THE GEOMETRY OF FORGETTING IN NGPT: DISENTANGLING WEIGHT ROTATIONS AND SCALE DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

nGPT achieves high training efficiency by constraining its major vector-valued parameters to unit hyperspheres while learning explicit scale parameters. This parameterization naturally decomposes parameter updates into weight rotations and scale changes. However, when a pretrained model is further trained, it remains unclear which of these geometric changes harm the behavior learned during pretraining. We study this question in domain-adaptive continued pretraining, analyzing which kinds of rotations and scale changes drive forgetting. We show that, at matched rotation angles, rotational damage varies strongly with alignment to task-sensitive directions. At matched displacement sizes, scale damage varies with the spread of per-token logit reweighting. Based on these channel-specific mechanisms, we design a two-term regularizer and use it to intervene on the two channels separately. Weight rotations drive most net forgetting, while scale damage is largely offset by output-projection interactions. With both penalty terms, suppressing task-sensitive rotations exposes this scale damage, which the scale penalty then controls. nGPT's hyperspherical parameterization thus provides not only a mechanism for training efficiency but also a coordinate system for understanding and controlling change during continued pretraining.

1 INTRODUCTION

nGPT (Loshchilov et al., 2025) was introduced as a normalized Transformer and has since been shown to reach a given validation loss with substantially fewer training tokens than comparable unnormalized Transformers (Loshchilov & Ginsburg, 2026). A key feature of nGPT is that it constrains its major vector-valued parameters to unit hyperspheres while learning explicit scale parameters. This parameterization gives post-pretraining updates a natural geometric interpretation. We study how these geometric changes contribute to forgetting during domain-adaptive continued pretraining (Gururangan et al., 2020).

Forgetting during continued pretraining is commonly measured as increased loss on the pretraining distribution (Gupta et al., 2023; Ibrahim et al., 2024). Related work on continual learning and forgetting localizes updates to particular layers or parameter subsets (Fernandez et al., 2024) or protects parameters important to the pretrained solution using Fisher information (Kirkpatrick et al., 2017), but does not examine the geometry of parameter motion. A natural way to study this geometry is to separate motion into distinct components. A magnitude–direction decomposition can be introduced through reparameterization (Liu et al., 2024), whereas nGPT exposes this structure natively: normalized weights move by rotation, while explicit scale parameters change independently. This turns the broad question of which parameter changes drive forgetting into a more concrete one: which rotations and scale changes are responsible?

Answering this requires a model of how the pretraining loss responds to a displacement in nGPT's coordinates. Around the pretrained model, a Fisher quadratic form in the displacement provides one, and it decomposes exactly into a rotation term, a scale term, and a cross-channel interaction (§3). The two channels require different accounts. Rotational damage is directional: at matched angles, it varies with how much the rotation aligns with a low-dimensional sensitive subspace for each weight vector. Scale damage, carried chiefly by the output scales s_z, is distributional: at matched displacement sizes, it varies with the spread of per-token logit reweighting. Both mechanisms are validated by causal interventions on a 0.5B-parameter nGPT adapted to code, math, and medicine (§4). Finally we turn

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.

the two mechanisms into a two-term regularizer and use it primarily as a causal intervention that tests the account, rather than as a competitive retention method (§5).

Our contributions are three findings.

- **Two channels, two mechanisms.** In nGPT's native coordinates, parameter displacement decomposes into rotations of unit weight vectors and changes of explicit scales. These two channels cause functional deviation through distinct mechanisms. Rotational damage is governed by how much of the rotation falls in a low-dimensional task-sensitive subspace, rather than by the angle alone. Scale damage is governed by the spread of the per-token logit reweighting, rather than by the size of the change alone.
- **Both mechanisms are causal and persistent.** Rotating every weight vector by the same angle produces many-fold differences in forgetting depending only on direction. Scale damage tracks the spread of the logit reweighting, not its size, across 280 synthetic displacements with predictions fixed in advance. Directions identified as sensitive before continued-pretraining remain substantially more damaging than random directions throughout training.
- **The channels cancel at the output layer, and the account is actionable.** A rotation-only penalty substantially reduces forgetting in all three domains, and combining the rotation and scale penalties performs best. A scale-only penalty has little effect because reducing scale damage also removes its compensating interaction with output-projection rotations.

2 Related Work

nGPT and normalized parameterizations. nGPT constrains weight vectors to the unit hypersphere while learning explicit scale parameters (Loshchilov et al., 2025), with follow-up work improving its training recipe and studying learning-rate transfer across scale (Loshchilov & Ginsburg, 2026; Shigida et al., 2026). Transferable scaling has also been studied under other hyperspherical parameterizations (Ren et al., 2026). Direction–magnitude decompositions have a longer history in optimization (Salimans & Kingma, 2016; Hägele et al., 2026), and DoRA applies the split in adaptation (Liu et al., 2024). These works use such parameterizations primarily for optimization or adaptation; we use nGPT's native geometry to ask which parameter motions disrupt behavior learned during pretraining.

Forgetting in continued pretraining. Continued pretraining is a standard domain-adaptation recipe (Gururangan et al., 2020; Chen et al., 2023; Azerbayev et al., 2024), and its forgetting and adaptation dynamics have been studied across distribution shift, learning-rate schedules, data mixing, and model scale (Gupta et al., 2023; Ibrahim et al., 2024; Béthune et al., 2025; Wang et al., 2025; Liao et al., 2025). Forgetting has also been studied in continual fine-tuning (Luo et al., 2025; Li et al., 2024), continual pretraining of aligned models (Li & Lee, 2024), and broader post-training regimes (Harmon et al., 2026). Some apparent forgetting may instead reflect shifted task inference rather than lost capability (Kotha et al., 2024). We instead ask which parameter motions cause the loss increase on the pretraining distribution.

Localizing and constraining updates. Continual-pretraining updates concentrate in particular layers and parameter subsets (Fernandez et al., 2024). Forgetting can be mitigated by anchoring Fisher-important parameters to previously learned values, as in EWC (Kirkpatrick et al., 2017), or by restricting update directions to preserve previously learned functions (Farajtabar et al., 2020). Recent LLM methods constrain adaptation to selected subspaces (Wang et al., 2023; Lu et al., 2025; Xiong & Xie, 2026; Nayak et al., 2026), identify vulnerable directions in function space (Hidekel & Raviv, 2026), or select adaptation subspaces from target-data Fisher curvature (Feng et al., 2026). Learning without Forgetting (Li & Hoiem, 2016) preserves an old model's outputs during adaptation by using knowledge distillation (Hinton et al., 2015). Our analysis of the scale channel instead characterizes how displacements of one native coordinate, the output scales, distort those outputs. We explain and intervene on this mechanism rather than matching the behavior itself. Our local model shares EWC's Fisher quadratic form, while our rotation penalty is a tangent-space, Kronecker-factored approximation to this form. The key distinction is the coordinate system. In nGPT's native geometry, the same quadratic form separates exactly into rotation, scale, and interaction terms. This allows each channel to be tested by intervention and makes the scale channel's cancellation against output-projection rotations visible.

3 A GEOMETRIC ACCOUNT OF FORGETTING IN NGPT

Our central hypothesis is that the damage caused by an update depends not only on how much the parameters change, but also on how they change. We set up coordinates on nGPT's parameter manifold (§3.1), derive a Fisher-based local model of functional deviation that decomposes exactly over the two channels (§3.2), and characterize what makes each channel harmful (§3.3, §3.4).

3.1 PARAMETER GEOMETRY OF NGPT

nGPT normalizes the vectors forming its embeddings and weight matrices, and also keeps its hidden states at unit norm (Loshchilov et al., 2025). Each normalized parameter vector $w_i \in \mathbb{R}^{d_i}$ therefore satisfies

$$\|w_i\|_2 = 1, \qquad w_i \in S^{d_i-1}.$$

For an attention or MLP block $T \in \{A, M\}$ with normalized output h_T, nGPT updates

$$h \leftarrow \mathrm{Norm}\big(h + \boldsymbol{\alpha}_T \odot (h_T - h)\big),$$

where $\boldsymbol{\alpha}_T$ is a learned vector of eigen learning rates. The effects of magnitude removed by normalization are controlled separately through explicit learned scale parameters. For example, an output logit is

$$z_v = s_{z,v} \langle h, w_v \rangle,$$

so w_v determines the direction of the output projection vector while $s_{z,v}$ controls its logit scale. The normalized weight parameters therefore lie on a product of hyperspheres, while the explicit scale coordinates—including eigen learning rates and activation and logit scales, with s_z central to our analysis—form a Euclidean factor. We use the raw trainable scale parameters as coordinates; these are unconstrained, although the forward pass applies $|\cdot|$ to the eigen learning rates. The resulting parameter manifold is

$$\mathcal{M} = \left(\prod_{i=1}^{N} S^{d_i-1}\right) \times \mathbb{R}^M, \qquad S^{d_i-1} = \{w_i \in \mathbb{R}^{d_i} : \|w_i\|_2 = 1\}.$$

We measure the displacement between the pretrained model θ_A and a second model—the continued-pretraining endpoint θ_B unless stated otherwise—by the Riemannian logarithm on each factor (Absil et al., 2008). On a spherical factor, with tangent space $T_{w_A} = \{v : \langle w_A, v\rangle = 0\}$,

$$\xi_w = \mathrm{Log}_{w_A}(w_B) = \tfrac{\theta}{\sin\theta}\big(w_B - \cos\theta\, w_A\big), \tag{1}$$

where $\cos\theta = \langle w_A, w_B\rangle$, for non-antipodal w_A, w_B and with the continuous limit at $\theta = 0$. Then $\xi_w \in T_{w_A}$ and $\|\xi_w\| = \theta$: the magnitude of a rotation is its geodesic angle. On the Euclidean factors scale displacements are linear differences $\xi_s = s_B - s_A$. Every parameter displacement thus decomposes into *rotations* of normalized weight vectors and *scale* changes.

3.2 A LOCAL MODEL OF FUNCTIONAL DEVIATION

The quantity we ultimately want to explain is the standard forgetting measure

$$\Delta L_A = L_A(\theta_B) - L_A(\theta_A),$$

the increase of next-token loss on the pretraining distribution. Our analysis is built instead on the *functional deviation* from the old model (writing $p_A := p_{\theta_A}$, $p_B := p_{\theta_B}$),

$$\mathcal{K}_A = \mathbb{E}_x\big[D_{\mathrm{KL}}\big(p_A(\cdot|x) \,\|\, p_B(\cdot|x)\big)\big],$$

whose derivatives at θ_A admit exact expressions. The two are related by an identity, not an assumption: with q_A the true conditional of the pretraining distribution,

$$\Delta L_A - \mathcal{K}_A = \mathbb{E}_x \textstyle\sum_y (q_A - p_A) \log \frac{p_A}{p_B}, \tag{2}$$

a coupling between the old model's calibration error and the update, which vanishes when $p_A = q_A$. We do not assume it small but measure it: at the continued-pretraining endpoints $\Delta L_A/\mathcal{K}_A$ stays within 1.5% of unity (other perturbation classes: Appendix C.1), so $\mathcal{K}_A$ stands in for forgetting throughout.

