i\|^2 = \sum_i \|h_i\|^2 + \sum_{i\neq j} h_i^\top h_j$. Writing every scalar as $x = [x]_+ - [-x]_+$, applied to each inner product $h_i^\top h_j$ gives the dot forms, and applied to each coordinate product $h_{ik}h_{jk}$ before summing over k gives the coordinate forms; both split the same off-diagonal sum. (ii) For a pair, $h_i^\top h_j = \sum_k h_{ik}h_{jk} = A_{ij} - C_{ij}$ with $A_{ij}, C_{ij} \geq 0$. For nonnegative A, C, $[C - A]_+ = C - \min(A, C)$ and $[A - C]_+ = A - \min(A, C)$. Summing over ordered pairs gives $N_{\mathcal{S}} = N_{c,\mathcal{S}} - \sum_{i\neq j} \min(A_{ij}, C_{ij})$ and the same for P. (iii) $a_k b_k = \sum_i \sum_j [h_{ik}]_+ [-h_{jk}]_+$. Terms with $i = j$ vanish, since $[h_{ik}]_+ [-h_{ik}]_+ = 0$. For $i \neq j$, $[-h_{ik}h_{jk}]_+ = [h_{ik}]_+ [-h_{jk}]_+ + [-h_{ik}]_+ [h_{jk}]_+$, and summing over ordered pairs counts each of the two products once in $a_k b_k$ and once in its reversed order, so $N_{c,\mathcal{S}} = 2\sum_k a_k b_k$. Similarly $\|a\|^2 + \|b\|^2 = \sum_k \sum_{i,j} \big([h_{ik}]_+ [h_{jk}]_+ + [-h_{ik}]_+ [-h_{jk}]_+\big)$, whose $i = j$ terms sum to $D_{\mathcal{S}}$ and whose $i \neq j$ terms sum to $P_{c,\mathcal{S}}$.

Algorithm 1 One GCAM step.

Input: batch of $B > 1$ samples, parameters θ, coordinate set $\mathcal{S}$, coefficient λ.
1. Compute the per-sample gradients $g_i = \nabla_\theta \ell_i$ (`torch.func.vmap` over `grad`), keeping the graph for differentiation, and h_i (zero outside $\mathcal{S}$).
2. Form $a = \sum_i [h_i]_+$ and $b = \sum_i [-h_i]_+$ (elementwise) and $R_{\mathcal{S}} = 2a^\top b/[B(B-1)]$.
3. Compute $u = \bar{g} + \lambda \nabla_\theta R_{\mathcal{S}}$ with respect to all parameters, differentiating through step 1; through second derivatives this also changes parameters outside $\mathcal{S}$.
4. Pass u to the base optimizer (AdamW here) in place of $\bar{g}$.

Relation to the norm. By Proposition 1(i), the GCAM term of Equation 7 can be written as

$$R_{\mathcal{S}} = \frac{B}{B-1}\left[\frac{D_{\mathcal{S}} + P_{c,\mathcal{S}}}{B^2} - \|\bar{h}\|^2\right], \tag{11}$$

so D and P_c enter $R_{\mathcal{S}}$ with a positive sign, as they do in Q; GCAM is not the negative of a norm penalty.

Finite-step interference. Let $\mathcal{F}_\tau = [B(B-1)]^{-1} \sum_{i\neq j} [\ell_i(\theta - \tau h_j) - \ell_i(\theta)]_+$ be the average loss increase that a step of size τ on one sample causes another, and $R_{\mathcal{S}}^{\mathrm{dot}} = N_{\mathcal{S}}/[B(B-1)]$ its first-order value from Equation 2. If each per-sample Hessian has operator norm at most β along the steps, then

$$|\mathcal{F}_\tau - \tau R_{\mathcal{S}}^{\mathrm{dot}}| \leq \frac{\beta\tau^2 D_{\mathcal{S}}}{2B}, \qquad \mathcal{F}_\tau \leq \tau R_{\mathcal{S}} + \frac{\beta\tau^2 D_{\mathcal{S}}}{2B}. \tag{12}$$

Proof. Taylor's theorem gives remainder e_{ij} with $|e_{ij}| \leq \beta\tau^2\|h_j\|^2/2$. Because the projector is orthogonal, $g_i^\top h_j = h_i^\top h_j$. The positive-part function is 1-Lipschitz, hence

$$\left|[\ell_i(\theta - \tau h_j) - \ell_i(\theta)]_+ - \tau\left[-h_i^\top h_j\right]_+\right| \leq \tfrac{1}{2}\beta\tau^2\|h_j\|^2.$$

Summing over ordered pairs produces $(B-1)D_{\mathcal{S}}$ in the remainder and proves the first inequality of Equation 12; substituting $R_{\mathcal{S}}^{\mathrm{dot}} \leq R_{\mathcal{S}}$ (Proposition 1(ii)) yields the second. This is a local finite-step statement, not a population-risk theorem.

