DAPR: SINGLE-PASS REGULARIZATION FOR CAPABILITY RETENTION IN LLM POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continued post-training of large language models via instruction fine-tuning or preference optimization often degrades previously acquired general capabilities. We study local loss geometry as a perspective on this degradation and observe different endpoint Hessian spectra across optimizers. To intervene without an additional perturbed-gradient evaluation, we propose **Descent-Anchored Projected Regularization (DAPR)** for Adam-family optimizers, which decouples parameter updates into a dual-component architecture: an unmodified primary task-optimization channel driven by Adam moment estimates, and a state-dependent geometric correction regulated by a moving loss center. Above the center, DAPR executes preconditioned descent, and at or below the center, it applies a stabilized projected ascent correction that limits direct cancellation of the Adam direction, combined with an initialization-anchoring penalty that restrains parameter drift. Local two-step analysis identifies conditions under which the correction acts along a weighted gradient-norm penalty, linking its mechanism to Sharpness-Aware Minimization (SAM) while operating strictly within a single forward–backward pass ($\approx$ AdamW runtime). Across 10k step SFT and DPO on four model configurations scaling from 4B to 14B, DAPR improves capability retention in extended SFT, with lower endpoint task losses on the evaluated 4B/8B models and setting-dependent results under DPO. Finally, we introduce **LA-DAPR**, which accommodates heterogeneous convergence rates in multilingual tuning via language-specific dynamic centers, consistently improving cross-lingual capability retention.

1 INTRODUCTION

Supervised fine-tuning (SFT) and direct preference optimization (DPO) adapt large language models (LLMs) to instructions and human preferences (Ouyang et al., 2022; Rafailov et al., 2023). However, optimizing directly for target alignment frequently triggers degradation of previously acquired core capabilities, a phenomenon known as alignment tax (Lin et al., 2026). In our controllable preliminaries, under extended 10k step SFT on UltraChat, standard AdamW causes severe degradation in mathematical and scientific reasoning, where GSM8K plunges by 15.77 percentage points (pp) on Llama-3-8B (from 49.43% to 33.66%) and by 13.33 pp on Gemma-3-4B (from 37.83% to 24.50% as presented in Table 2). Continued post-training therefore demands coordinating target-task learning with principled capability retention.

Our idea starts from local loss-landscape geometry that provides a fundamental perspective on this degradation (Foret et al., 2021). In our diagnostic subspace probes across 4B–14B architectures, AdamW has larger measured endpoint top Hessian eigenvalues than SAM (Table 1). For example, on Llama-3-8B, AdamW's measured top eigenvalue is $\lambda_{\max}(\boldsymbol{H}) = 1{,}984.0$ with trace $\mathrm{Tr}(\boldsymbol{H}) = 5{,}420$, whereas Sharpness-Aware Minimization (SAM) suppresses this sharpness to $\lambda_{\max} = 1{,}380.0$ and retains $+12.96$ pp GSM8K over AdamW. SAM minimizes loss within an adversarial parameter neighborhood (Foret et al., 2021), but computing this perturbed gradient requires an auxiliary forward–backward pass per step, imposing a severe $1.87\times$–$1.98\times$ runtime penalty. This motivates the question regarding the efficiency: *can we retain the primary task-optimization channel while using the current gradient to apply a state-dependent geometric correction that suppresses sharpness at single-pass computational cost?* Intuitively, SAM's neighborhood objective introduces an implicit gradient-norm penalty at first order in the perturbation radius. Ideally, in a simplified scalar

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.

Table 1: **Subspace curvature and training cost after 10K SFT steps.** Geometry is evaluated in the trainable LoRA parameter subspace across scales. DAPR suppresses dominant Hessian eigenvalues ($\lambda_{\max}$) relative to AdamW while preserving single-backward runtime ($1.01\times$–$1.07\times$). Extended evaluations appear in Table 9. Runtime is averaged over 30 steps after five AdamW warm-up steps, with CUDA synchronization before and after timing, and normalized to AdamW. Only gradient clearing, forward/backward computation, and optimizer updates are timed.

LLM	Optimizer	Local geometry			Training cost	
		Top eigenvalue $\lambda_{\max}$	Hessian trace $\mathrm{Tr}(\boldsymbol{H})$	Gradient norm $\|\nabla\mathcal{L}\|_2$	Backward passes	Relative runtime
Gemma-3 4B	AdamW	437.2	415	0.210	1	1.00×
	SAM	295.0	310	0.160	2	1.98×
	DAPR (Ours)	**312.8**	1,059	**0.172**	**1**	**1.01×**
Llama-3 8B	AdamW	1,984.0	5,420	0.285	1	1.00×
	SAM	1,380.0	3,620	0.185	2	1.96×
	DAPR (Ours)	**1,440.8**	**3,890**	**0.198**	**1**	**1.01×**
Gemma-3 12B	AdamW	462.7	1,329.2	9.736	1	1.00×
	SAM	8.9	77.6	0.692	2	1.87×
	DAPR (Ours)	**164.0**	**193.8**	**2.629**	**1**	**1.07×**
Qwen-2.5 14B	AdamW	312.4	985.2	8.412	1	1.00×
	SAM	6.4	54.2	0.518	2	1.89×
	DAPR (Ours)	**112.5**	**142.1**	**1.845**	**1**	**1.04×**

model, an oscillating point crossing an operating threshold generates a squared-gradient-norm regularization term in its two-step expansion, sharing a curvature-dependent gradient direction with SAM's local expansion (Appendix A.1). This connection motivates using successive updates to introduce a regularizing correction without an auxiliary gradient evaluation.

We propose **Descent-Anchored Projected Regularization (DAPR)** around this design to work for Adam-family optimizers by coordinating the correction's direction, strength, and loss-dependent branch selection. DAPR decouples parameter updates into a dual-channel architecture. Unmodified task gradients drive Adam moment estimates, while a state-dependent geometric correction is applied separately. A moving loss center regulates the operating regime, selecting preconditioned descent above the center or projected ascent at or below it. A stabilized projection limits direct cancellation of the Adam direction, while initialization anchoring penalizes parameter displacement from the starting checkpoint. Empirically, as presented in Table 1 across 10k-step SFT on UltraChat, DAPR restricts GSM8K forgetting to just 2.58 pp on Llama-3-8B (46.85% vs. 33.66% for AdamW, matching SAM's 46.62%) and suppresses top Hessian eigenvalues to $\lambda_{\max} = 1{,}440.8$ (vs. 1,984.0 for AdamW) at near-zero runtime overhead. Meanwhile, DAPR can handle multiple moving centers simultaneously for different languages to support multilingual instruction tuning, and the language-aware extension **LA-DAPR** improves target performance by accommodating heterogeneous convergence rates via language-specific dynamic centers, requiring only that these centers be computed separately for each language.

Contributions. **(1) Conditional update design:** DAPR coordinates a retained Adam-family task-update channel with separately controlled geometric corrections using one forward–backward pass, and LA-DAPR extends the design through language-specific centers. **(2) Local mechanism analysis:** Projection geometry and two-step decompositions explain the correction's directional properties and its conditional connection to gradient-norm regularization. **(3) Cross-scale capability evaluation:** Experiments on four model configurations (Gemma-3-4B, Llama-3-8B, Gemma-3-12B, Qwen-2.5-14B) show strong retention gains in extended SFT on 4B/8B models and characterize setting-dependent outcomes under DPO and larger-scale training at near-AdamW cost.

2 METHODOLOGY

DAPR builds on the local regularization effect of a descent–ascent pair. We formulate **Descent-Anchored Projected Regularization (DAPR)** for continued post-training by retaining task-gradient moments, coordinating correction and task directions, and adapting intervention to training state. We develop the method for Adam-family optimizers (Kingma & Ba, 2015); below, Adam denotes their shared moment-estimation mechanism and the resulting adaptive direction.