The local model is a Taylor expansion of $\mathcal{K}_A$ around θ_A. Since $\mathcal{K}_A \geq 0$ with equality at θ_A, its gradient vanishes at θ_A without any convergence assumption, and on tangent directions its Riemannian Hessian is the true Fisher information (Appendix B.1):

$$\nabla^2 \mathcal{K}_A(\theta_A) = F_A := \mathbb{E}_x \mathbb{E}_{y\sim p_A}\big[g_A\, g_A^\top\big], \qquad g_A := \nabla_\theta \log p_A. \tag{3}$$

For a tangent displacement $\xi = (\xi_w, \xi_s) \in T_{\theta_A}\mathcal{M}$, with $\mathrm{Exp}_{\theta_A}(\xi) = \big(\mathrm{Exp}_{w_A}(\xi_w),\, s_A + \xi_s\big)$ componentwise, the second-order geodesic expansion is determined entirely by this Hessian because $\nabla \mathcal{K}_A(\theta_A) = 0$, $\mathrm{Hess}\,\mathcal{K}_A(\theta_A)[\xi,\xi] = \xi^\top F_A\, \xi$, so the only approximation is the truncation at finite displacement,

$$\mathcal{K}_A\big(\mathrm{Exp}_{\theta_A}(\xi)\big) = \mathcal{K}_A^{(2)}(\xi) + O(\|\xi\|^3), \qquad \mathcal{K}_A^{(2)}(\xi) := \tfrac{1}{2}\,\xi^\top F_A\, \xi, \tag{4}$$

whose validity radius we locate empirically (Appendix B.2). In the eigenbasis of F_A, this becomes $\mathcal{K}_A^{(2)} = \frac{1}{2}\sum_i \lambda_i \langle u_i, \xi\rangle^2$. At fixed displacement magnitude, deviation is larger when more of the displacement projects onto high-curvature directions.

Splitting F_A into the blocks matching the two components of ξ gives an exact decomposition of the quadratic model,

$$\mathcal{K}_A^{(2)} = \underbrace{\tfrac{1}{2}\xi_w^\top F_{ww}\xi_w}_{Q_w} + \underbrace{\tfrac{1}{2}\xi_s^\top F_{ss}\xi_s}_{Q_s} + \underbrace{\xi_w^\top F_{ws}\xi_s}_{I_{ws}}. \tag{5}$$

Q_w and Q_s are the within-channel quadratic terms and I_{ws} their interaction, so neither Q_w nor Q_s alone is a channel's total contribution to the deviation. Because nGPT separates the channels explicitly, we can intervene on each in isolation to identify its mechanism and measure their interaction separately at the endpoint (§5.3). The next two questions are which directions are high-curvature for rotations (§3.3) and which relative logit distortions are high-curvature for scales (§3.4).

3.3 Sensitive Directions of Weight Rotations

The weight factor $\prod_i S^{d-1}$ is a product manifold, so each weight vector w_i carries its own tangent space and Q_w splits exactly as

$$Q_w = \textstyle\sum_i Q_{w_i} + \sum_{i<j} I_{ij}^w, \quad Q_{w_i} = \tfrac{1}{2}\xi_i^\top F_{ii}\,\xi_i,$$

where F_{ii} is the canonical restriction of the Fisher to the i-th spherical factor, defined on the tangent space via $P_{w_i} = I - w_i w_i^\top$:

$$F_{ii} = \mathbb{E}_x \mathbb{E}_{y\sim p_A}\big[g_i^{\tan}(g_i^{\tan})^\top\big], \qquad g_i^{\tan} = P_{w_i}\nabla_{w_i}\log p.$$

We do not equate $\sum_i Q_{w_i}$ with the total deviation—the cross-row terms remain unmodeled—but validate total effects by measuring ΔL_A directly.

The curvature of the old-task KL when w_i is rotated toward a unit tangent direction v is $v^\top F_{ii} v$; the span U_k of the top-k eigenvectors of F_{ii} is the row's *sensitive subspace*. The curvature along a unit direction inside it lies between λ_k and λ_1, and at most λ_{k+1} in its orthogonal complement, so a rotation by θ costs between $\frac{1}{2}\lambda_k\theta^2$ and $\frac{1}{2}\lambda_1\theta^2$ inside and at most $\frac{1}{2}\lambda_{k+1}\theta^2$ outside: same rotation angle, very different deviation. Empirically this subspace is low-dimensional in every matrix type: across all 96 query, key, value and MLP up-projection matrices, the leading $k = 64$ tangent directions carry a median 90% of the row's Fisher energy (range 76–98%), the MLP down-projection is nearly as concentrated (median 85% over its normalized columns), and even the least concentrated type, the attention output projection, reaches a median 63% (Figure 6, Appendix C.2).

To quantify where actual displacements go, note that a uniformly random tangent direction has $\mathbb{E}[\|U_k^\top \xi\|^2/\|\xi\|^2] = k/D$ ($D = d-1$), so we define the *excess concentration*

$$\rho = \frac{\|U_k^\top \xi\|^2/\|\xi\|^2}{k/D}, \tag{6}$$

equal to 1 for random drift; we do not model the origin of excess concentration.

Concentration alone does not determine the deviation: ρ ignores the eigenvalues, and sensitivity varies substantially even within the leading subspace. The regularizer of §5.1 therefore weights the retained directions by their Fisher eigenvalues rather than treating them uniformly. For tractability, the implementation retains the leading $k = 64$ directions; Appendix C.2 quantifies their Fisher-energy coverage and gives the shared-basis approximation used in practice, and Appendix B.1 bounds when the sensitive subspace dominates.

3.4 Relative Logit Distortion from Scale Updates

For the scale channel Eq. (5) already supplies the local theory, $Q_s = \frac{1}{2}\,\xi_s^\top F_{ss}\,\xi_s$; its mechanism differs from the rotations'. We state the result for the output scales s_z, which carry most of the channel's damage (Appendix C.4). At the output layer $z_v = s_v u_v$ with $u_v = \langle h, w_v \rangle$, a displacement δs of the output scales shifts the logits by exactly $u \odot \delta s$, and from the closed-form score $\partial \log p_y / \partial s_v = (\mathbb{1}[y{=}v] - p_v)u_v$, writing $m := u \odot \delta s$,

$$E_S(\delta s) := \delta s^\top F_{s_z s_z}\,\delta s = \mathbb{E}_x\big[\mathrm{Var}_{y\sim p_A}(m_y)\big], \tag{7}$$

the *scale Fisher energy*: the spread, over the tokens the old model considers plausible, of the per-token logit reweighting the displacement induces. By Eq. (4) the predicted deviation of a scale-only displacement is $\frac{1}{2}E_S + O(\|\delta s\|^3)$; we call the range over which the quadratic term dominates the *quadratic regime*, whose extent depends on the direction through higher cumulants (Appendix B.3). The claim is the mirror image of §3.3: what governs this channel is not the size of the scale displacement but the relative logit distortion it creates—a shift common to all logits has zero variance and costs nothing, while a displacement of the same norm concentrated on task-A's plausible tokens can be maximally damaging. §4.3 tests this by varying E_S at controlled displacement size; when only s_z moves an exact global theory is available, which enables those measurements and explains their departures from the quadratic prediction (Appendix B.3).

4 Empirical Validation

The predictions of §3 follow from local Fisher geometry alone, so whether they hold on a real 0.5B model is a separate question. After describing the setting (§4.1), we test each prediction by controlled intervention: sensitive directions for rotations (§4.2), relative logit distortion for scales (§4.3).

4.1 Experimental Setting

We pre-train a $\approx$0.5B-parameter nGPT on OpenWebText (Gokaslan & Cohen, 2019) for $\approx$5.2B tokens using the unmodified official NVIDIA implementation[1] (Loshchilov et al., 2025), which builds on nanoGPT (Karpathy, 2022), following the released recipe (Appendix D). This model is task A (θ_A), and its corpus defines the distribution whose loss we track. Following the domain-adaptive pretraining setting (Gururangan et al., 2020), we then continue pretraining θ_A on three target domains: *code*, *math*, and *medicine*. We use CodeParrot-clean (CodeParrot, 2022), OpenWebMath (Paster et al., 2024), and the PubMed abstracts of the MedRAG corpus (Xiong et al., 2024), respectively, for 300M tokens each with no replay, so the unregularized runs measure unmitigated forgetting. Adaptation is measured by validation loss on the target domain, while forgetting is measured as ΔL_A (§3.2). The coupling Eq. (2) is calibrated on every perturbation class used below (Appendix C.1). Sensitive subspaces are estimated from task-A data alone, before adaptation (ranks in Appendix D).

4.2 Analysis of Weight Rotations

Directional causality. We rotate every weight vector of θ_A by the same angle and vary only the direction: into the sensitive subspaces (their leading $k = 64$ directions, the rank the regularizer of §5 protects), uniformly at random, or within the orthogonal complements. In the small-angle regime ($\theta \leq 1^\circ$), all three classes scale as θ^2, appearing as parallel lines in Figure 1a. The constant vertical gaps isolate the effect of direction: sensitive-direction rotations produce $7\times$ the functional deviation of random rotations and $12\times$ that of orthogonal rotations, while random and orthogonal directions differ far less. At matched angles, direction changes the damage by roughly an order of magnitude, and this ordering persists well beyond the radius in which the quadratic model is accurate (Appendix B.2). Since the subspaces are estimated from task-A data alone, before any continued pretraining, this is a causal test rather than a circular fit, and the directional ordering persists along the continued-pretraining trajectories: directions identified as sensitive at θ_A remain about $7\times$ more damaging than a random direction at the endpoints (Appendix B.2).

[1] `https://github.com/NVIDIA/ngpt`

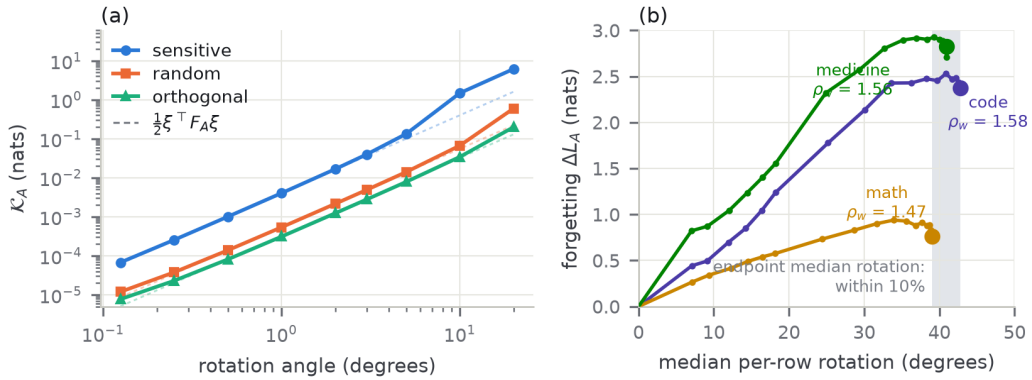

Figure 1: Direction modulates rotational damage at matched angles. (a) All weight vectors of θ_A are rotated by the same angle while only the direction varies; vertical gaps isolate the effect of direction (solid: measured $\mathcal{K}_A$; dashed: quadratic model). (b) For the three unregularized runs, forgetting is plotted against median per-row rotation along training; large dots mark endpoints, whose rotation medians differ by less than 10%, and labels show endpoint ρ_w.

Same rotation, different forgetting. The causal pattern is visible in the actual continued-pretraining runs, traced checkpoint by checkpoint in Figure 1b. Endpoint rotation magnitudes are nearly identical across the three domains—median per-row rotations are within 10% of each other—yet forgetting differs by $3.7\times$, so median rotation angle alone does not explain forgetting. The separation is a property of the whole path, not only of the endpoints: read at any common rotation angle, the three runs' forgetting already differs several-fold (2.1/0.9/2.6 nats at 30° on code/math/medicine). The displacements themselves land preferentially on the sensitive subspaces: their excess concentration Eq. (6) is $\rho_w = 1.5$–1.6, against 1 for a random direction (Appendix C.1).