Sharpness and norm. Under the Hessian bound, every perturbation in the radius ρ ball satisfies $|L(\theta+\delta) - L(\theta) - \bar{g}^\top \delta| \le \beta_L \rho^2/2$. Maximizing the linear term gives $\rho\|\bar{g}\|$, which proves the first-order statement for SAM in Section 3, with the $O(\rho^2)$ term at most $\frac{1}{2}\beta_L\rho^2$, including $\bar{g} = 0$. A PAC-Bayes result also contains its distributional assumptions and complexity term (Maurer, 2004; Foret et al., 2021); substituting a coordinate-conflict upper bound does not remove them.

Derivative and sign convention. For nonzero selected h_{ik}, differentiating $a^\top b$ with respect to h_{ik} gives b_k for positive coordinates and $-a_k$ for negative ones. With $H_i = \nabla^2_\theta \ell_i$ and $w_{ik} = b_k$ on positive selected coordinates, $w_{ik} = -a_k$ on negative selected coordinates and zero elsewhere, multiplication by $2/[B(B-1)]$ and the chain rule give

$$\nabla_\theta R_{\mathcal{S}} = \frac{2}{B(B-1)} \sum_i H_i^\top w_i. \tag{13}$$

The mixed blocks of H_i carry this correction to parameters outside $\mathcal{S}$. At zero, the rectifier's chosen subgradient must be specified by the implementation; the displayed smooth derivative applies away from these boundaries.

Hessian–gradient form of the correction. For a pair with inner product $c_{ij} = g_i^\top g_j$ and per-sample Hessians H_i,

$$\nabla_\theta c_{ij} = H_i g_j + H_j g_i. \tag{14}$$

For a conflicting pair ($c_{ij} < 0$) the penalty $r_{ij} = [-c_{ij}]_+$ has gradient $-(H_i g_j + H_j g_i)$; its descent correction combines the sensitivity of each sample's gradient to the other sample's gradient direction ($H_i g_j$ is the derivative of g_i along g_j). Under the gradient flow of this pair term alone, $\dot\theta = -\lambda \nabla r_{ij}$, one has $\dot c_{ij} = \lambda \|H_i g_j + H_j g_i\|^2 \ge 0$: the negative inner product becomes less negative (not necessarily orthogonal, since the norms can shrink). GCAM penalizes coordinate products: on a region where the product signs are fixed, with M_{ij} the diagonal mask of the coordinates in $\mathcal{S}$ where $h_{ik}h_{jk} < 0$, the pair term $-g_i^\top M_{ij} g_j$ has gradient $-(H_i M_{ij} g_j + H_j M_{ij} g_i)$, and summing over ordered pairs gives (13) with $w_i = -\sum_{j\neq i} M_{ij} h_j$. The pairwise monotonicity does not survive the sum over pairs, the task loss, AdamW's preconditioning or the coordinate bound; for the penalty alone only $\dot R_{\mathcal{S}} = -\lambda \|\nabla R_{\mathcal{S}}\|^2$ holds. For a small plain gradient step on the penalty alone, $\delta\theta = -\eta\lambda \nabla R_{\mathcal{S}}$,

$$\Delta N_{\mathcal{S}} = -\eta\lambda \langle \nabla N_{\mathcal{S}}, \nabla R_{\mathcal{S}} \rangle + O(\|\delta\theta\|^2) = -\frac{\eta\lambda}{B(B-1)} \langle \nabla N_{\mathcal{S}}, \nabla N_{c,\mathcal{S}} \rangle + O(\|\delta\theta\|^2), \tag{15}$$

negative when the two gradients are positively aligned. Under plain gradient flow of $L + \lambda R_{\mathcal{S}}$, likewise, $\dot Q = -\langle \nabla Q, \nabla L \rangle - \lambda \langle \nabla Q, \nabla R_{\mathcal{S}} \rangle$ with a cross term of no fixed sign, so penalizing N_c and reducing the norm are connected but not interchangeable.

Near a common stationary point θ^*, with $\delta = \theta - \theta^*$ and Lipschitz Hessians, $g_i = H_i^* \delta + O(\|\delta\|^2)$ and

$$c_{ij} = \delta^\top H_i^* H_j^* \delta + O(\|\delta\|^3), \tag{16}$$

so the interaction of two samples measures how their curvature responses to the same displacement overlap: with orthonormal bases U_i, U_j of the nonzero eigenspaces, $|c_{ij}| \le \|\delta\|^2 \|H_i^*\| \|H_j^*\| \|U_i^\top U_j\| + O(\|\delta\|^3)$, small when those subspaces are nearly orthogonal (Sankararaman et al., 2020, App. C, a heuristic under their assumptions). No conclusion about Hessian rank follows. The same terms appear, with a minus sign between them, in the difference between the two orders of two sample steps, $\eta^2(H_j g_i - H_i g_j) + O(\eta^3)$ (Dherin, 2023); the penalty acts on their sum, not on that difference.