2.1 Overall Design

We first isolate the effect in a simplified crossing model inspired by threshold-based loss regularization (Ishida et al., 2020). A descent step followed by partial ascent evaluates gradients at successive parameters, and their difference introduces a curvature-dependent term without another gradient evaluation within either step.

Algorithm 1: DAPR and LA-DAPR with Adam Moment Estimates

Input: Step size η, Adam parameters $\beta_1, \beta_2, \epsilon_{\text{Adam}}$, projection stabilizer ϵ_{proj}, α_0, γ, center parameters τ, τ_s, λ, anchor strength λ_{base}, and variant (DAPR or LA-DAPR).
Initialize: Zero Adam moments for each tensor; center statistics as in Sections 2.3 and 1; initial parameters $\boldsymbol{\theta}_0^{(j)}$.

1. Compute the active-batch loss L_t and all task-gradient blocks $\boldsymbol{g}_t^{(j)}$ with one forward–backward pass.
2. Update the shared center, or the active language's center for LA-DAPR; c_t below denotes the selected center.
3. For each parameter tensor j with a gradient, update Adam moment estimates and the adaptive direction:

$$\begin{aligned}\boldsymbol{m}_t^{(j)} &= \beta_1 \boldsymbol{m}_{t-1}^{(j)} + (1-\beta_1)\boldsymbol{g}_t^{(j)},\\ \boldsymbol{v}_t^{(j)} &= \beta_2 \boldsymbol{v}_{t-1}^{(j)} + (1-\beta_2)(\boldsymbol{g}_t^{(j)})^2,\\ \boldsymbol{P}_t^{(j)} &= \operatorname{diag}((\sqrt{\widehat{\boldsymbol{v}}_t^{(j)}} + \epsilon_{\text{Adam}}\mathbf{1})^{-1}), \qquad \boldsymbol{u}_t^{(j)} = \boldsymbol{P}_t^{(j)}\widehat{\boldsymbol{m}}_t^{(j)}.\end{aligned}$$

   Hats denote the usual bias correction at that tensor's update count.

4. For each block, set $\alpha_t^{(j)} = \alpha_0/(1+\gamma\|\boldsymbol{g}_t^{(j)}\|_2)$. If $L_t \le c_t$, compute

$$\boldsymbol{q}_t^{(j)} = \boldsymbol{P}_t^{(j)}\left(\boldsymbol{g}_t^{(j)} - \frac{(\boldsymbol{g}_t^{(j)})^\top \boldsymbol{P}_t^{(j)}\boldsymbol{u}_t^{(j)}}{(\boldsymbol{u}_t^{(j)})^\top \boldsymbol{P}_t^{(j)}\boldsymbol{u}_t^{(j)} + \epsilon_{\text{proj}}}\boldsymbol{u}_t^{(j)}\right).$$

5. Apply Eq. (2) to each block and set $\boldsymbol{\theta}_{t+1}^{(j)} = \boldsymbol{\theta}_t^{(j)} - \eta\boldsymbol{\Delta}_t^{(j)}$.

Proposition 1 (Local regularization of a crossing pair). For a fixed twice continuously differentiable loss $\mathcal{L}$ with Lipschitz gradient and Hessian, let $\boldsymbol{g}_t = \nabla\mathcal{L}(\boldsymbol{\theta}_t)$. Consider a step $-\eta(1+\alpha_t)\boldsymbol{g}_t$ above the center followed by $+\eta(1-\alpha_{t+1})\boldsymbol{g}_{t+1}$ at or below it, where η is fixed and $\alpha_t = \alpha_0/(1+\gamma\|\boldsymbol{g}_t\|_2)$, $0 < \alpha_0 < 1$, $\gamma > 0$. The two-step expansion is

$$\boldsymbol{\theta}_{t+2} = \boldsymbol{\theta}_t - 2\alpha_t\eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t) - \eta^2(1-\alpha_t^2)\,\nabla_{\boldsymbol{\theta}}\big(\tfrac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_2^2\big)\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} + \mathcal{R}_t. \tag{1}$$

Here $\|\mathcal{R}_t\|_2 = O(\eta^2\|\boldsymbol{g}_t\|_2^2 + \eta^3\|\boldsymbol{g}_t\|_2^2)$ is under the stated bounds. The expansion contains a task-descent component and a squared-gradient-norm correction. SAM's local expansion introduces a gradient-norm penalty with the same Hessian–gradient direction but different scaling. Appendix A.1 details this local mechanism, its remainder bounds, and the comparison with SAM.

To use this mechanism during post-training, the descent–ascent intervention must coexist with Adam's historical state and continued task learning. DAPR therefore retains the Adam update and applies a separate conditional correction. Building on this local mechanism, the moving center regulates transitions between descent and ascent corrections, while projection limits direct cancellation of the retained Adam update. Let L_t denote the active-batch loss. Partition the trainable parameters into tensors indexed by j, and write their vectorizations as $\boldsymbol{\theta}_t^{(j)}$, with gradients $\boldsymbol{g}_t^{(j)} = \nabla_{\boldsymbol{\theta}^{(j)}}\mathcal{L}(\boldsymbol{\theta}_t)$. Inner products and Euclidean norms within a block equal Frobenius inner products and norms of the original tensor. Adam forms $\boldsymbol{P}_t^{(j)} = \operatorname{diag}((\sqrt{\widehat{\boldsymbol{v}}_t^{(j)}} + \epsilon_{\text{Adam}}\mathbf{1})^{-1})$ with numerical stabilizer $\epsilon_{\text{Adam}} = 10^{-8}$ and base direction $\boldsymbol{u}_t^{(j)} = \boldsymbol{P}_t^{(j)}\widehat{\boldsymbol{m}}_t^{(j)}$. DAPR computes coefficients and projections separately for each tensor; all blocks share the loss and center c_t:

$$\begin{aligned}\boldsymbol{\Delta}_t^{(j)} &= \boldsymbol{u}_t^{(j)} + \boldsymbol{\delta}_t^{(j)},\\ \boldsymbol{\delta}_t^{(j)} &= \begin{cases}\alpha_t^{(j)}\boldsymbol{P}_t^{(j)}\boldsymbol{g}_t^{(j)}, & L_t > c_t,\\ -\alpha_t^{(j)}\boldsymbol{q}_t^{(j)} + \lambda_{\text{base}}(\boldsymbol{\theta}_t^{(j)} - \boldsymbol{\theta}_0^{(j)}), & L_t \le c_t.\end{cases}\end{aligned} \tag{2}$$

Each block is updated by $\boldsymbol{\theta}_{t+1}^{(j)} = \boldsymbol{\theta}_t^{(j)} - \eta \boldsymbol{\Delta}_t^{(j)}$. The center selects additional preconditioned descent above the threshold or projected ascent with initialization anchoring at or below it. Both branches retain the Adam update and add the selected correction.

2.2 Adapting the Mechanism to Adam-Family Optimizers

Separating task moments from the correction. The intended descent–ascent effect depends on the realized update directions. Feeding sign-reversed gradients into Adam moment estimators mixes the current intervention with historical momentum, which can delay or prevent the intended first-order ascent (Appendix A.2). DAPR drives both moment estimators strictly with unmodified task gradients, applying the state-dependent correction externally after forming the base Adam direction $\boldsymbol{u}_t^{(j)}$. The resulting moments are driven by task gradients along the DAPR training trajectory.