4.3 Analysis of Scale Updates

Spread causality. We construct 280 synthetic s_z displacements whose sizes and induced logit distortions are controlled independently, across pattern, shuffled, random, uniform, top-Fisher and rare-token families (construction in Appendix C.1), spanning eleven orders of magnitude in E_S. Figure 2a plots each probe's deviation against its leading-order prediction $\frac{1}{2}E_S$, computed from the pretrained model alone. Across the bulk of the sweep the measured deviation follows the prediction closely, and the two kinds of departure are both predicted by the exact scale-only theory (Appendix B.3). The branches above the line are the rare-token families: they move tokens to which p_A assigns almost no mass, so E_S predicts almost no damage until the shift lifts those tokens into plausibility and higher cumulants take over. The tail below the line collects the largest logit distortions (the largest norms of every family, and the top-Fisher direction at every norm): the softmax saturates, so the deviation grows only linearly in the shift while $\frac{1}{2}E_S$ grows quadratically. Since the displacements are applied interventionally and their E_S is fixed from the pretrained model beforehand, the agreement is a causal, out-of-sample test of the local prediction of §3.4.

In the wild, spread—not size alone—sets the damage. Figure 2b follows the scale channel along real training. At every checkpoint of the three unregularized runs, that checkpoint's s_z is transplanted alone into θ_A ($\approx 0.01\%$ of parameters), and the resulting damage is plotted against the accumulated drift. Medicine accumulates less scale drift than code (0.42 vs. 0.48) but causes more damage, consistent with the annotated $\frac{1}{2}E_S$ values. At the matched norm marked by the diamonds, the three runs already differ in damage (0.14–0.20 nats), in the order of their own $\frac{1}{2}E_S$ at those checkpoints. The checkpoint damages track the quadratic prediction $\frac{1}{2}E_S$, with the endpoint deviations explained by the exact theory (Appendix B.3).

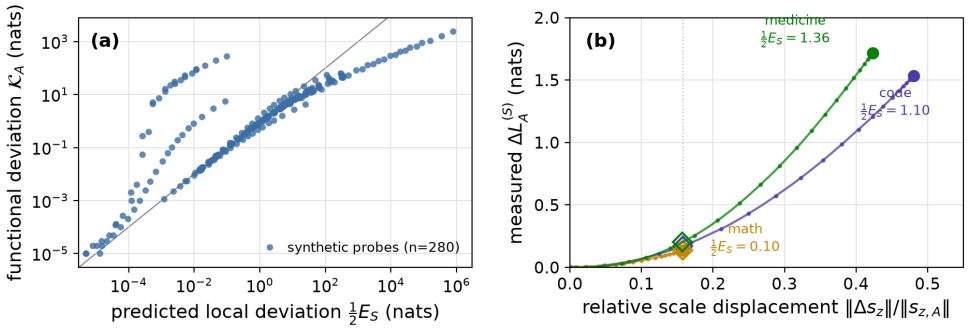

Figure 2: Scale-channel causality and verification. (a) Synthetic s_z displacements: evaluated deviation $\mathcal{K}_A$ against the prediction $\frac{1}{2}E_S$ from Eq. (7); the gray line is $y = x$, the branches above it are the rare-token families, and the tail below it the largest displacements. (b) The three unregularized runs evaluated at every checkpoint: damage from transplanting that checkpoint's s_z alone into θ_A, against the size of the drift so far; endpoints are annotated with $\frac{1}{2}E_S$, and the diamonds mark each run at the same displacement norm (dotted guide).

5 From Mechanism to Intervention

To test whether the identified mechanisms can guide forgetting reduction, we derive a two-term regularizer from them (§5.1), evaluate it across conditions and domains (§5.2), and explain one surprising result (§5.3).

5.1 Channel-Specific Regularization

To control the two channels separately, we use their respective terms in the local form $\frac{1}{2}\xi^\top F_A \xi$ as regularizers targeting the mechanisms of §3.3 and §3.4. Writing $\xi_w = \mathrm{Log}_{w_A}(w)$ and $\delta s = s_z - s_A$ for the displacements at the current parameters during training,

$$R_w = \mu_w \, \xi_w^\top F_{ww} \, \xi_w = \mu_w \textstyle\sum_i \lambda_i \big\langle u_i, \mathrm{Log}_{w_A}(w) \big\rangle^2, \tag{8}$$

$$R_{s_z} = \mu_s \, E_S(\delta s) = \mu_s \, \delta s^\top F_{s_z s_z} \, \delta s = \mu_s \, \mathbb{E}_x \, \mathrm{Var}_{y \sim p_A} \big[u_{A,y}(x) \, (s_{z,y} - s_{A,y}) \big], \tag{9}$$

and minimize $L_B + R_w + R_{s_z}$. R_w and R_{s_z} are both channel-restricted quadratic penalties, but they target different structures. R_w penalizes rotations toward task-A-sensitive directions, whereas R_{s_z} penalizes scale changes that distort relative token scores on task A. The loss thus controls the two mechanisms separately, in the coordinates in which each was causally tested; their interaction is measured rather than modeled in §5.3 (a separable bound on the full-block form: Appendix B.1).

Eq. (8) is the canonical local objective, but implementing R_w requires choosing how to scale its contributions across the model's weight matrices (e.g., query, key, value, attention-output and MLP projections), whose raw Fisher magnitudes vary widely and are poorly calibrated for setting relative penalty strengths. We therefore keep the Fisher eigendirections and relative spectrum within each weight matrix, but normalize each weight matrix's retained spectrum to unit mean (Appendix C.2). This keeps each weight matrix's directional geometry and discards the unreliable cross-matrix scale.

Implementation. Both penalties use task-A statistics estimated at θ_A and held fixed: the rotation term uses the per-row shared-basis approximation of Appendix C.2, while the scale term uses a closed form over cached pretrained token statistics. Both are therefore fixed local surrogates for the mechanisms, and their effects over a full run are evaluated empirically in Table 1. Although the quadratic approximation is local, the pretrained sensitive directions retain their causal ranking along the continued-pretraining trajectories (Appendix B.2, Figure 4).

5.2 Results on Continued Pretraining

We continue pretraining on each domain under four conditions—unregularized, scale penalty only, rotation penalty only, and both—with coefficients fixed once and shared across domains, three data-

	code			math			medicine		
Condition	adapt.↓	L_A ↓	ΔL_A ↓	adapt.↓	L_A ↓	ΔL_A ↓	adapt.↓	L_A ↓	ΔL_A ↓
θ_A, no FT	*2.95*	*2.76*	—	*3.25*	*2.76*	—	*3.27*	*2.76*	—
unregularized	0.80	5.13	$+2.372 \pm .006$	2.44	3.52	$+0.764 \pm .001$	2.33	5.59	$+2.830 \pm .004$
rotation only	0.86	3.13	$+0.372 \pm .002$	2.46	2.90	$+0.138 \pm .000$	2.43	3.62	$+0.867 \pm .008$
scale only	0.80	5.10	$+2.347 \pm .003$	2.44	3.53	$+0.768 \pm .001$	2.33	5.50	$+2.739 \pm .007$
both	0.86	3.02	$\mathbf{+0.265 \pm .002}$	2.46	2.89	$\mathbf{+0.132 \pm .000}$	2.43	3.31	$\mathbf{+0.550 \pm .010}$

Table 1: Continued pretraining under four conditions, coefficients fixed once across domains: adaptation (domain validation loss), task-A loss L_A, and forgetting ΔL_A, over three data-order seeds evaluated on identical fixed batches: ΔL_A is mean $\pm$ half-range (per-seed values in Table 9), adaptation and L_A the seed means. Bold marks the lowest regularized forgetting per domain.

order seeds per cell. These runs test whether regularizing the identified mechanisms changes forgetting as predicted, while the single-channel interventions reveal how the two channels contribute in the presence of their interaction; they are not a mitigation benchmark. Across every domain and seed (Table 1), the rotation-only penalty removes 69–84% of forgetting, the scale-only penalty changes little (within 3.2% of unregularized, adaptation unchanged to two decimals), and the combined penalty removes 81–89% at an adaptation cost of at most 0.10 nats, with the margin over rotation-only largest on code and medicine and negligible on math—the domain ordering §4.3 predicted.

Is the penalty doing more than slowing learning down? A single (adaptation, forgetting) pair cannot separate a penalty that protects task A selectively from one that merely shortens the distance traveled, so we trace trade-off curves on code: the unregularized run with its schedule rescaled to reduced token budgets, the unregularized run at the full budget with the peak learning rate divided by 2 to 16 (the same number of smaller steps, hence the closest unregularized analogue of a penalty that merely shrinks updates), and the combined penalty at reduced coefficients (Figure 7). The curves separate immediately: at a matched adaptation loss of 0.86 the reduced-budget run has accumulated $+2.07$ nats of forgetting and the reduced-learning-rate run about $+1.2$, against $+0.27$ for the penalty, so the penalty moves the model onto a different curve rather than a shorter or slower way along the same one. Which directions are protected also matters: a rank- and spectrum-matched penalty on random subspaces reduces forgetting, but at any matched level of protection incurs at least twice the adaptation cost (Appendix C.3). The same appendix compares against an EWC-style diagonal-Fisher weight anchor built from the same task-A statistics, which traces a worse trade-off because it suppresses rotation broadly rather than redirecting it.

5.3 Why Rotation and Scale Regularization Behave Differently

The obvious diagnosis of the scale-only row is that the penalty was too weak. The opposite is true: the audit below shows it suppressing exactly what it targets—the scale channel's standalone deviation collapses in every domain, by more than an order of magnitude on code and medicine—while total forgetting barely moves (Figure 3). The discrepancy lies in how the two channels combine, and the audit below measures that interaction directly.

Auditing where the damage sits. We audit each endpoint with the instrument of §4.3: hold one channel at θ_A—s denoting all explicit scales—and measure the exact deviation of the other,

$$\mathcal{K}_w = \mathbb{E}_x D_{\mathrm{KL}}\big(p_A \,\|\, p(w_B, s_A)\big), \qquad \mathcal{K}_s = \mathbb{E}_x D_{\mathrm{KL}}\big(p_A \,\|\, p(w_A, s_B)\big). \tag{10}$$

Their failure to add up defines an exact endpoint analogue of the interaction in Eq. (5),

$$I_{\mathrm{end}} := \mathcal{K}_A(\theta_B) - \mathcal{K}_w - \mathcal{K}_s, \tag{11}$$

which coincides with I_{ws} only to leading order and is reported as a measurement. Figure 3 decomposes every condition on every domain this way; interpolating the two displacements independently shows the interaction is smooth in both fractions, not an endpoint artifact (Appendix C.4, Figure 9).