Integrator probes. Heun uses $k_1 = \nabla L(\theta)$, $k_2 = \nabla L(\theta - \eta k_1)$ and returns $(k_1 + k_2)/2$. RK4 uses the classical stages at θ, $\theta - \eta k_1/2$, $\theta - \eta k_2/2$, and $\theta - \eta k_3$, returning $(k_1 + 2k_2 + 2k_3 + k_4)/6$. The same batch is used throughout. Both have leading expansion $g_{\text{probe}} = \bar g - \frac{\eta}{2} H \bar g + O(\eta^2)$; EGR instead supplies $\bar g + 2\alpha H \bar g$. The subsequent AdamW step applies momentum, scaling, and decay once. These probes are not Runge–Kutta integration of the optimizer state (Ayadi & Turinici, 2020; Dherin et al., 2025). Random-shuffling SGD has a separate batch-dispersion correction (Smith et al., 2021); its sampling assumptions do not establish an exact AdamW implicit term in our protocol.

B PROTOCOL, IMPLEMENTATION, AND REPORTING

The arithmetic model is a pre-LayerNorm Transformer with learned positions, four attention heads, and a GELU MLP of width $4d$, trained in fp32. Split permutation seed 1234 gives 2,822 training and 6,587 held-out pairs at fraction .30. Initialization seeds are retained in matched method comparisons. Batch 0 in saved configurations denotes full batch; measurements of gradient interactions use the stated fixed training probe of $B_p = 256$ samples, distinct from the optimization batch. Training-only coordinate quantities use 256 probe samples; sparse energy/norm records are shown only where saved rather than interpolated.

EMA-Grokfast uses $(\alpha, \lambda) = (.98, 2)$, initialized at zero; SAM uses the same batch at the perturbation and gradient evaluations, with the radius given with each table. GCAM practical tables use value/output projection coordinates and no cap. Weight decay is decoupled, not an additional L_2 loss term. For the EGD* comparison, saved configuration specifies AdamW as its base, all two-dimensional parameter tensors, and no randomized-SVD rank truncation. This identifies the tested variant; it is not a claim of an identical protocol to every experiment in the EGD paper.

MSQ and the sign sums. Per coordinate, $B^2\bar{h}_k^2 = (a_k - b_k)^2$ and $\sum_i h_{ik}^2 = D_k$, so before clipping and temporal averaging the unbiased squared-mean estimate that MSQ uses in place of Adam's second moment is $\widehat{\mu}_k^2 = (P_{c,k} - N_{c,k})/(B(B-1))$, with $P_{c,k}$ and $N_{c,k}$ the terms of coordinate k in (5). MSQ uses this statistic to precondition; GCAM differentiates a penalty on $N_{c,k}$.

Adapted MSQ. The mean task gradient is clipped to norm 1 before updating its first moment and mean-gradient second moment v_A. The per-example moment v_M uses unclipped sample gradients. Bias-corrected moments give $\widehat{\mu}^2 = (v_A - v_M/B)/(1 - 1/B)$; the normalized update $m/(\sqrt{[\widehat{\mu}^2]_+} + 10^{-6})$ is clamped coordinate-wise to $[-3, 3]$ before LR scaling, with decoupled decay. Clipping can make the two streams inconsistently scaled. This comparison is the implemented adaptation, not a complete reproduction of every reference result (Roulet & Agarwala, 2025).

Outcomes and budgets. A seed reaches 99% (at time t_{99}) at its first evaluation with test accuracy $\geq .99$, ends at 99% if its final evaluation is $\geq .99$, and stays at 99% if every evaluation from t_{99} to the budget is $\geq .99$; it cannot detect excursions between evaluations. Seeds that never reach 99% remain in every denominator. A median t_{99} is given only if every seed reaches 99% (otherwise –); conditional medians in supporting tables are marked. Standard cohorts use 20k updates, fraction .20 uses 40k, and fraction .15 uses 120k; the addition–multiplication task uses 120k. Coefficients in Table 1, for its rows in order: GCAM$_{\mathrm{VO}}$ $\lambda = .1/.05/.05/.05/.05/.1/.1$; SAM $\rho = .05/.2/.2/.05/.05/.05/.05$; EGR $\alpha = .00025/.2/.0003/.0003/.2/.2/.001$; Grokfast $\lambda = 2$.

Seeds. The fourteen batch-256 seeds of Section 6 (1, 2, 42–47, 51–53, 61, 64, 68) were chosen from AdamW outcomes and from which baseline runs already existed. The eight on which AdamW does not reach 99% (1, 2, 43, 51, 52, 61, 64, 68) are every such seed among the 39 previously run in the reference setting (0–2, 28, 39, 42–75); the six on which it does (42, 44–47, 53) are six of the seven generalizing seeds that already had Heun and EGR runs. Failures are therefore over-represented (8 of 14, against 920 of 5,483 seeds in the population of Appendix F), so rescue counts on this set are counts, not rates.