Limiting direct directional cancellation. Separating the states determines where the correction is applied; its direction still needs to be coordinated with the Adam direction. An unprojected ascent correction can directly oppose the retained task update. For an arbitrary parameter block, suppress the superscript (j) in this paragraph and Proposition 2. DAPR controls this interaction using the weighted inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{P}_t} = \boldsymbol{x}^\top \boldsymbol{P}_t \boldsymbol{y}$:

$$\boldsymbol{g}_{\mathrm{proj}} = \frac{\boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{u}_t}{\boldsymbol{u}_t^\top \boldsymbol{P}_t \boldsymbol{u}_t + \epsilon_{\mathrm{proj}}} \boldsymbol{u}_t, \qquad \boldsymbol{g}_{\mathrm{ortho}} = \boldsymbol{g}_t - \boldsymbol{g}_{\mathrm{proj}}, \tag{3}$$

where we set the projection stabilizer $\epsilon_{\mathrm{proj}} = 10^{-8}$ empirically. The parameter-space correction is $\boldsymbol{q}_t = \boldsymbol{P}_t \boldsymbol{g}_{\mathrm{ortho}}$. The following result characterizes both its direction and the choice of projection.

Proposition 2 (Directional and variational characterization). Let the current $\boldsymbol{P}_t \succ 0$, gradient, and Adam direction be fixed. With $a_t = \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{u}_t$ and $b_t = \boldsymbol{u}_t^\top \boldsymbol{P}_t \boldsymbol{u}_t$ ($b_t + \epsilon_{\mathrm{proj}} > 0$), the correction satisfies $\boldsymbol{u}_t^\top \boldsymbol{q}_t = \frac{\epsilon_{\mathrm{proj}} a_t}{b_t + \epsilon_{\mathrm{proj}}}$ and $\boldsymbol{g}_t^\top \boldsymbol{q}_t = \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{g}_t - \frac{a_t^2}{b_t + \epsilon_{\mathrm{proj}}} \geq 0$. With only the projection stabilizer set to zero and $\boldsymbol{u}_t \neq \boldsymbol{0}$, the correction is strictly orthogonal to $\boldsymbol{u}_t$ ($\boldsymbol{u}_t^\top \boldsymbol{q}_t = 0$), and is the unique solution of $\min_{\boldsymbol{z}:\, \boldsymbol{u}^\top \boldsymbol{z} = 0} \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{P}\boldsymbol{g}\|_{\boldsymbol{P}^{-1}}^2$. For $\boldsymbol{u}_t = \boldsymbol{0}$, set $\boldsymbol{q}_t = \boldsymbol{P}_t \boldsymbol{g}_t$. This constraint is imposed separately in every block. The proof, the product of block constraints, and the soft-constraint formulation are in Appendix A.3. Thus projection retains an ascent component while limiting direct cancellation of the Adam direction. Therefore, state separation addresses historical mixing, and projection addresses the current update geometry.

Initialization anchoring. While projection coordinates the current update direction, we still need to control accumulated parameter displacement. Specifically, DAPR adds an initialization anchor $\lambda_{\mathrm{base}}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_0)$ ($\lambda_{\mathrm{base}} = 0.01$) exclusively in the below-center branch. It is the gradient of $\frac{\lambda_{\mathrm{base}}}{2} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2^2$, penalizing displacement from initialization. Under exact block projection, $\boldsymbol{u}_t^\top \boldsymbol{\Delta}_t = \|\boldsymbol{u}_t\|_2^2 + \lambda_{\mathrm{base}}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_0)^\top \boldsymbol{u}_t$. The anchor complements the directional projection with a displacement penalty. Section 3.3 reports the component comparisons.

2.3 Adapting the Correction to Training State

Dynamic operating center. The crossing mechanism needs a reference loss level that follows ongoing training. DAPR uses a moving center to select the correction branch, updating exponential moving averages (EMAs) of the loss and positive one-step progress:

$$\mu_t = (1 - \tau)\mu_{t-1} + \tau L_t, \quad s_t = (1 - \tau_s) s_{t-1} + \tau_s \max(0, L_{t-1} - L_t), \quad c_t = \mu_t - \lambda s_t, \tag{4}$$

where $\tau, \tau_s \in (0, 1)$ are smoothing factors (defaults $\tau = \tau_s = 0.1$), and $\lambda > 0$ controls the offset (default $\lambda = 1.0$). At the first observed batch, initialize the loss mean and previous loss to that batch's loss and the progress statistic to zero, giving a center equal to the initial observed loss. The loss EMA provides a recent reference level, while positive progress shifts the threshold below it. At each step, DAPR updates the shared statistics and compares the active loss with the center to select the correction branch. This makes branch selection depend on the current batch and recent progress rather than a fixed loss level.

Adaptive strength. Gradient norms vary across tensor blocks and training steps. For each tensor block, DAPR scales the correction via $\alpha_t^{(j)} = \alpha_0/(1+\gamma\|\boldsymbol{g}_t^{(j)}\|_2)$ (defaults $\alpha_0 = 0.5$, $\gamma = 1.0$). A larger block gradient yields a smaller coefficient. The current gradient norm therefore controls the block-specific correction strength. In this way, branch selection is governed by global loss dynamics, while local gradient norms modulate intervention strength.

2.4 Discussion

Interaction between task updates and intervention. We now examine how the scalar crossing effect manifests in the practical multi-tensor rule. The following decomposition fixes loss, step size, and preconditioner over a crossing pair while accounting for Adam moments, dynamic block scaling, projection rotation, and initialization anchoring.

Lemma 1 (Two-Step Decomposition of the Practical DAPR Update). Concatenate the blocks and let $\boldsymbol{P} = \mathrm{blkdiag}_j(\boldsymbol{P}^{(j)})$ be fixed over the pair.

$$\boldsymbol{A}_t = \mathrm{blkdiag}_j(\alpha_t^{(j)}\boldsymbol{I}_j), \quad \boldsymbol{\Pi}_t = \mathrm{blkdiag}_j\left(\frac{\boldsymbol{u}_t^{(j)}(\boldsymbol{u}_t^{(j)})^\top \boldsymbol{P}^{(j)}}{(\boldsymbol{u}_t^{(j)})^\top \boldsymbol{P}^{(j)}\boldsymbol{u}_t^{(j)} + \epsilon_{\mathrm{proj}}}\right), \quad \boldsymbol{Q}_t = \boldsymbol{I} - \boldsymbol{\Pi}_t.$$

For $\epsilon_{\mathrm{proj}} = 0$, set the block of $\boldsymbol{\Pi}_t$ to zero when $\boldsymbol{u}_t^{(j)} = \boldsymbol{0}$. The subscripts $\perp$ and $\|$ denote the stabilized decomposition, with strict orthogonality in the $\epsilon_{\mathrm{proj}} = 0$ case. Write $\boldsymbol{g}_{\perp,t} = \boldsymbol{Q}_t\boldsymbol{g}_t$, $\boldsymbol{g}_{\|,t} = \boldsymbol{\Pi}_t\boldsymbol{g}_t$, $\Delta\boldsymbol{A}_t = \boldsymbol{A}_{t+1} - \boldsymbol{A}_t$, $\Delta\boldsymbol{Q}_t = \boldsymbol{Q}_{t+1} - \boldsymbol{Q}_t$, and $\boldsymbol{d}_t = \boldsymbol{u}_t + \boldsymbol{A}_t\boldsymbol{P}\boldsymbol{g}_t$. For gradients of the same fixed loss with an L_H-Lipschitz Hessian and fixed step size, an above-center step (followed by a below-center step) satisfies

$$\begin{aligned}\boldsymbol{\theta}_{t+2} = \boldsymbol{\theta}_t &- \eta(\boldsymbol{u}_t + \boldsymbol{u}_{t+1}) - \eta\boldsymbol{A}_t\boldsymbol{P}\boldsymbol{g}_{\|,t} + \eta\Delta\boldsymbol{A}_t\boldsymbol{P}\boldsymbol{g}_{\perp,t} \\ &- \eta^2\boldsymbol{A}_{t+1}\boldsymbol{P}\boldsymbol{Q}_t\boldsymbol{H}_t\boldsymbol{d}_t + \eta\boldsymbol{A}_{t+1}\boldsymbol{P}\Delta\boldsymbol{Q}_t\boldsymbol{g}_t \\ &- \eta\lambda_{\mathrm{base}}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_0) + \widetilde{\mathcal{R}}_t.\end{aligned} \tag{5}$$

Here $\boldsymbol{H}_t$ is the full parameter Hessian. The Adam term spans both steps, while remaining terms capture the parallel correction, coefficient shifts, projection rotation, and anchoring (proven in Appendix A.4). Corollary 1 identifies conditions where the leading correction acts as a preconditioned transverse gradient penalty. Section 3.2 evaluates empirical capability retention.