Damage and cancellation vanish together. At the unregularized endpoints, the rotations alone ($\mathcal{K}_w$) reproduce 100–110% of the full endpoint deviation $\mathcal{K}_A$, so the two channels combine strongly

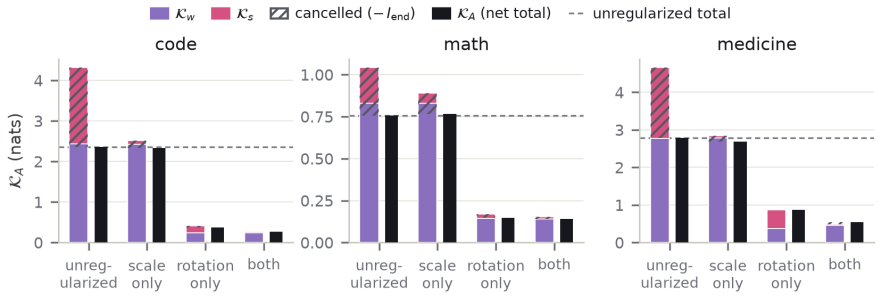

Figure 3: Exact endpoint decomposition $\mathcal{K}_A = \mathcal{K}_w + \mathcal{K}_s + I_{\text{end}}$ Eqs. (10)–(11). Left bar per condition: the standalone channel deviations, hatched where the negative interaction cancels them; black bar: the net $\mathcal{K}_A$; dashed guide: the unregularized total. Under scale-only the $\mathcal{K}_s$ block and its cancellation vanish together and the net bar stays on the guide; under rotation-only the $\mathcal{K}_s$ residue is no longer cancelled (clearest on medicine), which adding the scale term removes.

non-additively: the standalone scale deviation is almost exactly offset by the interaction. On code, $\mathcal{K}_s = +1.88$ nats while $I_{\text{end}} = -1.95$. The scale penalty sharply suppresses $\mathcal{K}_s$, but the negative interaction collapses with it. Their sum therefore remains near zero, while $\mathcal{K}_w$ changes little. Medicine shows the same pattern, and math a much smaller version of it.

At the output layer, output-weight rotations and output-scale changes both directly affect the same logits even with the hidden state fixed. For $z_v = s_v\langle h, w_v\rangle$, the first-order change is

$$\delta z_v = u_v\,\delta s_v + s_v\langle h, \delta w_v\rangle. \tag{12}$$

This shared-logit structure allows strong interaction; its sign and magnitude are measured, not derived.

The interaction is localized primarily to the output projection. Across all three domains, with the other rotation group held at θ_A, scale displacement interacts negatively with the output-projection rows and positively with the remaining rotations, and both change substantially once that group moves to its endpoint (Appendix C.4, Table 6). The output-row rotations are themselves strongly cancelled by the remaining rotations. Together, these effects indicate a three-way interaction consistent with a compensatory output-layer configuration adapted to a shifted internal representation.

Substitutable at the output, asymmetric under intervention. Eq. (12) makes the two channels locally substitutable at the output, but the full rotation channel is not symmetric with the scale channel. Non-output rotations also cause substantial standalone deviation, which is strongly cancelled by output-row rotations (Appendix C.4, Table 6). Suppressing rotations therefore reduces $\mathcal{K}_w$ and total forgetting, even as this cancellation collapses. On medicine, the rotation penalty leaves $\mathcal{K}_s = 0.50$ while I_{end} falls to -0.003, so the remaining scale deviation contributes almost fully. Adding the scale term removes this residue, with the largest gains on code and medicine and little gain on math, consistent with §4.3. Thus scale-only regularization suppresses standalone damage that was already largely cancelled, whereas rotation regularization suppresses damage that remains uncancelled.

6 Conclusion

We asked which geometric changes drive forgetting in nGPT, whose native coordinates separate weight rotations from explicit scale changes. At matched rotation angles, rotational forgetting varies strongly with alignment to task-sensitive directions. At matched scale-displacement sizes, scale damage varies with the spread of the per-token logit reweighting. Intervening on the two channels separately with a regularizer built from these mechanisms shows that weight rotations drive most of the net forgetting. Scale damage, in contrast, is largely cancelled by cross-channel interactions involving the output projection and becomes exposed once sensitive rotations are suppressed. Constraining both channels, with coefficients fixed once, removes most forgetting across all three domains. The scope and limitations of these conclusions are discussed in Appendix A. nGPT's parameterization provides not only a mechanism for training efficiency but also a coordinate system for understanding and controlling change during continued pretraining.

Reproducibility Statement

Appendix D gives the complete training and evaluation configuration (Table 8), the data-splitting protocol for every statistic that enters a training objective, per-seed results (Table 9), and the checkpointing cadence used for the trajectory figures. All datasets are public and are cited in §4.1; pretraining uses the unmodified official nGPT implementation, and continued pretraining adds a minimal patch whose components are listed in Appendix D. The Fisher-geometric results underlying §3 are proved in Appendix B.1, the validity range of the local model is characterized in Appendices B.2 and C.1, and the exact scale-only theory is derived in Appendix B.3. The construction of the rotation and scale probes (Appendix C.1), the implementation of the regularizer including its Fisher statistics and normalization (Appendix C.2), and the controls and endpoint audits (Appendices C.3 and C.4) are documented in sufficient detail to support reimplementation from the official code.

AI Use Statement

We used ChatGPT and Claude Code to assist with literature retrieval, theoretical development and mathematical derivations, coding and debugging, executing experiments, and drafting and revising the manuscript. The authors reviewed and verified the AI-assisted theoretical and mathematical content and experimental results, determined which material to include and how to present it in the manuscript, interpreted the findings, and take full responsibility for the final content of this work.

References

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. URL https://sites.uclouvain.be/absil/amsbook/.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.10631.

Louis Béthune, David Grangier, Dan Busbridge, Eleonora Gualdoni, Marco Cuturi, and Pierre Ablin. Scaling laws for forgetting during finetuning with pretraining data injection. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4020–4042. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bethune25a.html.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. MEDITRON-70B: Scaling medical pretraining for large language models, 2023. URL https://arxiv.org/abs/2311.16079.

CodeParrot. CodeParrot-clean: Deduplicated and filtered Python code from GitHub. Hugging Face dataset, https://huggingface.co/datasets/codeparrot/codeparrot-clean, 2022.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773. PMLR, 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Zhi-Quan Feng, Ying-Jia Lin, and Hung-Yu Kao. Learning in the Fisher subspace: A guided initialization for LoRA fine-tuning, 2026. URL https://arxiv.org/abs/2605.01046.

Jared Fernandez, Yonatan Bisk, and Emma Strubell. Gradient localization improves lifelong pretraining of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16188–16195. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.949. URL https://aclanthology.org/2024.findings-emnlp.949/.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus. https://Skylion007.github.io/OpenWebTextCorpus, 2019.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re)warm your model?, 2023. URL https://arxiv.org/abs/2308.04014.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Alexander Hägele, Alejandro Hernández-Cano, Atli Kosson, and Martin Jaggi. Improving neural network training by decoupling the magnitude and direction of weight vectors, 2026. URL https://arxiv.org/abs/2606.25971.

Jackson Harmon, Andreas Hochlehnert, Matthias Bethge, and Ameya Prabhu. Mapping post-training forgetting in language models at scale. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/bc1c5e5fb8ed1ef9b9b5abced2022e40-Abstract-Conference.html.

Ido Nitzan Hidekel and Dan Raviv. Catastrophic forgetting is low-rank: A function-space theory for continual adaptation, 2026. URL https://arxiv.org/abs/2606.18024.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=DimPeeCxKO.

Andrej Karpathy. nanoGPT. GitHub repository, 2022. URL https://github.com/karpathy/nanoGPT.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://doi.org/10.1073/pnas.1611835114.

Suhas Kotha, Jacob Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/692ae28fda9bfbde7c01b13bf5a03395-Abstract-Conference.html.

Chen-An Li and Hung-Yi Lee. Examining forgetting in continual pre-training of aligned large language models, 2024. URL https://arxiv.org/abs/2401.03129.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4297–4308. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.249. URL https://aclanthology.org/2024.findings-emnlp.249/.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Computer Vision – ECCV 2016*, pp. 614–629, Cham, 2016. Springer International Publishing. doi: 10.1007/978-3-319-46493-0_37. URL https://link.springer.com/chapter/10.1007/978-3-319-46493-0_37.

Chonghua Liao, Ruobing Xie, Xingwu Sun, Haowen Sun, and Zhanhui Kang. Exploring forgetting in large language model pre-training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2112–2127. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.105. URL https://aclanthology.org/2025.acl-long.105/.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.09353.

Ilya Loshchilov and Boris Ginsburg. Training nGPT, 2026. URL https://arxiv.org/abs/2608.01284.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. nGPT: Normalized transformer with representation learning on the hypersphere. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b7c12689a89e98a61bcaa65285a41b7c-Abstract-Conference.html.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.940. URL https://aclanthology.org/2025.acl-long.940/.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231. URL https://doi.org/10.1109/TASLPRO.2025.3606231.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417. PMLR, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Oleg Silkin, Mustafa Eyceoz, Hao Wang, Aldo Pareja, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in LLMs for continual learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/22620cccae29bbb3f18d226ea0320ff5-Abstract-Conference.html.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An open dataset of high-quality mathematical web text. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5949a8750a110ce1f0631b1776c500a2-Abstract-Conference.html.

Liliang Ren, Yang Liu, Yelong Shen, and Weizhu Chen. Rethinking language model scaling under transferable hypersphere optimization, 2026. URL https://arxiv.org/abs/2603.28743.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, pp. 901–909, 2016. URL https://arxiv.org/abs/1602.07868.

Boris Shigida, Boris Hanin, and Andrey Gromov. Learning rate transfer in normalized transformers, 2026. URL https://arxiv.org/abs/2604.27077.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Xingjin Wang, Howe Tissue, Lu Wang, Linjing Li, and Daniel Dajun Zeng. Learning dynamics in continual pre-training for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64525–64548. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25cx.html.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6233–6251. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.372. URL https://aclanthology.org/2024.findings-acl.372/.

Yifeng Xiong and Xiaohui Xie. OPLoRA: Orthogonal projection LoRA prevents catastrophic forgetting during parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34088–34096, 2026. doi: 10.1609/aaai.v40i40.40703. URL https://ojs.aaai.org/index.php/AAAI/article/view/40703.

A Limitations

Scale, language, and evaluation. Our results are established for a 0.5B-parameter model with a 5.2B-token pretraining budget and English-only corpora; whether the identified mechanisms persist at substantially larger scales remains untested. At the same time, this model size is consistent with prior nGPT work (Loshchilov et al., 2025), making it a well-motivated setting for the present mechanistic experiments. Forgetting is measured as held-out next-token loss on the pretraining distribution, which captures distributional retention but does not establish retention of capabilities such as reasoning, factual knowledge, or instruction following. The intervention results use three data-order seeds per condition, whereas the mechanism analyses and endpoint audits are reported for a single seed.

Local approximation and omitted interactions. Our analysis relies on a local Fisher approximation around the pretrained model. We use it to identify task-sensitive directions rather than as a globally accurate model of endpoint damage (Appendix B.2), and we do not model the evolution of the sensitive subspaces during continued pretraining. The per-row treatment omits cross-row interactions, and the implemented regularizer is a tractable approximation to the full Fisher geometry (Appendix C.2). Cross-channel cancellation is measured empirically rather than derived from the local theory (§5.3, Table 6).

The intervention is a mechanism test, not a mitigation benchmark. The regularizer is used as a mechanistic intervention rather than proposed as a general forgetting-mitigation method. We therefore evaluate it against controls that test the identified parameter-space mechanisms (Appendix C.3), rather than attempting a comprehensive comparison with retention methods based on different intervention principles. The analysis also relies on nGPT's native rotation–scale parameterization; extending the decomposition to unnormalized Transformers requires estimating an analogous geometry and is beyond this paper's scope. Our treatment of the interaction is also incomplete: the local Fisher cross term I_{ws} and the finite-displacement measurement I_{end} coincide only to leading order. Relating them beyond the local regime, and designing regularizers that explicitly control this interaction, remain open. Finally, the pretraining corpus already contains code and math, so domain overlap may partially mask forgetting and make our aggregate estimates conservative.