C COMPONENT PENALTIES, PLACEMENT AND COST

Component penalties. Before selecting a module, we intervene over all parameters:

$$L_C = L + \lambda C/B^2, \qquad C \in \{D + P_c - N_c,\ D,\ P_c,\ N_c,\ P_c - N_c\}. \tag{17}$$

Both gradient factors are differentiated. These are explicit component interventions, not deletion of a known implicit AdamW term. In the 14-seed batch-256 set, AdamW reaches 99% on 6 seeds; penalties on the whole norm, on D, on P_c and on N_c each rescue all 8 others, and the penalty on the signed interaction $P_c - N_c$ rescues none at this coefficient. Their final accuracies differ even when all runs reach 99%. Common-coefficient B^2-normalized penalties are separated from the

Table 3: All-parameter component interventions. Reach/End: seeds reaching 99% at any evaluation / at the final evaluation; Rescued: of the eight seeds on which AdamW does not reach 99%. Full batch: three seeds, with the indicated illustrative coefficients. Batch 256: 14 seeds, coefficient .02 and B^2 normalization. Full-batch N_c uses $B(B-1)$ normalization. The full-batch doses are not equal-strength interventions.

	Full batch, seeds 42–44			Batch 256, $\lambda = .02$, 14 seeds	
Penalty	λ	t_{99}	Reach/End	Reach/End	Rescued
None	0	5,600	3/3	6/6	0/8
$D + P_c - N_c$	.00025	4,200	3/3	14/14	8/8
D	.00025	5,100	3/3	14/9	8/8
P_c	.002	3,600	3/3	14/14	8/8
N_c	.01	3,500	3/2	14/13	8/8
$P_c - N_c$	.001	4,700	3/2	6/5	0/8

illustrative full-batch doses; neither a common coefficient nor an equal scalar penalty guarantees an equal delivered optimizer correction.

The success of the positive-component and whole-norm controls shows that rescue is not unique to the negative-interaction penalty. The interpretation of GCAM therefore rests on its selective objective, rather than on failure of these controls.

Placement details. Figure 9 shows how the shares of Figure 4a change over training. Table 4 lists the per-seed times behind Figure 4b and two further comparisons: on six further seeds (48–53) with the full-batch coefficients fixed, GCAM_{VO} is earlier on all six (median 2,500 against 5,050 updates; GCAM_{all} never reaches 99% on one), and with 20% training data it is earlier on all six (6,750 against 9,250). After first reaching 99%, test accuracy with full-batch training falls below 99% on every seed for both placements: GCAM_{VO} at 1–77% of the later evaluations (seed 44 down to 0.85%) and GCAM_{all} at 4–37% (lowest 0.88%). At equal coefficients (full batch, seeds 42–44), GCAM_{all} reaches 99% on one of three seeds at both $\lambda = .02$ and $.05$, and GCAM_{VO} on all three, ending below 99% on one and two of them.

Table 4: Per-seed t_{99} for the placement comparison of Figure 4b (coefficients as there). –: never reaches 99%; †: test accuracy below 99% at the last evaluation.

Setting (seeds)	t_{99}, GCAM_{all} / GCAM_{VO}, per seed					
Full batch (42–47)	3,300 / 2,800	3,500 / 3,000	4,000† / 2,100	3,200 / 3,200	3,000 / 2,200	2,900 / 2,600
Full batch (48–53)	4,900 / 1,800	–† / 2,500	5,800 / 2,500	4,900 / 2,800	5,200† / 2,100	3,200 / 2,600
Batch 256 (42–47)	2,900 / 2,700	4,000 / 2,900	3,400 / 2,800	4,000 / 3,300	3,400 / 3,400	3,900 / 3,200†
20% data (42–47)	10,500 / 7,600†	9,900 / 5,400	8,700 / 6,000	9,800 / 7,000	8,700 / 6,800	7,800 / 6,700

Cost per step. Table 5 gives the cost of one training step for each method in the reference model (modular addition, one layer, width 128, four heads, MLP width 512, batch 256). FLOPs count matrix multiplications only (`torch.utils.flop_counter`): the forward pass and every backward pass, including the second-order pass through a gradient penalty; elementwise operations and the optimizer update are not counted. Time and peak memory are measured with each method alone in its own process on an otherwise idle RTX A6000 (median of $2 \times 3 \times 50$ steps from initialization, optimizer update included). At this model size a step's time is set mostly by kernel launches, and repeated measurements of the same method differ by up to 20%. SAM takes two forward and backward passes, EGR differentiates through the gradient norm, Grokfast adds only elementwise filtering, and MSQ* sums the squared per-sample gradients inside the backward pass without storing them, as Roulet & Agarwala (2025) do (exact here; each sample is a three-token sequence, so the sums cover pairs of positions); our training runs formed them from stored per-sample gradients instead, which gives the same update at 11.3 ms and 305 MiB. GCAM adds per-sample gradients and a backward pass through the penalty; GCAM_{VO} forms them for V,O only, which gives the same penalty and gradient as forming them for all parameters (checked on one batch in double precision),