Computational complexity. Let d denote the trainable parameter count and C_{FB} the forward–backward pass cost. Because $\boldsymbol{P}_t$ is diagonal, preconditioning, inner products, and anchoring operate entrywise in $O(d)$ arithmetic without dense matrices. DAPR costs $C_{\mathrm{FB}} + O(d)$ per step, matching Adam-family optimizers, whereas SAM requires $2C_{\mathrm{FB}} + O(d)$. Buffers add $O(d)$ memory; center statistics need $O(1)$ scalar state.

Model Extension. Multilingual instruction tuning exhibits heterogeneous loss dynamics across linguistic families. A shared loss center pools statistics across languages, so one language's loss history can affect another language's branch selection. The language-aware extension (**LA-DAPR**) maintains independent center statistics $\{\mu_t^{(l)}, s_t^{(l)}, c_t^{(l)}\}$ for each language $l \in \mathcal{L}$ while sharing underlying model parameters and Adam state. For a language labeled with a language ID l in a mini-batch,

$$\mu_t^{(l)} = (1-\tau)\mu_{t-1}^{(l)} + \tau\mathcal{L}^{(l)}(\boldsymbol{\theta}_t), \quad s_t^{(l)} = (1-\tau_s)s_{t-1}^{(l)} + \tau_s\max\left(0, \mathcal{L}_{\mathrm{prev}}^{(l)} - \mathcal{L}^{(l)}(\boldsymbol{\theta}_t)\right), \quad c_t^{(l)} = \mu_t^{(l)} - \lambda s_t^{(l)}. \tag{6}$$

When language l first appears, initialize $\mu^{(l)} = \mathcal{L}_{\mathrm{prev}}^{(l)} = L_t$ and $s^{(l)} = 0$, so $c_t^{(l)} = L_t$. Subsequently, $\mathcal{L}_{\mathrm{prev}}^{(l)}$ is the loss from the most recent batch of that language. Only the active language's history is updated, and $c_t^{(l)}$ is supplied to the conditional update. For K languages, tracking separate center statistics requires $O(K)$ scalar storage and $O(1)$ work per step, preserving single-pass asymptotic efficiency. In other words, the multilingual extension requires no additional computation but simply needs a language ID for each language to record each language's history separately. Language-specific centers give different reported loss-center branch-switch milestones (Chinese at 6,980–7,240 steps vs. 3,650–4,110 for European languages), yielding $+4.70$ pp accuracy gains on Chinese for Gemma-3-4B (Section 3.4). Algorithm 1 details the complete algorithm.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We evaluate capability retention under continued post-training, its dependence on the training setting, and adaptation to multilingual data. The main experiments use Gemma-3-4B and Llama-3-8B, with scaling studies on Gemma-3-12B and Qwen-2.5-14B (Gemma Team et al., 2025; Grattafiori et al., 2024; Qwen Team et al., 2024). We use UltraChat-200k for instruction SFT (Ding et al., 2023), UltraFeedback-Binarized for DPO (Cui et al., 2024), and joint Chinese, German, Spanish, and French Aya training for multilingual evaluation (Singh et al., 2024). We additionally evaluate limited-data SFT on LIMA. Capability benchmarks are ARC Challenge, MMLU, HellaSwag, and GSM8K, evaluated using `lm-evaluation-harness` (Gao et al., 2024); multilingual results report target-language instruction-following accuracy. GSM8K uses exact match, with flexible answer extraction in the DPO stress and scaling studies.

We use AdamW (Loshchilov & Hutter, 2019) as the standard optimizer reference and SAM ($\rho = 0.05$) as the two-pass baseline. All post-training experiments use LoRA ($r = 16, \alpha = 32$, dropout 0.05) targeting query and value projections (W_q, W_v). SFT uses learning rates $\eta = 2 \times 10^{-4}$ (4B), 1×10^{-4} (8B), and 2×10^{-5} (12B/14B) with cosine decay and 5% linear warmup. Standard DPO uses $\beta = 0.10$ and $\eta = 5 \times 10^{-6}$ (2×10^{-5} on 12B/14B), while DPO stress jointly sets $\beta = 0.01$ and $\eta = 2 \times 10^{-5}$. UltraChat and UltraFeedback use a 10K-step horizon without early stopping, and LIMA runs for five epochs. Table 2 reports benchmark scores across scales (4B/8B results are averaged over three seeds). Additional experimental details appear in Appendix B.

3.2 CROSS-ARCHITECTURE CAPABILITY RETENTION ACROSS SCALES (4B–14B)

To quantify capability retention relative to pre-adaptation weights, Table 2 reports unadapted base scores alongside 10K-step SFT, standard DPO, and stress DPO endpoints across four model configurations (4B–14B), measuring benchmark forgetting via $\mathrm{Forget}(b) = s_{\mathrm{base}}(b) - s_{\mathrm{post}}(b)$. Figure 1 illustrates the full 10K-step trajectories. We structure our empirical investigation around three central questions:

Model	Method	SFT				Standard DPO ($\beta = 0.10$)				Stress DPO ($\beta = 0.01$)			
		ARC-C	MMLU	HS	GSM8K	ARC-C	MMLU	HS	GSM8K	ARC-C	MMLU	HS	GSM8K
Gemma-3 4B	Base	54.78	57.90	75.93	37.83	54.78	57.90	75.93	37.83	54.78	57.90	75.93	37.83
	AdamW	50.85	54.80	72.85	24.50	**58.11**	58.58	**77.97**	44.05	56.66	58.50	77.60	44.73
	SAM	54.30	57.08	75.38	**35.95**	57.85	58.60	77.92	44.20	59.98	**58.52**	80.20	**50.11**
	DAPR	**54.44**	**57.15**	**75.47**	35.80	57.42	**58.61**	77.90	**44.58**	**60.49**	58.21	**80.40**	49.36
Llama-3 8B	Base	54.35	62.18	79.19	49.43	54.35	62.18	79.19	49.43	54.35	62.18	79.19	49.43
	AdamW	51.44	57.65	70.45	33.66	56.14	63.30	81.46	**54.59**	58.62	63.00	83.35	**54.59**
	SAM	55.10	61.18	74.70	46.62	56.35	63.38	81.52	54.10	**60.24**	**63.11**	**84.89**	50.64
	DAPR	**55.29**	**61.25**	**74.85**	**46.85**	**56.66**	**63.47**	**81.56**	53.90	57.76	62.54	83.44	50.80
Gemma-3 12B	Base	64.76	71.46	81.78	71.19	64.76	71.46	81.78	71.19	64.76	71.46	81.78	71.19
	AdamW	64.16	71.53	**82.10**	**74.37**	**70.82**	**71.88**	**84.44**	**76.88**	72.01	71.73	84.90	**78.32**
	SAM	64.59	71.73	81.93	74.22	66.72	71.54	83.49	73.92	**72.78**	**71.79**	**87.37**	77.03
	DAPR	**64.85**	**71.76**	81.96	74.00	65.27	71.61	82.36	73.31	72.27	71.34	84.37	76.72
Qwen-2.5 14B	Base	59.04	77.51	82.96	89.54	59.04	77.51	82.96	89.54	59.04	77.51	82.96	89.54
	AdamW	59.90	77.28	**83.06**	88.02	59.47	77.75	83.13	89.92	62.46	77.82	83.71	86.58
	SAM	61.01	77.18	82.94	88.48	**60.07**	**77.87**	**83.97**	89.61	**65.27**	77.67	**86.94**	**89.46**
	DAPR	**61.09**	**77.29**	82.97	**88.63**	59.90	77.81	83.25	**89.99**	64.42	**77.89**	83.92	76.50