B Theory and scope of the local model

B.1 Proofs

Why the local KL curvature is the Fisher Eq. (3). The local model in Eq. (4) treats the Fisher as the quadratic curvature of the functional deviation $\mathcal{K}_A$ around the pretrained model. We justify that

identification here. Let

$$k(\theta) = \mathbb{E}_x \mathbb{E}_{y\sim p_A}[\log p_A - \log p_\theta].$$

Its gradient is

$$\nabla k(\theta) = -\mathbb{E}_x \mathbb{E}_{y\sim p_A}[\nabla_\theta \log p_\theta].$$

At $\theta = \theta_A$, the model distribution inside the expectation is p_A, so by the score identity,

$$\nabla k(\theta_A) = -\mathbb{E}_x \mathbb{E}_{p_A}[g_A] = 0.$$

Thus the quadratic term does not require θ_A to be a stationary point of the pretraining loss.

Differentiating once more and using the information-matrix equality gives

$$\nabla^2 k(\theta_A) = -\mathbb{E}_x \mathbb{E}_{p_A}[\nabla^2 \log p_A] = \mathbb{E}_x \mathbb{E}_{p_A}[g_A g_A^\top] = F_A.$$

Because the normalized weights lie on spheres, the relevant second derivative is along a geodesic γ on the parameter manifold. For $\xi = \dot\gamma(0)$,

$$\frac{d^2}{dt^2} k(\gamma(t))\bigg|_{t=0} = \xi^\top F_A \xi + \big\langle \nabla k(\theta_A), \mathrm{II}(\xi,\xi)\big\rangle.$$

The second-fundamental-form term vanishes because $\nabla k(\theta_A) = 0$. Hence the Riemannian Hessian of $\mathcal{K}_A$ on tangent directions is the Fisher quadratic,

$$\mathrm{Hess}\, \mathcal{K}_A(\theta_A)[\xi,\xi] = \xi^\top F_A \xi,$$

which yields Eq. (4).

Why the leading Fisher eigenspace captures sensitive rotations. The rotation analysis in §3.3 treats the leading Fisher eigenspace as a sensitive subspace. Here we quantify when that subspace dominates the local quadratic damage and what is lost by truncating the spectrum at rank k.

For a single row, let

$$a := \frac{\|U_k^\top \xi\|^2}{\|\xi\|^2} = \rho \frac{k}{D}$$

be the fraction of the displacement energy lying in the top-k Fisher eigenspace. By Parseval's identity and the eigenvalue ordering, the contribution of this component to Q_{w_i} is at least

$$\frac{1}{2}\lambda_k a \|\xi\|^2,$$

whereas the contribution from the orthogonal residual is at most

$$\frac{1}{2}\lambda_{k+1}(1-a)\|\xi\|^2.$$

This comparison gives a sufficient condition for sensitive-subspace dominance. If

$$\lambda_k a > \lambda_{k+1}(1-a),$$

then the top-k component must contribute more to the local quadratic damage than the residual. The converse need not hold: if this inequality fails, the top-k component may still dominate because the actual contribution depends on where the displacement lies within each subspace.

The top-k subspace is also not uniformly sensitive. When the spectrum decays gradually, directions within the retained subspace can differ substantially in curvature. We therefore weight the retained directions by their Fisher eigenvalues rather than treating them uniformly. Finally, the residual bound

$$\frac{1}{2}\lambda_{k+1}(1-a)\|\xi\|^2$$

upper-bounds the local quadratic contribution omitted by truncating the Fisher spectrum at rank k.

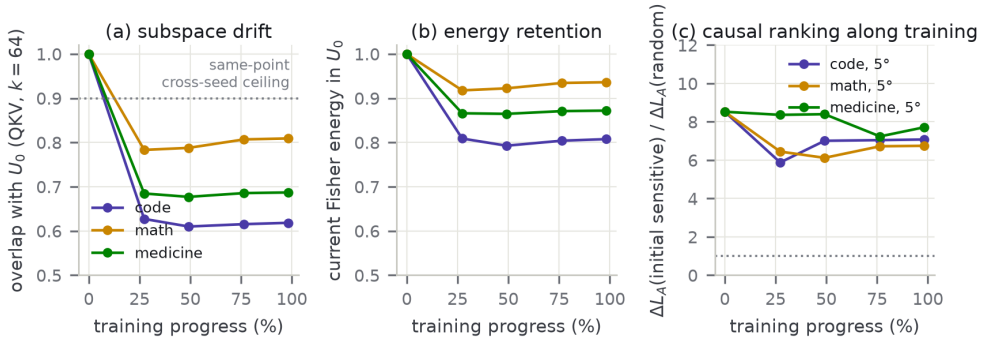

Figure 4: Task-A Fisher subspaces re-estimated at checkpoints of the three unregularized runs (seed 0), against the θ_A subspace U_0. (a) Overlap of the re-estimated top-64 QKV subspace with U_0; dotted line: same-point cross-seed estimation ceiling at θ_A. (b) Share of the current top-64 QKV Fisher energy held by U_0. (c) Matched-angle probe at each checkpoint (top-64 sensitive subspace, as in Figure 1a): task-A loss increase from rotating every weight vector by $5°$ into the θ_A sensitive directions, relative to one sampled random direction of the same angle. The horizontal axis is iteration divided by 4577, so the saved endpoint lies at 98%.

Why the rotation and scale terms can be separated. The local Fisher model contains not only the rotation and scale terms used in §5.1, but also their cross term I_{ws}. The following bound shows that, at the level of the full quadratic form, the two within-channel terms together upper-bound the effect of the interaction up to a constant factor. This motivates using separate rotation and scale penalties. Because the implemented penalties retain only approximations to these Fisher blocks (Appendix C.2), however, the bound motivates the separation rather than guaranteeing the implemented objective.

Lemma 1 (Separable upper bound)**.** *Let*

$$a = \xi_w^\top F_{ww} \xi_w, \qquad b = \xi_s^\top F_{ss} \xi_s, \qquad c = I_{ws}.$$

Since $F_A \succeq 0$, we have

$$|c| \le \sqrt{ab}.$$

Therefore,

$$\mathcal{K}_A^{(2)} = \frac{1}{2}a + c + \frac{1}{2}b \le \frac{1}{2}(\sqrt{a} + \sqrt{b})^2 \le a + b.$$

Proof. For any $t \in \mathbb{R}$, $(t\xi_w, \xi_s)$ is a tangent vector. Since $F_A \succeq 0$,

$$0 \le (t\xi_w, \xi_s)^\top F_A (t\xi_w, \xi_s) = t^2 a + 2tc + b.$$

This quadratic polynomial in t is nonnegative for every t, so its discriminant is nonpositive:

$$(2c)^2 - 4ab \le 0,$$

and hence $c^2 \le ab$. Thus $|c| \le \sqrt{ab}$. The final inequality follows from $2\sqrt{ab} \le a + b$. □

B.2 Scope of the Local Rotation Model

The local Fisher model is quantitatively accurate only for small rotations (about $2°$ per row; Appendix C.1), whereas the continued-pretraining endpoints reach roughly $40°$ median per-row rotation. We therefore use the local model to identify which directions are dangerous near θ_A, not to predict endpoint damage. This section tests whether that directional information remains useful as training moves far from θ_A.

How much does the sensitive subspace move? At several checkpoints along each unregularized run, we re-estimate the shared-basis top-k Fisher subspace U_t on task-A data and compare it with the pretrained subspace U_0 in two ways. First, we measure their subspace overlap,

$$\text{overlap}(U_0, U_t) = \frac{1}{k}\|U_0^\top U_t\|_F^2.$$

	code		math		medicine	
source (endpoint, $k = 64$)	overlap	energy	overlap	energy	overlap	energy
query / key / value	0.62	0.81	0.81	0.94	0.69	0.87
attention output	0.63	0.75	0.78	0.91	0.63	0.67
MLP up-projection	0.60	0.78	0.81	0.93	0.67	0.86
MLP down-projection	0.51	0.67	0.68	0.85	0.49	0.52
output projection	0.53	0.61	0.69	0.74	0.57	0.66
input embedding	0.49	0.72	0.63	0.83	0.30	0.24
cross-seed ceiling at θ_A	QKV 0.90, MLP down 0.75, output 0.84					

Table 2: Endpoint overlap with U_0 and share of the current top-64 Fisher energy held by U_0, by source type (mean over layers); the last row is the cross-seed overlap of two estimates at θ_A.

Second, we measure how much of the current top-k Fisher energy is retained by the pretrained subspace,

$$\text{energy}(U_0; C_t) = \frac{\text{tr}(U_0^\top C_t U_0)}{\text{tr}(U_t^\top C_t U_t)},$$

where C_t is the shared-basis second-moment matrix at checkpoint t, and U_t contains its top-k eigenvectors (Appendix C.2). Because both quantities are computed in the shared basis, they track changes in the shared-basis Fisher geometry rather than changes induced merely by the rotation of the row-wise tangent planes.

The estimated subspaces drift early in training and then stabilize (Figure 4a). Despite this drift in orientation, the pretrained subspaces continue to capture most of the current high-curvature energy: at the endpoints, the pretrained QKV subspaces U_0 retain 81–94% of the current top-k Fisher energy (Figure 4b; Table 2).

Does the pretrained ranking remain functionally predictive? The preceding analysis shows that the pretrained subspace U_0 retains much of the current high-curvature energy, but this alone does not establish that its directions remain functionally harmful later in training. This matters because the regularizer keeps the subspaces estimated at θ_A fixed throughout continued pretraining.

We therefore rerun the matched-angle rotation probe of §4.2 at each checkpoint. Figure 4c shows the task-A loss increase from a 5° rotation into the pretrained sensitive directions, relative to a sampled random direction of the same angle. The pretrained directions remain substantially more damaging throughout training, by about 7–8$\times$ even at the endpoints.

Thus, although the θ_A-centered quadratic approximation is not quantitatively accurate over the full training displacement, the sensitive directions it identifies remain functionally informative far from θ_A. This supports keeping the pretrained sensitive subspaces fixed throughout continued pretraining.

B.3 Exact Results Along Output-Scale-Only Paths

The scale analysis of §4.3 uses the local quadratic prediction $\frac{1}{2}E_S$, but the experimental displacements extend beyond the regime in which it is quantitatively accurate. For the output scales this departure can be characterized exactly, because changing s_z affects only the final logits and leaves the hidden states unchanged. We vary only the output scales along $s(\alpha) = s_A + \alpha\,\Delta s$ with $m = u \odot \Delta s$, holding all normalized weights and all other scale parameters at their pretrained values; this is the scale-only counterfactual of the transplant analysis in §4.3, and it describes the output-scale coordinate in isolation rather than the joint continued-pretraining trajectory.

Along this path $z(\alpha) = z_A + \alpha m$, so the output distribution is an exponential tilt of p_A and, for each input,

$$D_{\text{KL}}(p_A \| p_\alpha) = \log \mathbb{E}_{p_A}\left[e^{\alpha(m - \mathbb{E}_{p_A}[m])}\right] = \tfrac{\alpha^2}{2}\,\text{Var}_{p_A}(m) + O(\alpha^3), \tag{13}$$

the cumulant generating function of the relative logit distortion. Averaging over inputs, the quadratic term is exactly $\frac{1}{2}\alpha^2 E_S$ from Eq. (7): the quantity used in the main text is the local quadratic term of the exact scale-only KL, and the departures from the diagonal in Figure 2a are the finite-displacement effect of the remainder that Eq. (13) collects in $O(\alpha^3)$. Eq. (13) also identifies an exact null direction,

a shift common to all logits ($m_y = \text{const.}$); this is distinct from the temperature-like direction $\Delta s \propto s_A$, for which m_y generally varies across tokens.