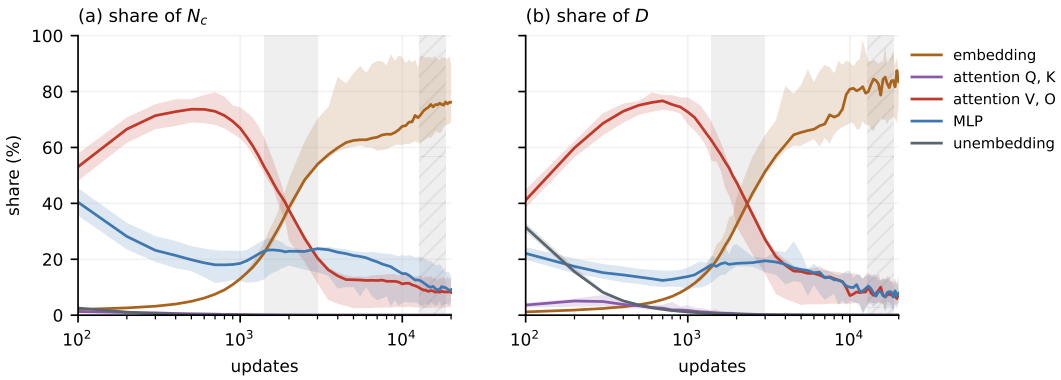


Figure 9: Share of N_c (a) and of D (b) held by each parameter group at every probed update (AdamW, batch 256, reference setting; line: median over 14 seeds, band: range). Grey band: t_{mem} of the six seeds that AdamW generalizes (1,400–3,000); hatched band: t_{mem} of the eight it does not (12,800–18,500).

the path our runs used. GCAM needs the sign of each per-sample coordinate, so unlike MSQ it has to form the per-sample gradients of the value weights, whose gradient sums over input positions.

Table 5: Cost of one training step, reference model (width 128, batch 256): matrix-multiply GFLOP and ratio to AdamW, time per step in ms and peak GPU memory in MiB (Appendix C, *Cost per step*).

Method	GFLOP	× AdamW	ms	MiB
AdamW	0.97	1.00	4.3	28
Grokfast	0.97	1.00	4.9	29
MSQ*	1.93	2.00	8.3	41
SAM	1.93	2.00	9.3	29
EGR	2.90	3.00	10.6	46
GCAM_{VO}	2.82	2.92	15.6	146
GCAM_{all}	3.87	4.00	22.1	734

D HARDER SETTINGS: LEARNING RATES, RETENTION AND MULTIPLICATION

These results extend Table 1. Table 6 repeats width 48 across learning rates for AdamW and GCAM_{VO}; Table 7 and Figure 10 give retention after t_{99} at width 48 for every method; Table 8 gives 15% training data across learning rates; Table 9 compares all methods in the reference setting at learning rate .005; and Table 10 replaces addition by multiplication.

At learning rate .002, GCAM has a lower fraction of evaluations below 95% after t_{99} than MSQ on each of 6 paired seeds; cohort medians are 13.46% and 27.88%, respectively. This is relative retention, not absence of dips. SAM in the same condition has no recorded evaluation below 95% after t_{99} and finishes above 99% on 6/6. At learning rate .003, GCAM and MSQ finish above 99% on 1/6 and 0/6, respectively, despite reaching 99% earlier.

E GRADIENT GEOMETRY AND CIFAR-10 PROTOCOL

Participation ratio. For unit-normalized gradients $u_i = g_i/\|g_i\|$, let C be their cosine matrix and $C_0 = JCJ$, where $J = I - \mathbf{1}\mathbf{1}^\top/B_p$ centers the probe, and

$$r_{\text{eff}} = \text{tr}(C_0)^2/\|C_0\|_F^2 \tag{18}$$

the participation ratio, the effective number of directions the cloud of gradients occupies (Section 7).

Table 6: Width 48 across learning rates: modular addition, batch 256, 30% training data, weight decay 1, 20k updates, default data split, seeds 42–47 (fewer where a count has a smaller denominator). GCAM_{VO} uses $\lambda = .02$ at every learning rate. Median t_{99} is given only when every seed reaches 99%.

Learning rate	.0005	.001	.0015	.002	.003	.005	.007	.01	.02
AdamW, reach 99%	1/6	3/6	3/5	1/6	0/3	4/6	2/2	2/2	0/2
AdamW, end ≥99%	1/6	1/6	3/5	1/6	0/3	2/6	1/2	1/2	0/2
AdamW, median t_{99}	–	–	–	–	–	–	5,250	10,250	–
GCAM_{VO}, reach 99%	3/3	6/6	6/6	6/6	6/6	6/6	1/2	0/2	0/2
GCAM_{VO}, end ≥99%	3/3	6/6	5/6	6/6	6/6	6/6	1/2	0/2	0/2
GCAM_{VO}, median t_{99}	6,000	3,300	2,000	1,700	1,150	2,200	–	–	–

Table 7: Width 48, batch 256, 30% training data, learning rate 10^{-3}, seeds 42–47, 20k updates. "Stay" requires test accuracy ≥99% at every evaluation after t_{99}; "Lowest after t_{99}" is the minimum over seeds that reach 99%; the last column is the largest per-seed share of evaluations after t_{99} below 95%. MSQ* and EGD* (Saheb Pasand & Dohmatob, 2026) are the implementations described in Appendix B. For GCAM_{VO} the "Stay" count depends on the split of the data: on two other random splits it is 1/6 and 3/6 (all six seeds reach 99% and end above it on both).