Table 2: **Capability retention across scales (4B–14B)** after 10K steps of UltraChat SFT, standard UltraFeedback DPO, and stress DPO. Scores are percentages (means over three seeds for 4B/8B; SDs in Appendix B). Bold marks top performance among adapted methods; Base denotes pre-adaptation checkpoint. HS: HellaSwag. Multi-milestone stress trajectories and displacement appear in Table 7.

Q1: Does DAPR mitigate capability forgetting in long-horizon SFT on small models? Concretely, small models have insufficient model capacity to adopt alignment or preference in post-training, which are prone to capability forgetting. In extended SFT on capacity-constrained mod-

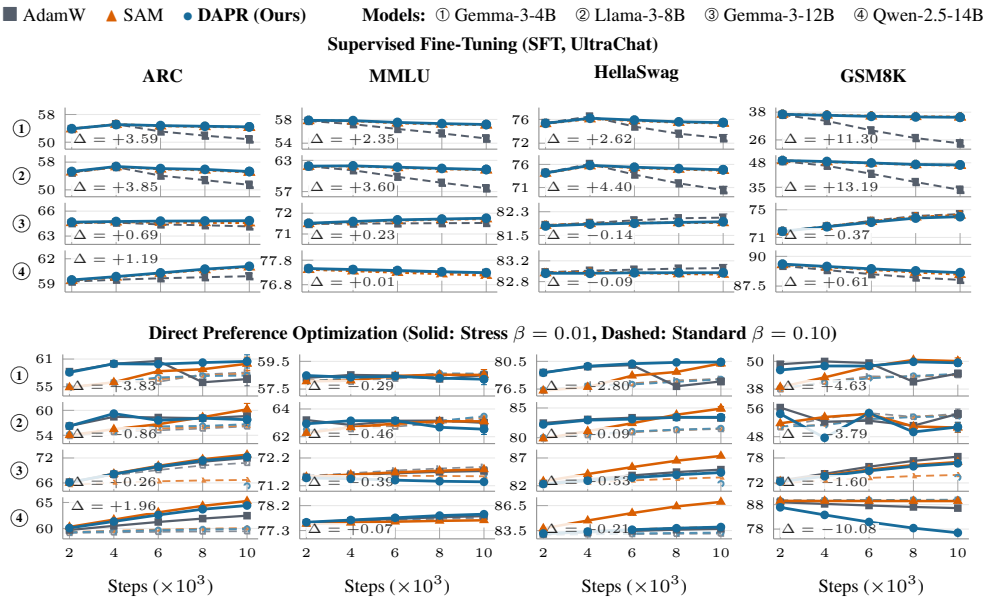

Figure 1: **Capability retention trajectories under SFT (top) and DPO (bottom) across scales (4B–14B).** Rows denote architectures: ① Gemma-3-4B, ② Llama-3-8B, ③ Gemma-3-12B, ④ Qwen-2.5-14B. In DPO, solid curves denote stress ($\beta = 0.01, \eta = 2 \times 10^{-5}$) and dashed denote standard DPO ($\beta = 0.10, \eta = 5 \times 10^{-6}$). Insets ($\Delta$): final DAPR–AdamW stress margins (pp). Error bars: 10K SD across 3 seeds for 4B/8B. Scores in %; steps in thousands.

els ($\leq$ 8B), standard AdamW triggers severe reasoning degradation as expected. On Llama-3-8B, $\text{Forget}(\text{GSM8K}) = +15.77$ pp (falling to 33.66%), while Gemma-3-4B suffers +13.33 pp degradation (24.50%). DAPR substantially reduces this degradation, restricting GSM8K forgetting to 2.58 pp on Llama (46.85%) and 2.03 pp on Gemma (35.80%), matching two-pass SAM (46.62% and 35.95%) within 0.23 pp at single-pass cost. Across ARC Challenge, MMLU, and HellaSwag, DAPR improves capability retention relative to AdamW (+2.35 to +4.40 pp over AdamW), consistent with the local two-step analysis in Appendix A.4, where Corollary 1 identifies a weighted gradient-norm regularization effect under the stated conditions. Furthermore, DAPR preserves target-task learning alongside capability retention, achieving lower mean training and validation losses than AdamW in this setting (Figure 2). Figure 2 (Appendix B.2.3) confirms that improved retention coexists with lower held-out validation loss (0.431 vs. 0.449 on Gemma, 0.526 vs. 0.570 on Llama).

Q2: How does DAPR behave under standard versus aggressive (stress) preference optimization? Under standard DPO ($\beta = 0.10$, Table 2), all three optimizers perform comparably on 4B and 8B models (differing by $\leq$ 0.69 pp), potentially reflecting the regularizing effect of DPO's reference-policy constraint. However, when preference optimization is aggressively stressed ($\beta = 0.01, \eta = 2 \times 10^{-5}$), standard AdamW destabilizes: on Gemma-4B, DAPR outperforms AdamW by +3.83 pp on ARC, +2.80 pp on HellaSwag, and +4.63 pp on GSM8K (49.36% vs. 44.73%), closely tracking SAM (50.11%). On Llama-8B, retention under stress is mixed, matching SAM's trajectory. Sequence margin diagnostics in Table 8 (Appendix B.2.4) reveal that AdamW suffers severe margin inversion on Gemma-4B (collapsing to -34.44), whereas DAPR maintains a positive margin (+3.91, matching SAM at +0.82). Table 7 (Appendix B.2.2) details the multi-milestone progression and parameter displacement $\|\Delta W\|_F$.

Q3: How do retention dynamics scale to 12B and 14B architectures? Q1 and Q2 examine capability retention under SFT and DPO on 4B/8B models. As parameter capacity increases to Gemma-3-12B and Qwen-2.5-14B, the capability retention landscape shifts. In 10K-step SFT, larger models exhibit heightened resilience against catastrophic forgetting. On the other hand, DAPR still yields modest improvements on ARC and GSM8K on Qwen-14B (88.63% vs. 88.02%) while preserving base capabilities within 0.91 pp. Under standard DPO, differences remain minimal on

Qwen-14B (89.99% vs. 89.92% on GSM8K), and under stress DPO, DAPR maintains competitive accuracy on Gemma-12B (72.27% ARC, 76.72% GSM8K) and Qwen-14B (64.42% ARC, 77.89% MMLU). However, DAPR trails AdamW by 5.55 pp on ARC-C under standard DPO on Gemma-12B and by 10.08 pp on GSM8K under stress DPO on Qwen-14B. These findings suggest that greater model capacity helps reduce alignment tax under SFT.