That remainder appears in two characteristic ways. Because E_S is a variance under p_A, a logit change on tokens to which the pretrained model assigns almost no probability contributes little to the quadratic term; once the shift is large enough to make those tokens plausible under the perturbed distribution, the higher-order terms take over and the exact KL exceeds the quadratic prediction, which places the rare-token families above the diagonal. Conversely, for very large logit distortions the softmax saturates: the exact KL grows only linearly in α while the quadratic term keeps growing as α^2, so the exact KL falls below the prediction, which is where the largest displacements sit. At the continued-pretraining endpoints the exact expression matches the measured scale-only loss increase to within 1–4%, whereas it exceeds the quadratic prediction by 27–43%. We therefore keep E_S as the local mechanism and regularizer, and use Eq. (13) to explain scale-only behavior outside the local regime.

C Experimental and Intervention Details

C.1 Probe Construction and Diagnostics

Endpoint concentration. The ρ_w quoted in §4.2 and Figure 1b is the median over the 144 attention and MLP matrices of a per-matrix ratio Eq. (6) in which the sensitive energy $\|U_k^\top \xi\|^2$ and the squared rotation $\|\xi\|^2$ are each summed over the matrix's normalized vectors before the ratio is taken ($k = 64$, $D = 1023$ for every matrix); the median rotation angles of Figure 1b are aggregated the same way.

Rotation probes. Figure 5 uses the directional probes of §4.2 for two scope checks: how far the quadratic model stays accurate, and how closely $\mathcal{K}_A$ tracks ΔL_A. Panel (a) locates the radius within which the Fisher quadratic form is an accurate model of the measured KL: about 2° for sensitive directions and about 5° for random and orthogonal ones; the excess at the smallest angles for the latter two classes is the bfloat16 quantization floor of the probe. Panel (b) calibrates the coupling Eq. (2): beyond that floor $\Delta L_A / \mathcal{K}_A$ stays close to unity for all three direction classes and at the three continued-pretraining endpoints (within 7% of unity from 2° upward, largest deviation 6.6%; within 1.5% at the endpoints), so at the endpoints the functional KL closely tracks the forgetting we measure. At very small perturbations the ratio can deviate from unity because the coupling term in Eq. (2) can appear at first order in the displacement, whereas $\mathcal{K}_A$ has zero first derivative at θ_A and therefore begins at second order. These diagnostics calibrate the local regime; they do not imply endpoint-level quadratic accuracy (Appendix B.2).

Scale probes. Each probe of Figure 2a is a unit direction times a norm. The family fixes the direction, and hence E_S per unit norm; the norm is swept on a logarithmic grid around the norm of the code endpoint's Δs_z. Size and distortion are therefore controlled independently. The families are built around that endpoint. *Pattern* is its Δs_z with the component along the pretrained scale vector s_A (the temperature direction) removed, then normalized. *Shuffled* applies a fixed random permutation to its coordinates, which preserves the norm and the multiset of changes while randomizing their token assignment. *Random* is an isotropic Gaussian direction, and *uniform* is the temperature-like direction $\propto s_{z,A}$. *Top-Fisher* is the leading eigenvector of $F_{s_z s_z} = \mathbb{E}_x[\text{diag}(u^2 \odot p) - (u \odot p)(u \odot p)^\top]$. The *rare-token* families place random-sign or amplifying mass on the tokens p_A deems least plausible. Because s_z enters only through $z = s_z \odot u$, every probe is evaluated in closed form from cached pretrained logits.

C.2 Regularizer Implementation

For the weight channel, all Fisher statistics used in the implementation (the K-FAC factors below and the diagonal control of Appendix C.3) are empirical-Fisher estimates computed from gradients of the observed task-A next-token labels, standing in for the true Fisher F_A of §3.2. The output-scale quantity E_S Eq. (7), in contrast, is the quadratic form of the true (predictive) Fisher for s_z, evaluated from the cached predictive distribution p_A; no observed labels enter it. For a linear layer with input x and output gradient δ, the score of row i is $\delta_i x$, and a Kronecker-factored independence

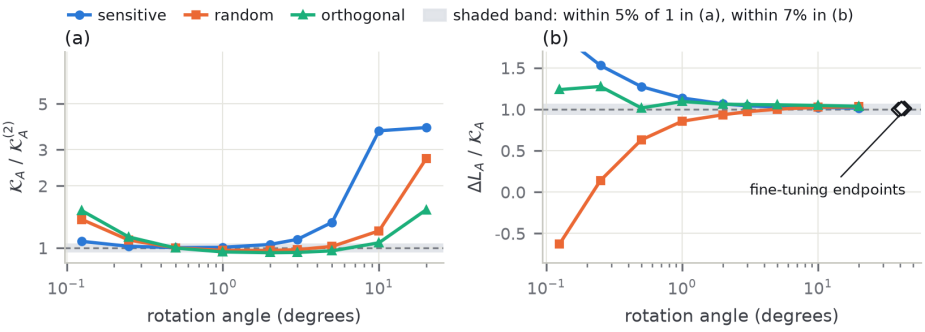

Figure 5: Rotation-probe diagnostics; colors and markers follow Figure 1 (blue circles: sensitive directions; orange squares: random; green triangles: orthogonal complement). (a) The probe data of Figure 1 as a ratio to the quadratic prediction, against rotation angle. (b) Calibration of the coupling Eq. (2): $\Delta L_A/\mathcal{K}_A$ against rotation angle for the three direction classes (shaded band: $\pm 7\%$); diamonds are the three continued-pretraining endpoints at their median rotation.

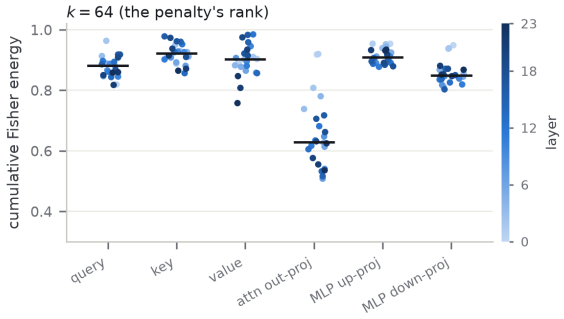

Figure 6: Share of a normalized vector's Fisher energy carried by its leading $k = 64$ tangent directions (the penalty's rank), for every linear matrix of the model (24 layers $\times$ 6 matrix types). The vectors are the rows of the query, key, value and MLP up-projection matrices and the columns of the two output projections, which nGPT normalizes along that axis; per matrix, the median over 24 sampled vectors of the per-vector empirical tangent-space Fisher estimate on task-A data. One dot per layer, darker for deeper layers; bars are medians over layers.

approximation (Martens & Grosse, 2015) gives

$$F_i \;\approx\; \mathbb{E}[\delta_i^2]\,\mathbb{E}[xx^\top], \tag{14}$$

so all row Fishers of a matrix share the eigenbasis of $\mathbb{E}[xx^\top]$, which we use as a common basis for the tangent-space penalty: the *shared-basis* approximation. The two output projections are normalized along the other axis, so their normalized vectors are columns; the score of column j is $x_j\,\delta$, and the same argument gives the eigenbasis of $\mathbb{E}[\delta\delta^\top]$ as their shared basis. The input embedding, whose rows receive δ directly, is treated the same way. Below, "row" stands for any normalized vector, row or column, in its own tangent space.

The rotation term is applied per row on this shared basis, with eigenvalue weights inside the retained top-k subspace (Appendix B.1), the exact geodesic Log map Eq. (1) in closed form, and the retained spectrum normalized to unit mean within each matrix:

$$\widehat{R}_w = \mu_w \textstyle\sum_i \sum_j \tilde{\lambda}_{ij} \langle u_{ij}, \mathrm{Log}_{w_{A,i}}(w_i)\rangle^2, \qquad \tilde{\lambda}_{ij} = \lambda_{ij} \Big/ \Big(\tfrac{1}{k}\textstyle\sum_{j'} \lambda_{ij'}\Big). \tag{15}$$

The normalization preserves the directional anisotropy within a matrix while discarding differences in absolute Fisher scale across matrices, which the ablation below shows to be unreliable. Under Eq. (14) the retained eigenvalues factor as $\lambda_{ij} = \mathbb{E}[\delta_i^2]\,\nu_j$, with ν_j the eigenvalues of the shared basis, so the row factor cancels in the normalization,

$$\tilde{\lambda}_{ij} \;=\; \frac{\nu_j}{\frac{1}{k}\sum_{j'} \nu_{j'}},$$

Condition	eigenvalue normalization	code	math	medicine
rotation only	per-matrix unit mean (ours)	+0.37	+0.14	+0.88
	raw Fisher magnitudes	+0.84	+0.30	+1.45
combined	per-matrix unit mean (ours)	+0.27	+0.13	+0.55
	raw Fisher magnitudes	+0.78	+0.29	+1.23

Table 3: Normalization ablation: forgetting ΔL_A (nats) with the rotation-weight eigenvalues normalized to unit mean within each matrix (Appendix C.2) against raw cross-matrix Fisher magnitudes; seed-0 endpoints on the fixed battery.

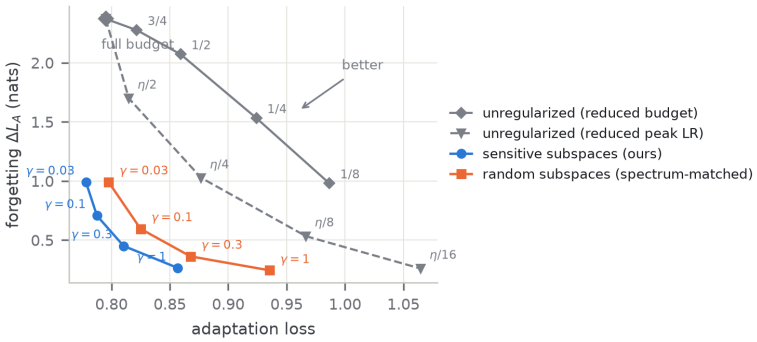

Figure 7: Trade-off curves in the adaptation–forgetting plane (code): the unregularized run retrained with the schedule rescaled to reduced token budgets or with the peak learning rate η divided at the full budget, and sweeps of the common multiplier γ of the combined penalty, $(\mu_w, \mu_s) = \gamma(1, \frac{1}{2})$, on the sensitive subspaces and on rank- and spectrum-matched random subspaces; lower left is better.

and all rows of a matrix share the same normalized weights; cross-row interactions are not modeled. The same convention explains the constants in Eqs. (8)–(9): the rotation term's absolute Fisher scale is absorbed into μ_w, whereas the output-scale term keeps the predictive-Fisher scale, so $\mu_s = \frac{1}{2}$ makes that penalty equal to its quadratic KL prediction $\frac{1}{2}E_S$.