Method	Reach 99%	Median t_{99}	Stay ≥99%	Lowest after t_{99}	Evaluations <95%
AdamW	3/6	–	0/6	97.46%	0.00%
Grokfast	6/6	6,450	5/6	98.85%	0.00%
MSQ*	6/6	4,650	0/6	57.46%	2.56%
SAM	3/6	–	2/6	98.42%	0.00%
EGD*	6/6	5,900	6/6	99.24%	0.00%
GCAM_{VO}	6/6	3,300	5/6	98.68%	0.00%

Connection to the interaction decomposition. For the unnormalized gradient Gram $K_{ij} = g_i^\top g_j$,

$$\frac{\operatorname{tr}(K)^2}{\|K\|_F^2} = \frac{D^2}{\sum_i \|g_i\|^4 + \sum_{i \neq j} (g_i^\top g_j)^2}. \tag{19}$$

This identity shows why scalar positive/negative sums do not determine directional concentration. Equation 18 additionally normalizes each nonzero sample gradient and centers the cloud; it is a different participation ratio. We calculate training and held-out quantities separately (256 and 128 fixed probe examples), use a 10^{-30} norm floor in the stored numerical analysis, and retain the full-dimensional statistics. The classifier head is excluded only in the stated CIFAR panels; this coordinate restriction is distinct from the scope of the penalty used during training.

The grokking cohort comprises seeds 42–47, width 48, MLP width 192, one layer, batch 256, fraction .30, LR .001, WD 1, and GCAM VO .02. CIFAR distribution analysis uses seeds 41–43 and the $\lambda = .05$ runs without augmentation (Table 2 reports the best tested, .1). The curves are recomposed from existing checkpoint analyses; no missing checkpoint is imputed. Common-update contrasts should not be interpreted as holding learning stage fixed.

CIFAR protocol and reporting. The model has four attention heads, MLP width 512, patch size 4, GELU, and a class-token readout. Batches are shuffled IID; crop/flip and no augmentation are separate conditions. All 50,000 training images are used. The final evaluation uses the trainer's fixed 2,000-image test subset, not the full CIFAR-10 test set. Table 2 reads the completion marker's evaluation after 39,100 updates. The last periodic log can precede training completion and gives slightly different numbers; it is not substituted for the final evaluation. Three common seeds are used because all displayed methods, including GCAM at every coefficient, have these seeds. Duplicate checkpoint reruns do not increase the seed count. Additional reference seeds and the full coefficient grid are not pooled with these seeds.

The all-parameter GCAM row penalizes N_c/B^2, with both gradient factors differentiated. At $B = 128$, the pair-average $N_c/[B(B-1)]$ is $128/127$ times larger for a fixed coefficient. EGR uses coefficient .02; SAM radius is .05; Grokfast uses amplification 2 and EMA coefficient .98. The MSQ variant estimates its second-moment correction from eight chunks, not individual samples,

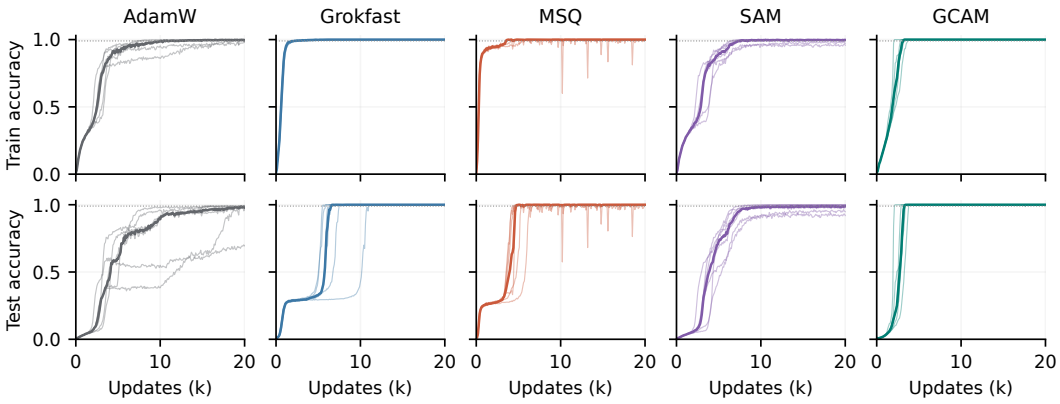


Figure 10: Training (top) and test (bottom) accuracy at width 48 (setting of Table 7) for the same six seeds under each method; thin lines are individual seeds, thick lines pointwise medians, without smoothing. On two other random splits of the data GCAM_{VO} again reaches 99% on all six seeds (medians 3,100 and 3,400 updates) but stays above it at every later evaluation on one and three.