3.3 Component Ablations and Hyperparameter Analysis

Table 3 reports a systematic evaluation across 15 configurations on Gemma-3-4B after 10K UltraChat SFT steps. The reference AdamW baseline and default DAPR rows match the benchmark evaluation in Table 2, where default DAPR achieves a three-task mean of 62.35% (+2.85 pp over AdamW's 59.50%), leading across ARC (54.44%), MMLU (57.15%), and HellaSwag (75.47%). Ablating individual components confirms the necessity of each design choice: disabling orthogonal projection lowers the mean to 60.70% (dropping MMLU to 55.11%), while fixing α reduces the mean to 60.71% (lowering MMLU to 55.41%). Removing initialization anchoring ($\lambda_{\text{base}} = 0$) permits parameter drift, dropping the mean to 61.00%, and omitting anchoring while relying solely on weight decay ($\text{wd} = 0.01$) degrades retention to 60.04%. In the center-offset sweep, moderate values ($\lambda \in [0.2, 1.0]$) yield robust retention (60.61%–62.35%), whereas an overly wide offset ($\lambda = 2.0$) delays intervention, degrading performance to 59.16% (falling below AdamW). Correction scales in $\alpha_0 \in [0.3, 0.7]$ provide stable preservation (60.79%–62.35%), whereas weak correction ($\alpha_0 = 0.1$) or overly aggressive perturbation ($\alpha_0 = 0.9$) degrades retention (60.19% and 60.63%, respectively).

Configuration	ARC	MMLU	HellaSwag	Mean
AdamW	50.85	54.80	72.85	59.50
DAPR (default)	**54.44**	**57.15**	75.47	**62.35**
Component and regularization controls				
Without projection	51.54	55.11	75.46	60.70
Fixed α	51.37	55.41	75.35	60.71
Without anchoring ($\lambda_{\text{base}} = 0$, wd $= 0.0$)	50.51	56.66	**75.82**	61.00
Weight decay only ($\lambda_{\text{base}} = 0$, wd $= 0.01$)	48.98	55.80	75.34	60.04
Anchoring + weight decay ($\lambda_{\text{base}} = 0.01$, wd $= 0.01$)	52.56	56.51	75.61	61.56
Center-offset multiplier λ				
$\lambda = 0.2$	51.28	55.86	75.80	60.98
$\lambda = 0.5$	50.34	56.29	75.21	60.61
$\lambda = 2.0$	47.78	53.87	**75.82**	59.16
$\lambda = 3.0$	50.77	56.49	75.07	60.78
Correction scale α_0				
$\alpha_0 = 0.1$	49.91	54.86	75.78	60.19
$\alpha_0 = 0.3$	51.19	55.60	75.73	60.84
$\alpha_0 = 0.7$	50.77	56.26	75.35	60.79
$\alpha_0 = 0.9$	50.60	56.24	75.04	60.63

Table 3: **Systematic ablation and sensitivity results across 15 configurations** on Gemma-3-4B after 10K UltraChat SFT steps. Scores are percentages; bold marks highest score per column. The reference AdamW baseline and default DAPR match the benchmark evaluation in Table 2, achieving top retention (62.35%, +2.85 pp over AdamW).

Model	Method	ARC	MMLU	HS	GSM8K
Llama-3 8B	Base	54.35	62.18	79.19	49.43
	AdamW	49.91	**57.89**	80.66	30.93
	DAPR	**50.26**	57.83	**80.90**	**31.99**
Gemma-3 4B	Base	54.78	57.90	75.93	37.83
	AdamW	**49.74**	**54.09**	**77.35**	30.78
	DAPR	47.70	53.72	77.27	**31.24**

Table 4: **Capability retention after 5-epoch LIMA SFT** (%). Bold marks top performance between AdamW and DAPR; Base denotes pre-adaptation checkpoint. HS: HellaSwag.

To test whether DAPR functions when data is scarce and optimizer statistics are limited, we evaluate 5-epoch SFT on Gemma-3-4B and Llama-3-8B using LIMA's 1,000 demonstrations (Zhou et al.,

2023). Table 4 reports retention relative to unadapted base checkpoints. DAPR yields smaller gains than in extended SFT (GSM8K improves by $+1.06$ pp on Llama, $+0.46$ pp on Gemma) and remains below base capabilities (31.99% vs. 49.43% on Llama), demonstrating that retention benefits do not transfer uniformly to low-data regimes.

3.4 Multilingual Instruction Tuning

Model	Language	AdamW	SAM	DAPR	LA-DAPR	Branch switch step
Gemma 3-4B	Chinese	30.71	38.65	34.20	**38.90**	7,240
	German	40.20	**47.90**	46.10	47.85	3,820
	Spanish	42.15	**48.75**	47.20	48.60	3,950
	French	41.30	**46.80**	45.40	46.72	4,110
Llama 3-8B	Chinese	33.45	41.10	36.80	**41.25**	6,980
	German	44.10	**51.40**	49.30	51.35	3,650
	Spanish	45.80	**52.60**	50.90	52.45	3,810
	French	45.10	**51.20**	49.60	51.10	3,920

Table 5: **Multilingual instruction accuracy after 10K-step Aya SFT** (%; mean over 3 seeds). Bold marks the highest mean for each model–language pair. Branch switch step marks transition to projected ascent ($\mathcal{L}^{(l)}(\boldsymbol{\theta}_t) \leq c_t^{(l)}$).

We evaluate language-aware centering (LA-DAPR) on Gemma-3-4B and Llama-3-8B using joint 10K-step multilingual Aya fine-tuning (Chinese, German, Spanish, French). As shown in Table 5, LA-DAPR outperforms shared-center DAPR across all languages, with the largest gains in Chinese ($+4.70$ pp on Gemma, $+4.45$ pp on Llama; 1.32–2.05 pp on European languages). LA-DAPR surpasses AdamW across all eight settings and matches SAM within 0.25 pp at single-pass cost. Reported branch-switch milestones (Chinese at $6{,}980$–$7{,}240$ steps vs. $3{,}650$–$4{,}110$ for European languages) confirm that branch selection naturally accommodates distinct linguistic convergence dynamics (Appendix B.3.1).

4 Related Work

Capability degradation and parameter regularization. Post-training adaptation frequently induces severe forgetting of pre-adaptation competencies (Lin et al., 2026), motivating continuous tracking suites like CapTrack (Thede et al., 2026). Standard mitigations constrain policy distributions via output-space KL divergence against a frozen reference model (Ouyang et al., 2022; Rafailov et al., 2023), incurring auxiliary forward passes and memory overhead. Conversely, parameter-space regularizers like Elastic Weight Consolidation (EWC; Kirkpatrick et al. 2017) penalize movement along Fisher information directions. However, strong parameter constraints can limit adaptation to new tasks, creating a trade-off between retention and plasticity (Ning et al., 2026). DAPR avoids auxiliary inference by coupling active Adam updates with dynamic transverse corrections and base anchoring.

Sharpness-aware optimization for language models. Sharpness-Aware Minimization (SAM) (Foret et al., 2021) encourages solutions with low loss in a local parameter neighborhood. Bahri et al. (2022) demonstrate that SAM improves language model generalization across language understanding and question-answering tasks. Subsequent work extends sharpness-aware fine-tuning through dynamic layer selection (Cheng et al., 2026). Building on this line of research, DAPR studies capability retention during LLM post-training through a state-dependent correction coordinated with the Adam update.

5 Conclusion

We presented **DAPR**, an efficient single-pass framework coordinating target-task learning and capability retention in language model post-training. Decoupling updates into a persistent Adam channel and a state-dependent transverse correction, DAPR substantially improves capability retention in extended SFT on 4B/8B models, with setting-dependent outcomes under DPO and larger-scale training, while language-specific centers enhance multilingual retention at zero auxiliary compute.