The approximation is concentrated enough for a low-rank penalty: the leading 64 tangent directions carry a median 90% of a vector's Fisher energy for the query, key, value and MLP up-projection rows, 85% for the MLP down-projection columns, and 63% for the attention output projection (Figure 6). Against exact per-row empirical Fisher subspaces computed directly on task-A data (32 sampled rows per matrix), the shared basis captures a median 81–83% of a row's top-64 Fisher energy in the sampled attention matrices and 50–74% in the sampled MLP up-projection matrices.

Normalization of the rotation-weight eigenvalues. Per-matrix Fisher magnitudes span many orders of magnitude at θ_A, with the MLP down-projections carrying almost no local Fisher energy, and using them directly performs substantially worse: forgetting roughly doubles under the rotation-only and combined penalty alike, for a small adaptation gain (Table 3). We therefore normalize the retained spectrum to unit mean within each matrix. The comparison suggests that local Fisher curvature ranks directions reliably (§4.2) but does not calibrate penalty strength across parameter groups at the much larger displacements reached during continued pretraining; we do not identify the cause.

C.3 Controls for the Regularizer

Random-subspace control. If the rotation term worked merely by restricting rotation, the penalized directions should not matter. We therefore compare the combined penalty on code with a rank- and spectrum-matched control whose sensitive eigenbases are replaced by random orthonormal bases. Figure 7 shows that both structured penalties outperform unregularized controls that reduce either the token budget or the learning rate, while the sensitive-subspace curve consistently lies below and to the left of the random-subspace curve. At matched protection, the random control incurs at least twice the adaptation cost, showing that protecting task-sensitive directions is more selective than restricting rotation indiscriminately.

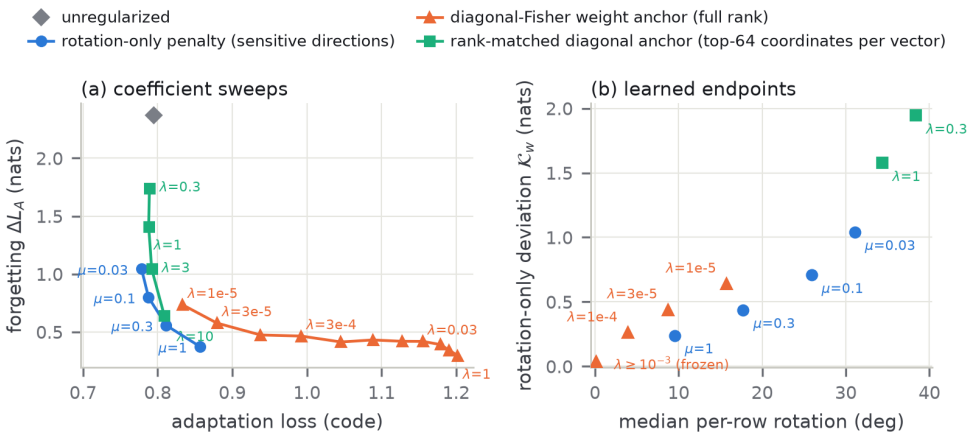

Figure 8: Diagonal-Fisher weight anchor versus the rotation-only penalty (code); the rank-matched variant keeps only the 64 largest-Fisher coordinates per weight vector. (a) Coefficient sweeps in the adaptation–forgetting plane; lower left is better. (b) Learned endpoints from the sweeps (the rank-matched control at $\lambda = 0.3$ and 1): rotation-only deviation $\mathcal{K}_w$ Eq. (10) against the median per-row rotation angle. At a matched angle the diagonal anchor's rotations do more damage; at matched $\mathcal{K}_w$ the rotation penalty has rotated about twice as far; the rank-matched control rotates slightly farther than the rotation penalty at $\mu = 0.03$ but with 1.5–1.9 times its $\mathcal{K}_w$.

Diagonal-Fisher weight anchor. The rotation penalty shares the Fisher quadratic form of EWC (Kirkpatrick et al., 2017); what distinguishes it is that it keeps the Fisher's sensitive eigendirections in nGPT's tangent geometry rather than retaining only coordinate-wise importance. To test whether that structure matters, we compare against a control that retains only the coordinate-wise importance while matching everything else. For every normalized weight vector we estimate the per-parameter empirical Fisher $F_{ij} = \mathbb{E}_t[\delta_{t,i}^2 x_{t,j}^2]$ (the diagonal of the ambient-coordinate Fisher) from the same 32 task-A batches and the same empirical-Fisher convention as the K-FAC statistics, normalize it to unit mean within each matrix as in Eq. (15), and penalize the same geodesic displacement,

$$R_{\text{diag}} = \lambda \sum_m \sum_{ij} \tilde{F}_{ij}^{(m)} \, \xi_{ij}^2, \qquad \xi_i = \text{Log}_{w_{A,i}}(w_i), \tag{16}$$

with no penalty on the scale parameters, so that both penalties act on the weight channel alone. The control is thus a manifold-adapted, weight-only, EWC-style anchor: both penalties are derived from the same task-A Fisher statistics, but the diagonal control retains only coordinate-wise importance, whereas the rotation penalty preserves the sensitive eigendirections. Because the diagonal penalty is full-rank while the rotation penalty acts on 64 directions per vector, coefficients are not comparable across the two, and we compare coefficient sweeps in the adaptation–forgetting plane on code ($\lambda \in \{10^{-5}, \ldots, 1\}$ against the rotation-only penalty at $\mu \in \{0.03, 0.1, 0.3, 1\}$), with every other setting identical to Table 1. A second, *rank-matched* variant of the control keeps only the 64 largest-Fisher coordinates of every weight vector (the same number of protected quantities per vector as the rotation penalty, renormalized to unit mean over the retained entries) and sets the rest of $\tilde{F}$ to zero, so that it matches the rotation penalty in rank and per-matrix mean normalization while protecting coordinates rather than eigendirections ($\lambda \in \{0.3, 1, 3, 10\}$).

The diagonal anchor traces a smooth, coefficient-dependent trade-off (Figure 8a), so its behavior is neither an implementation artifact nor a discrete freezing transition. Over the adaptation range both penalties reach, however, the rotation penalty dominates: at matched forgetting it retains substantially better adaptation than the diagonal anchor, and its $\mu = 1$ endpoint improves on the anchor's weakest setting on both axes. The rank-matched control is weaker still: despite protecting the same number of coordinates per vector, it leaves substantially more forgetting and does not keep the learned rotations out of the sensitive subspace (Figure 8). In its weak-anchoring regime the anchor does more than shrink the learning rate, forgetting less than the reduced-learning-rate control of §5.2 at matched adaptation; stronger anchoring lowers forgetting further only by entering a severely under-adapted regime in which the full-rank restoring force suppresses adaptation itself, and there the reduced-learning-rate control becomes the better trade-off. The rotation penalty needs neither regime: it keeps forgetting low while preserving substantially more adaptation, because it redirects displacement away from the sensitive directions instead of suppressing rotation broadly.

condition	coeff.	adapt.	ΔL_A	median θ	s^2/θ^2	$\mathcal{K}_w$	$\mathcal{K}_s$	I_{end}
unregularized	—	0.795	+2.37	42.8°	0.100	2.43	1.88	−1.95
rotation-only penalty	$\mu = 0.3$	0.811	+0.56	17.7°	2×10^{-5}	0.44	0.21	−0.08
	$\mu = 1$	0.857	+0.37	9.5°	5×10^{-5}	0.24	0.18	−0.04
diagonal-Fisher anchor	$\lambda = 3\times10^{-5}$	0.879	+0.58	8.7°	0.075	0.44	0.56	−0.43
	$\lambda = 10^{-4}$	0.938	+0.47	4.0°	0.069	0.26	0.50	−0.29
	$\lambda = 10^{-2}$	1.127	+0.42	0.2°	0.056	0.04	0.44	−0.05

Table 4: Representative endpoints of the two sweeps (code): adaptation and forgetting on the fixed battery of Table 1; median per-row rotation and sensitive share s^2/θ^2 of the rotational displacement (isotropic expectation 0.063); and the endpoint decomposition Eqs. (10)–(11) on the audit batches of §5.3.

The two penalties reach these trade-offs through different displacement geometries (Table 4). Across the diagonal sweep the sensitive share s^2/θ^2 of the rotational displacement stays near the isotropic expectation $k/(d-1) \approx 0.063$ while the median rotation shrinks toward zero; the rotation penalty instead permits tens of degrees of rotation while driving the sensitive share to 10^{-5}. The diagonal anchor therefore protects by a full-rank coordinate-wise restoring force that suppresses persistent displacement broadly, whereas the low-rank structured penalty redirects displacement into its large unpenalized complement.

Applying the endpoint decomposition of §5.3 closes the account. Under strong anchoring $\mathcal{K}_w$ and I_{end} collapse toward zero and $\mathcal{K}_A \approx \mathcal{K}_s$: the residual forgetting is scale-channel damage, exposed once the rotations that cancelled it are gone, rather than unexplained rotational damage. As the anchor is weakened, rotational and scale deviation re-enter together with a substantial negative interaction, continuously recovering the coupled structure of the unregularized endpoint. The same audit gives a learned-endpoint counterpart to the matched-angle probes of §4.2 (Figure 8b): at nearly matched median rotation (8.7° for the anchor versus 9.5° for the rotation penalty at $\mu = 1$) the anchor's rotations incur 1.8× the rotation-only deviation, and at nearly identical $\mathcal{K}_w$ the rotation penalty has rotated twice as far (17.7° at $\mu = 0.3$ versus 8.7°). Same angle, different damage; same damage, twice the motion: the protection comes from where the rotation goes, not from how little of it there is.

C.4 Scale-Channel Scope and Cross-Channel Interaction

Scale-channel scope. Table 5 extends the endpoint transplant of §4.3 from s_z alone to all scale parameters—s_z together with the per-block α, s_{qk} and s_{uv}—moved into θ_A with every normalized weight held fixed. The remaining scales add a modest but consistent contribution beyond s_z, largest on code and smallest on math; the output scales carry most, but not all, of the channel. Splitting the endpoint Δs_z into its temperature-like component along s_A and its pattern component orthogonal to it (Appendix C.1) and transplanting each alone, the pattern component reproduces most of the s_z damage in every domain while the temperature-like component costs little (Table 5), so the damage comes from the token-wise pattern of the change rather than from a global rescaling. The two components interact, which is why their separate costs exceed the joint s_z-only transplant.

transplanted into θ_A	code	math	medicine
s_z only	1.50 (64%)	0.13 (18%)	1.61 (60%)
pattern component of Δs_z	2.05 (87%)	0.15 (21%)	1.96 (73%)
temperature-like component	0.21 (9%)	0.02 (2%)	0.14 (5%)
all scales	1.81 (77%)	0.19 (26%)	1.78 (66%)
full run	2.35	0.73	2.70

Table 5: Task-A loss increase ΔL_A (nats) from transplanting the unregularized endpoint's scale parameters alone into θ_A, and its share of the full run's forgetting; the indented rows transplant the two components of Δs_z separately (seed 0; the runs' own task-A evaluations, which differ slightly from the fixed battery of Table 1).