Table 8: 15% training data across learning rates: modular addition, width 128, batch 256, weight decay 1, 120k updates, seeds 42–47. GCAM_{VO} uses $\lambda = .02$ at every learning rate; SAM $\rho = .05$, Grokfast $\lambda = 2$. Median t_{99} is given when every seed reaches 99%; "End" counts seeds with test accuracy $\geq$99% at the last evaluation.

Learning rate	Method	Reach 99%	Median t_{99}	End $\geq$99%
10^{-3}	AdamW	0/6	–	0/6
	SAM	6/6	40,750	6/6
	Grokfast	6/6	39,500	6/6
	MSQ	6/6	13,350	5/6
	GCAM_{VO}	6/6	50,800	5/6
$2 \cdot 10^{-3}$	AdamW	6/6	25,300	3/6
	SAM	6/6	13,250	6/6
	Grokfast	6/6	12,050	6/6
	MSQ	6/6	7,750	1/6
	GCAM_{VO}	6/6	8,150	5/6
$3 \cdot 10^{-3}$	AdamW	6/6	10,850	5/6
	SAM	6/6	8,100	5/6
	Grokfast	6/6	11,050	3/6
	MSQ	6/6	6,500	0/6
	GCAM_{VO}	6/6	5,900	1/6

and clips its preconditioned coordinate update at magnitude 3. Its poor fitting at LR 10^{-3} is a step-size effect, not a general failure of MSQ: at $3 \cdot 10^{-4}$, with depth, weight decay, augmentation and budget held fixed, it trains normally (Table 2).

E.1 Shared loss-slice visualization

Figure 7 reuses a matched cohort with modular addition modulo 97, one Transformer layer, width 48, four heads, MLP width 192, training fraction .3, batch 256, learning rate .001, weight decay 1 and seed 42. The runs share the initial parameters, data split and batch stream. GCAM penalizes all parameter coordinates with $\lambda = .005$ and no cap; EGR uses .01, SAM radius .05, Grokfast amplification 2 and EMA .98, and MSQ the repository implementation. This cohort is distinct from the VO comparisons in the main performance tables.

The affine slice (Li et al., 2018) passes through the shared initialization and GCAM's first memorization and sustained-generalization checkpoints (logged steps 2,900 and 3,100). Orthogonalized

Table 9: Reference setting at a high learning rate: modular addition, width 128, batch 256, 30% data, learning rate .005, weight decay 1, 20k updates, seeds 42–47. † Median over the seeds that reach 99%; ratio: median paired rival/GCAM time ratio over the n seeds on which both reach 99%. No method stays at 99% on any seed.

Method	Reach 99%	Median†	Ratio (n)	Stay ≥99%	Lowest after t_{99}	Evaluations <95%
AdamW	6/6	1,200	1.15 (6)	0/6	37.62%	27.60%
Grokfast	6/6	1,300	1.50 (6)	0/6	2.46%	13.16%
MSQ*	5/6	1,200	1.30 (5)	0/6	9.93%	45.74%
SAM	6/6	950	1.11 (6)	0/6	92.50%	1.05%
GCAM	6/6	900	1.00 (6)	0/6	21.97%	23.56%

Table 10: Multiplication extension: modulus 97, width 48, batch 256, fraction .30, LR .001, weight decay 1, 20k updates, seeds 42–47. GCAM coefficient .05. † Median over the seeds that reach 99%; the other seeds remain in the counts.

Method	Reach 99%	Median†	Stay ≥99%	Lowest after t_{99}	Evaluations <95%
Multiplication mod 97, width 48, learning rate .001					
AdamW	2/6	18,200	0/6	96.61%	0.00%
Grokfast	6/6	8,500	3/6	98.68%	0.00%
MSQ*	6/6	4,000	0/6	22.48%	1.88%
SAM	2/6	9,700	0/6	98.47%	0.00%
GCAM $\lambda = .05$	6/6	3,850	5/6	94.03%	0.59%

displacement directions define one origin and basis for all methods and both data splits. The contours evaluate task cross-entropy on all 2,822 training and 6,587 test examples; displayed trajectories are projections of saved parameter vectors. Initialization is stored separately from logged step 0, which follows the first update. The plotted time labels follow the saved logs.

The defining anchors lie on the plane; other checkpoints generally do not. At GCAM's step 3,100, the plane reproduces the actual test loss because this point is an anchor. Fidelity at an anchor does not validate the whole trajectory. The plane was chosen to examine this transition, not to establish a universally faithful two-dimensional landscape. The learning curves are evaluated on the actual checkpoints, including later instability. AdamW and EGR do not reach 99% training accuracy within this cohort's budget, so the comparison is not exclusively a post-memorization effect.