AI USE STATEMENT

During the preparation of this manuscript and the development of the experimental pipeline, the authors used large language models (LLMs) to assist with code debugging, parallel training orchestration, and drafting TikZ/PGFPlots visualization code. LLMs were also used to improve grammatical clarity, flag potential inaccuracies and inconsistencies in the manuscript's descriptions, and provide simulated reviewer feedback, including indicative scores and suggestions for revision. In addition to the authors' own checks, LLMs were used to review theoretical derivations for potential errors or gaps. The authors evaluated the AI-generated feedback and made all final revision decisions. All mathematical derivations, theoretical formulations, experimental configurations, and scientific interpretations were independently checked by the human authors. The authors reviewed all AI-assisted work and take full responsibility for the final content of the manuscript and accompanying artifacts.

REPRODUCIBILITY STATEMENT

Experiments were conducted on a Linux server equipped with eight NVIDIA A40 GPUs. Detailed software versions and environment configuration files will be released alongside the code. Algorithm 1 specifies the complete DAPR update rule, with the theoretical foundation, directional characterization, and remainder bounds detailed in Appendix A.1, A.3, and A.4. Section 3.1 and Appendix B provide full experimental protocols, hyperparameters, dataset splits, evaluation metrics, and compute environments. All code, training scripts, checkpoint evaluation pipelines, and configurations will be made publicly available in an open-source repository upon publication.

REFERENCES

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7360–7371, 2022.

Yifei Cheng, Xianglin Yang, Guoxia Wang, Chao Huang, Fei Ma, Dianhai Yu, Xiaochun Cao, and Li Shen. Sparse layer sharpness-aware minimization for efficient fine-tuning. *arXiv preprint arXiv:2602.09395*, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating Cross-Lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2475–2485, 2018. doi: 10.18653/v1/D18-1269. URL https://aclanthology.org/D18-1269/.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, et al. UltraFeedback: Boosting Language Models with Scaled AI Feedback. In *International Conference on Machine Learning*, pp. 9722–9744, 2024. URL https://proceedings.mlr.press/v235/cui24f.html.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing Chat Language Models by Scaling High-Quality Instructional Conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, 2023. doi: 10.18653/v1/2023.emnlp-main.183. URL https://aclanthology.org/2023.emnlp-main.183/.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.01412.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, et al. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Michael F. Hutchinson. A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. *Communications in Statistics—Simulation and Computation*, 18(3): 1059–1076, 1989. doi: 10.1080/03610918908812806. URL https://doi.org/10.1080/03610918908812806.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do We Need Zero Training Loss After Achieving Zero Training Error? In *International Conference on Machine Learning*, pp. 4604–4614, 2020. URL https://proceedings.mlr.press/v119/ishida20a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, et al. 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.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, et al. Sft doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in llms. In *International Conference on Learning Representations*, volume 2026, pp. 46954–46992, 2026.

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

Evan Ning, Wei Xue, Dong Lou, and Yike Guo. From Weights to Features: SAE-Guided Activation Regularization for LLM Continual Learning. *arXiv preprint arXiv:2606.26629*, 2026. URL https://arxiv.org/abs/2606.26629.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

Barak A. Pearlmutter. Fast Exact Multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994. doi: 10.1162/neco.1994.6.1.147. URL https://doi.org/10.1162/neco.1994.6.1.147.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023. URL https://arxiv.org/abs/2305.18290.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, et al. Aya Dataset: An Open-Access Collection for Multilingual Instruction Tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11521–11567, 2024. doi: 10.18653/v1/2024.acl-long.620. URL https://aclanthology.org/2024.acl-long.620/.

Lukas Thede, Stefan Winzeck, Zeynep Akata, and Jonathan Richard Schwarz. CapTrack: Multifaceted Evaluation of Forgetting in LLM Post-Training. *arXiv preprint arXiv:2603.06610*, 2026. URL https://arxiv.org/abs/2603.06610.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, et al. LIMA: Less Is More for Alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.11206.

A Theoretical Analysis

The analysis follows the method's construction. Appendix A.1 establishes the descent–ascent effect and its local connection to SAM. Appendix A.2 explains the obstacle posed by Adam's moment history, and Appendix A.3 characterizes the projected correction. Appendix A.4 brings these elements together in the practical two-step update. Algorithm 1 in Section 2 gives the complete algorithm.

A.1 Regularization from a Descent–Ascent Pair

We first establish the local effect that motivates DAPR, then compare its direction with SAM. Write $\boldsymbol{g}_t = \nabla\mathcal{L}(\boldsymbol{\theta}_t)$ and $\boldsymbol{H}_t = \nabla^2\mathcal{L}(\boldsymbol{\theta}_t)$ for a fixed loss.

A scalar crossing model. Inspired by the absolute-loss transformation of Ishida et al. (2020), consider the following descent-anchored scalar surrogate for a fixed differentiable loss $\mathcal{L}$, treating c_t and α_t as stop-gradient state:

$$F_t(\boldsymbol{\theta}) = \alpha_t\mathcal{L}(\boldsymbol{\theta}) + |\mathcal{L}(\boldsymbol{\theta}) - c_t| + c_t. \tag{7}$$

Its selected subgradient is $(\alpha_t + \sigma_t)\nabla\mathcal{L}$, where $\sigma_t = +1$ above the center and -1 below it; at equality we select -1 from the absolute-value subdifferential. Gradient descent on this surrogate selects task descent or partial ascent according to the loss position. This model isolates the crossing effect; the later decomposition incorporates the retained Adam channel and projected correction.

For the same fixed loss, assume an above-to-below crossing over the two steps.

Proposition 1 in Section 2.1 states the crossing-pair expansion. Its remainder satisfies $\|\mathcal{R}_t\|_2 \leq \gamma\alpha_0 L\eta^2(1+\alpha_t)\|\boldsymbol{g}_t\|_2^2 + \mathcal{O}(\eta^3\|\boldsymbol{g}_t\|_2^2)$. Under the additional non-degeneracy condition $\|\boldsymbol{H}_t\boldsymbol{g}_t\|_2 \geq m\|\boldsymbol{g}_t\|_2$ for a uniform $m > 0$, the remainder is asymptotically subordinate to the curvature term as $\|\boldsymbol{g}_t\|_2 \to 0$. The proof and this remainder comparison are given below.

Proof of Proposition 1. Both gradients below are evaluated on the same fixed loss; $\alpha_0 \in (0,1)$, $\gamma > 0$, and η is fixed over the pair. Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be twice continuously differentiable ($\mathcal{C}^2$) with gradient $\nabla\mathcal{L}(\boldsymbol{\theta})$ and Hessian $\boldsymbol{H}(\boldsymbol{\theta}) = \nabla^2\mathcal{L}(\boldsymbol{\theta})$. We make the following regularity assumptions:

(i) **L-Lipschitz Gradient (L-Smoothness):** $\|\nabla\mathcal{L}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta}')\|_2 \le L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$.

(ii) **L_H-Lipschitz Hessian:** $\|\nabla^2\mathcal{L}(\boldsymbol{\theta}) - \nabla^2\mathcal{L}(\boldsymbol{\theta}')\|_2 \le L_H\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$.

Consider a two-step crossing iteration between steps t and $t+1$:

1. At step t, the loss is above the operating center: $\mathcal{L}(\boldsymbol{\theta}_t) > c_t$. The update is:
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta(1+\alpha_t)\boldsymbol{g}_t \tag{8}$$
where $\boldsymbol{g}_t = \nabla\mathcal{L}(\boldsymbol{\theta}_t)$.
2. At step $t+1$, the loss drops below the center: $\mathcal{L}(\boldsymbol{\theta}_{t+1}) \le c_{t+1}$. The update is:
$$\boldsymbol{\theta}_{t+2} = \boldsymbol{\theta}_{t+1} + \eta(1-\alpha_{t+1})\boldsymbol{g}_{t+1} \tag{9}$$
where $\boldsymbol{g}_{t+1} = \nabla\mathcal{L}(\boldsymbol{\theta}_{t+1})$.