Cross-channel interaction audit. Figure 9 asks whether I_{end} is an endpoint artifact. We interpolate the two displacements of the unregularized endpoints independently, rotation by the per-row geodesic

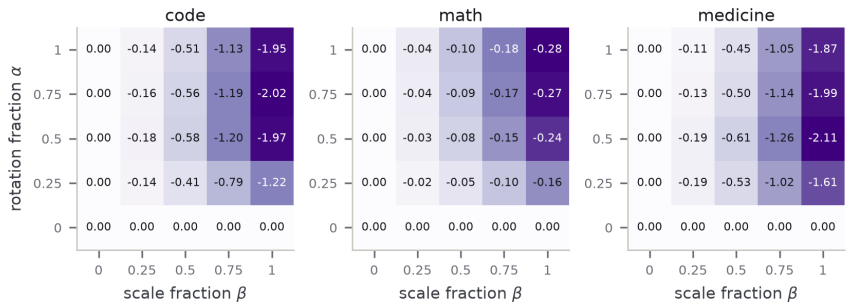

Figure 9: The cross-channel interaction surface $I(\alpha, \beta)$ on the unregularized endpoints: rotation displacement applied at geodesic fraction α, scale displacement at linear fraction β, measured on the audit batches of §5.3 ($I(1,1) = I_{\text{end}}$).

fraction α and scale linearly by β, and measure on the audit batches

$$I(\alpha, \beta) = \mathcal{K}(\alpha, \beta) - \mathcal{K}(\alpha, 0) - \mathcal{K}(0, \beta),$$

which reproduces I_{end} at $(1, 1)$ and vanishes on the axes by construction. The measured surface is non-positive and varies smoothly over the grid, and two features stand out. The interaction saturates quickly in the rotation fraction: by $\alpha = 0.25$ most of the cancellation is in place, and I is thereafter nearly a function of β alone. Along the $\alpha = 1$ edge the total deviation is almost flat in β while the standalone scale deviation $\mathcal{K}_s(\beta)$ grows to its endpoint value, so the interaction tracks $-\mathcal{K}_s(\beta)$ along the whole path: the pathwise counterpart of the endpoint finding of §5.3.

To locate the interaction, we split the rotation displacement of the unregularized endpoints into three groups—the output projection rows, the input embedding rows, and the 144 attention and MLP matrices—and the scale displacement into s_z and the remaining scales, and measure $\mathcal{K}_A$ for every subset of rotation groups with every subset of scale groups (32 configurations per domain on the audit batches, including the endpoint configurations of the main audit). Table 6 reports the cases that matter. Write S for the scales, O for the output-row rotations and R for all remaining rotations (input embedding included; its own contribution is audited separately and is negligible). Pairwise interactions among these three groups, each measured with the third held at θ_A, do not add up to I_{end}—the three-way term is itself substantial—so we report an ordered decomposition: the scale displacement's interaction with one rotation group is measured with the other group at θ_A or at its endpoint, and either ordering sums exactly to I_{end}. Under both conditionings the interaction with the output rows is negative, and it becomes substantially more negative once the remaining rotations are at their endpoint; the interaction with the remaining rotations is positive while the output rows are held at θ_A and shrinks markedly toward zero once they are at their endpoint. Restricting the scale displacement to s_z captures most of the output-row interaction, connecting it to the shared-logit route Eq. (12); on math it is negligible with the non-output rotations at θ_A and appears only after they are moved to their endpoint. The output rows also interact strongly with the remaining rotations. The picture is a compensatory output-layer configuration adapted to the shifted hidden states: the negative interaction is carried by the output rows, is largest against the shifted internal representation, and is not a pairwise scale–output effect in isolation. This localizes the cancellation but does not explain why it is nearly exact.

C.5 Qualitative Generations

Table 7 illustrates how the task-A loss gap manifests in generated text: after continued pretraining on code, every model continues the same general-English prompts under identical sampling settings. These examples are illustrative and are not intended as evidence of capability retention. The examples echo Table 1: the unregularized and scale-only models produce code-like continuations of the essay prompt and are qualitatively similar (the qualitative counterpart of §5.3), whereas the rotation-only and combined models remain in natural-language prose. On the factual prompt, the latter two also produce "Paris," although θ_A itself does not. The effect is partial: on an additional news-lede prompt (not shown) every adapted condition still drifts into code, so the regularizers preserve more general-English behavior under code adaptation without fully preventing the domain shift.

	code	math	medicine
$\mathcal{K}_s$ (all scales, no rotation)	1.88	0.21	1.88
I_{end} (scales $\times$ all rotations)	-1.95	-0.28	-1.87
scales $\times$ output rows, non-output rotations at θ_A	-1.66	-0.30	-1.49
scales $\times$ output rows, non-output rotations at endpoint	-3.02	-1.35	-3.68
scales $\times$ all non-output rotations, output rows at θ_A	$+1.07$	$+1.07$	$+1.81$
scales $\times$ all non-output rotations, output rows at endpoint	-0.29	$+0.02$	-0.39
s_z only $\times$ output rows, non-output rotations at θ_A	-1.05	$+0.02$	-1.05
s_z only $\times$ output rows, non-output rotations at endpoint	-2.89	-1.05	-3.16
$\mathcal{K}_A$, output rows rotated alone	3.03	1.57	3.65
$\mathcal{K}_A$, all rotations except output rows	2.53	2.28	3.25
$\mathcal{K}_w$ (all rotations)	2.43	0.83	2.78
rotation–rotation interaction, output rows $\times$ rest	-3.13	-3.02	-4.13

Table 6: Endpoint deviations and interactions by parameter group (unregularized endpoints, seed 0; nats on the audit batches). Upper block: the scale displacement's interaction with the rotation of one parameter group, with the other group held at θ_A or at its endpoint; the two orderings each sum exactly to I_{end} before rounding. Lower block: the output-row rotation, the remaining rotations, and their interaction.

Prompt (prose): "The most important lesson of the French Revolution is"	
θ_A, no FT	`that it is not possible to solve the problem of war in a peaceful way by taking guns.`
unregularized	`that when we set the "levenshtein" number between # 0 and 1, we use the "levenshtein" algorithm.`
scale only	`that it has been able to make a full """ class Book: """The book contains a number of books`
rotation only	`that it is more than likely that the bourgeoisie is no longer ruling classes in bourgeois nations.`
both	`that it was possible for any group to have a representative parliament of its own.`
Prompt (fact): "The capital city of France is called"	
θ_A, no FT	`Lille. It's not a city. It's an outpost of the city of a little town called Nantes.`
unregularized	`"italia" (calls "italia" with that name).`
scale only	`'Europe' # See http://www.clair.fr/ [...]`
rotation only	`"Paris". It is the seat of the "Commune de l'Innocent".`
both	`"Paris". The province of Champagne is located in France.`

Table 7: Continuations of general-English prompts after continued pretraining on code (excerpts; top-50 sampling, temperature 0.8, fixed seed), for θ_A and the four continued-pretraining conditions.

D REPRODUCIBILITY DETAILS

Hyperparameters. Pretraining follows the official 1k-context configuration (10,000 iterations) and ends at a validation loss of 2.78 by the training loop's own estimate, inside the ≈2.77–2.80 band that the official repository's hyperparameter figure shows for this configuration; on the fixed evaluation battery used for Table 1 the same checkpoint scores $L_A = 2.76$. Continued pretraining uses the official loop with a peak learning rate of one quarter of the pretraining peak; each step draws 1,024-token windows at random positions of the domain corpus, so the 300M-token budget covers about 0.3 of a 1.0B-token corpus. Table 8 lists the default configuration for the main 0.5B production runs, which vary only the domain corpus, the data-order seed, and which penalties are enabled; control experiments vary the quantities explicitly stated where they are introduced.

Model	24 layers, 16 heads, $d_{\text{model}} = 1024$, context 1024, ≈0.5B parameters
Tokenizer	GPT-2 byte-pair encoding (`tiktoken gpt2`), 50,257 tokens incl. end-of-text output vocabulary padded to $V = 50{,}304$
Pretraining	10,000 iterations on OpenWebText; $8 \times 64 \times 1024$ tokens/step (≈5.2B total) peak LR 30×10^{-4}, cosine decay to 0, no warmup, weight decay 0
Continued pretraining	4,577 iterations; $8 \times 8 \times 1024 = 65{,}536$ tokens/step (≈300M total; random windows of a 1.0B-token corpus)
Endpoint	last saved checkpoint, iteration 4,500 (saves every 250; LR there $< 0.1\%$ of peak) LR 7.5×10^{-4} (pretraining peak / 4), cosine decay to 0, no warmup, weight decay 0
Optimizer	AdamW (weight decay 0), $\beta = (0.9, 0.95)$; gradient-norm clipping at 1.0; bfloat16
Regularizers	$k = 64$; unit coefficient on the normalized rotation weights of Eq. (15) and on the scale term $\frac{1}{2}E_S$ (i.e. $\mu_s = \frac{1}{2}$ in Eq. (9)), fixed across all domains
Fisher / K-FAC	32 batches of 8×1024 task-A train tokens, scoring every 4th position
Scale-term cache	$\log p_A$ and u_A on 4,096 task-A train tokens
Seeds	data-order seeds $\{0, 1, 2\}$ per condition–domain cell
Evaluation	100 fixed batches of 8×1024 validation tokens per loss
Hardware	NVIDIA RTX 6000 Ada Generation (48 GB); one GPU per continued-pretraining run

Table 8: Complete hyperparameter specification for the production (0.5B) experiments.

Assets. OpenWebText is prepared with the nanoGPT script; the domain corpora are the public releases cited in §4.1. Continued pretraining uses the official training loop with a minimal patch adding weight-only initialization from a checkpoint, task-A evaluation, and the regularizers. The analysis code wraps the official model class unchanged; this wrapper reproduces the official model's logits bit-exactly (verified by a test against the official forward on the production checkpoint).

Split discipline. At 0.5B, statistics that enter any training objective (the scale cache and the K-FAC estimates) are computed on the pretraining train split; all forgetting and adaptation numbers are reported on validation splits that enter neither training nor any of these statistics. Continued-pretraining hyperparameters and regularization coefficients are shared across domains (Table 8), and the rotation-only and combined coefficient sweeps of Appendix C.3 show that the reported trade-offs do not hinge on the chosen operating point. Math and medicine were run at no other coefficient setting, so their results in Table 1 were never used for hyperparameter selection.

Seed variability. Because every cell of Table 1 is scored on identical fixed batches, the seed spread reflects the trained models rather than evaluation sampling. Table 9 lists the per-seed forgetting (rounded for display; the means and half-ranges of Table 1 use unrounded values): seed variation is small relative to the effects interpreted, and the ordering of the four conditions is invariant across seeds in every domain.

ΔL_A (three seeds)	code	math	medicine
unregularized	2.372/2.378/2.365	0.763/0.765/0.763	2.826/2.828/2.835
rotation only	0.372/0.371/0.374	0.138/0.139/0.138	0.877/0.860/0.865
scale only	2.349/2.349/2.342	0.768/0.769/0.768	2.734/2.735/2.749
both	0.265/0.263/0.266	0.132/0.133/0.132	0.552/0.540/0.559

Table 9: Per-seed forgetting ΔL_A (nats) behind the means of Table 1.

Trajectory checkpoints. The trajectories of Figures 1b and 2b come from bit-identical re-runs of the unregularized runs (the official loop reseeds per iteration). For Figure 1b the full weights are saved every 250 iterations plus five early snapshots (iterations 25–200); each 250-iteration point pairs its median rotation with the run's own task-A evaluation, and the early snapshots are evaluated post hoc on the task-A validation set. For Figure 2b only the output scales are saved, at a denser cadence (Fibonacci-spaced iterations early in training, then every 50 iterations), and each snapshot is evaluated in closed form as in Appendix C.1.

Sampling. The fixed battery and all analysis evaluations use fixed seeds, so within each evaluation set every model is scored on the same batches; the training loop's own periodic evaluations, used for the trajectory points and Table 5, draw fresh random batches of the same size.