F Early prediction of seed outcomes

To test relevance before the transition, we evaluate a frozen predictor of failure to reach 99% test accuracy within 20k updates. Training accuracy and loss form the baseline; the augmented predictor also uses module-wise N_c. On 128 new width 128 seeds, including 24 failures, ROC-AUC increases from .786/.803/.871 to .905/.907/.952 at updates 100/300/500. The earliest observed memorization in this evaluation cohort is update 800, so all three readouts precede the first time training accuracy reaches 99%.

At the frozen update 500 decision threshold, 19 of 24 failures are detected with 7 false alarms among 104 successful runs. However, the same predictor is near chance on 64 new width 48 seeds, including 31 failures. This establishes informative early measurements in the reference condition, not a universal predictor across architectures. The frozen evaluation is distinct from earlier retrospective comparisons, where positive-interaction features were also informative (below). Predictive information motivates a component intervention; it does not by itself identify a cause.

At each observation time, a separate logistic-regression model ($C = 1$) is fitted on standardized features from 5,483 existing seeds, including 920 within-budget failures. The baseline uses training accuracy and loss; the augmented model adds eight module-wise coordinate-conflict measurements. Preprocessing and models are frozen before the new-seed evaluation; the operating threshold is selected from training scores for approximately 10% false-positive rate. The outcome is failure to reach 99% test accuracy within 20k updates. No fitting uses the prospective evaluation seeds.

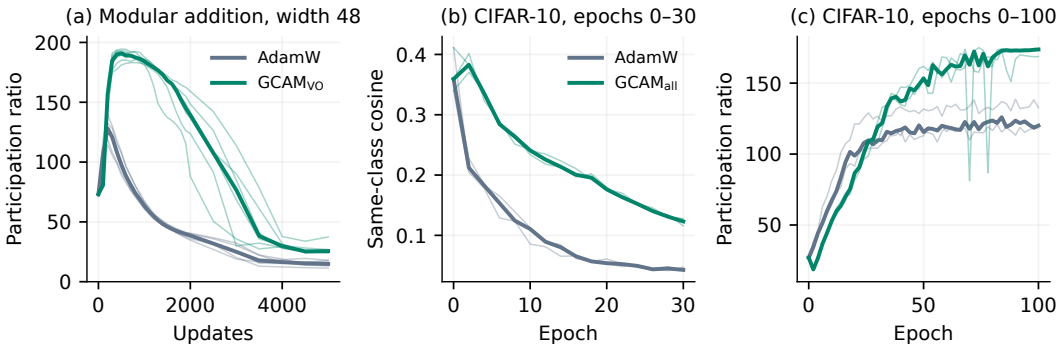


Figure 11: Spread of per-sample gradients. (a) Width 48 (setting of Figure 6): participation ratio of 256 unit-normalized training gradients over all parameters, six seeds. (b, c) CIFAR-10 without augmentation (setting of Table 2, GCAM$_{\text{all}}$ $\lambda = .05$): mean same-class cosine and participation ratio for 256 fixed training images, classifier weights excluded, three seeds. Thin lines: seeds; thick lines: pointwise medians.

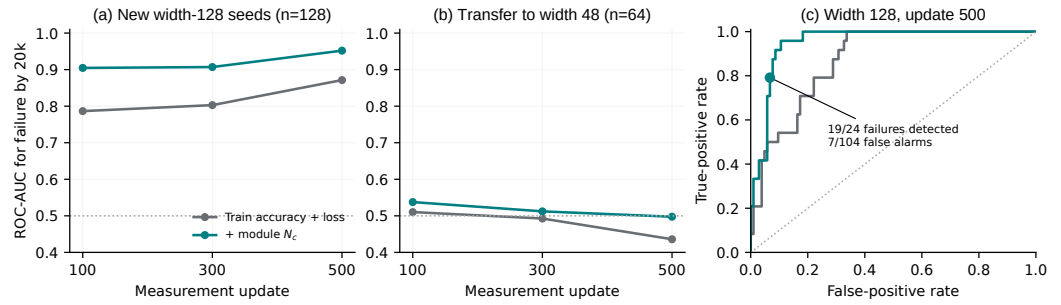


Figure 12: Prospective evaluation of frozen predictors. (a) New width-128 seeds: 128 runs, 24 failures. (b) Transfer to 64 new width-48 seeds: 31 failures; the added-interaction predictor is near chance. (c) At update 500 at width 128, the frozen operating point detects 19/24 failures with 7 false alarms. Plots use the frozen scores and complete outcome histories.

This protocol is separate from earlier retrospective log-feature cross-validation on 5,484 seeds with 921 failures. Positive-interaction features performed similarly in that earlier analysis; a prospective P_c comparison and incremental value over a separately fitted energy control are not established here. The width-transfer failure rules out treating these scores as a configuration-independent early-warning rule.