By Taylor's theorem with integral remainder, the gradient $\boldsymbol{g}_{t+1}$ around $\boldsymbol{\theta}_t$ is:
$$\boldsymbol{g}_{t+1} = \boldsymbol{g}_t + \boldsymbol{H}_t(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + \boldsymbol{r}_{\text{taylor}} \tag{10}$$
where $\boldsymbol{r}_{\text{taylor}} = \int_0^1 \left(\nabla^2\mathcal{L}(\boldsymbol{\theta}_t + s(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)) - \nabla^2\mathcal{L}(\boldsymbol{\theta}_t)\right)(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\, ds$. Using the L_H-Lipschitz continuity of the Hessian:
$$\|\boldsymbol{r}_{\text{taylor}}\|_2 \le \int_0^1 L_H s\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|_2^2\, ds = \frac{L_H}{2}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|_2^2 = \frac{L_H}{2}\eta^2(1+\alpha_t)^2\|\boldsymbol{g}_t\|_2^2 \tag{11}$$
Substituting $\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t = -\eta(1+\alpha_t)\boldsymbol{g}_t$, we obtain:
$$\boldsymbol{g}_{t+1} = \boldsymbol{g}_t - \eta(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t + \boldsymbol{r}_{\text{taylor}} \tag{12}$$
Substituting $\boldsymbol{g}_{t+1}$ and $\boldsymbol{\theta}_{t+1}$ into the parameter update for step $t+2$ gives:
$$\boldsymbol{\theta}_{t+2} = [\boldsymbol{\theta}_t - \eta(1+\alpha_t)\boldsymbol{g}_t] + \eta(1-\alpha_{t+1})\left[\boldsymbol{g}_t - \eta(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t + \boldsymbol{r}_{\text{taylor}}\right] \tag{13}$$
$$= \boldsymbol{\theta}_t - \eta(1+\alpha_t - 1 + \alpha_{t+1})\boldsymbol{g}_t - \eta^2(1-\alpha_{t+1})(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t + \eta(1-\alpha_{t+1})\boldsymbol{r}_{\text{taylor}} \tag{14}$$
$$= \boldsymbol{\theta}_t - \eta(\alpha_t + \alpha_{t+1})\boldsymbol{g}_t - \eta^2(1-\alpha_{t+1})(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t + \eta(1-\alpha_{t+1})\boldsymbol{r}_{\text{taylor}} \tag{15}$$
Let $\alpha_{t+1} = \alpha_t + \delta_t$. For the adaptive scale $\alpha(\boldsymbol{g}) = \frac{\alpha_0}{1+\gamma\|\boldsymbol{g}\|_2}$, the difference is bounded by:
$$|\delta_t| = |\alpha(\boldsymbol{g}_{t+1}) - \alpha(\boldsymbol{g}_t)| = \alpha_0\frac{\gamma|\|\boldsymbol{g}_{t+1}\|_2 - \|\boldsymbol{g}_t\|_2|}{(1+\gamma\|\boldsymbol{g}_{t+1}\|_2)(1+\gamma\|\boldsymbol{g}_t\|_2)} \le \alpha_0\gamma\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_2 \tag{16}$$
By L-Lipschitz continuity of the gradient, $\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_2 \le L\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|_2 = L\eta(1+\alpha_t)\|\boldsymbol{g}_t\|_2$. Therefore:
$$|\delta_t| \le \alpha_0\gamma L\eta(1+\alpha_t)\|\boldsymbol{g}_t\|_2 \tag{17}$$
Now decomposing $\alpha_{t+1} = \alpha_t + \delta_t$:
$$-\eta(\alpha_t + \alpha_{t+1})\boldsymbol{g}_t = -2\alpha_t\eta\boldsymbol{g}_t - \eta\delta_t\boldsymbol{g}_t \tag{18}$$
$$-\eta^2(1-\alpha_{t+1})(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t = -\eta^2(1-\alpha_t^2)\boldsymbol{H}_t\boldsymbol{g}_t + \eta^2\delta_t(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t \tag{19}$$
Combining these yields the exact expansion:
$$\boldsymbol{\theta}_{t+2} = \boldsymbol{\theta}_t - 2\alpha_t\eta\boldsymbol{g}_t - \eta^2(1-\alpha_t^2)\boldsymbol{H}_t\boldsymbol{g}_t + \mathcal{R}_t \tag{20}$$
where the exact remainder is:
$$\mathcal{R}_t = -\eta\delta_t\boldsymbol{g}_t + \eta^2\delta_t(1+\alpha_t)\boldsymbol{H}_t\boldsymbol{g}_t + \eta(1-\alpha_{t+1})\boldsymbol{r}_{\text{taylor}} \tag{21}$$
Using $\|\boldsymbol{H}_t\|_2 \le L$ and our previous bounds:
$$\|\mathcal{R}_t\|_2 \le \eta|\delta_t|\|\boldsymbol{g}_t\|_2 + \eta^2|\delta_t|(1+\alpha_t)L\|\boldsymbol{g}_t\|_2 + \eta(1-\alpha_{t+1})\frac{L_H}{2}\eta^2(1+\alpha_t)^2\|\boldsymbol{g}_t\|_2^2 \tag{22}$$
$$\le \alpha_0\gamma L\eta^2(1+\alpha_t)\|\boldsymbol{g}_t\|_2^2 + \alpha_0\gamma L^2\eta^3(1+\alpha_t)^2\|\boldsymbol{g}_t\|_2^2 + \frac{L_H}{2}\eta^3(1-\alpha_{t+1})(1+\alpha_t)^2\|\boldsymbol{g}_t\|_2^2 \tag{23}$$
$$= \mathcal{O}\left(\eta^2\|\boldsymbol{g}_t\|_2^2 + \eta^3\|\boldsymbol{g}_t\|_2^2\right) \tag{24}$$
Noting that $\boldsymbol{H}_t\boldsymbol{g}_t = \nabla_{\boldsymbol{\theta}}\left(\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_2^2\right)\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$, we obtain the effective update:
$$\boldsymbol{\theta}_{t+2} = \boldsymbol{\theta}_t - 2\alpha_t\eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t) - \eta^2(1-\alpha_t^2)\,\nabla_{\boldsymbol{\theta}}\left(\tfrac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_2^2\right)\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} + \mathcal{R}_t \tag{25}$$

When the curvature term dominates the remainder. Assume the trajectory remains uniformly non-degenerate: $\|\boldsymbol{H}_t\boldsymbol{g}_t\|_2 \geq m\|\boldsymbol{g}_t\|_2$ for some $m > 0$. Then the implicit curvature term has norm $\Omega(\eta^2\|\boldsymbol{g}_t\|_2)$. In contrast, the adaptive-coefficient disturbance is bounded by $\alpha_0\gamma L\eta^2(1+\alpha_t)\|\boldsymbol{g}_t\|_2^2 = \mathcal{O}(\eta^2\|\boldsymbol{g}_t\|_2^2)$. Along a sequence satisfying the stated uniform bounds with $\|\boldsymbol{g}_t\|_2 \to 0$, the quadratic bound vanishes faster than the linear lower bound. Thus, for fixed bounded η:

$$\lim_{\|\boldsymbol{g}_t\|_2\to 0} \frac{\|\mathcal{R}_t\|_2}{\eta^2(1-\alpha_t^2)\|\boldsymbol{H}_t\boldsymbol{g}_t\|_2} = 0 \tag{26}$$

Under the stated non-degeneracy condition, the adaptive-coefficient contribution is locally higher order than the curvature term in this two-step expansion. ■

Local expansion of SAM. For a fixed loss $\mathcal{L}$, write $\boldsymbol{g}_t = \nabla\mathcal{L}(\boldsymbol{\theta}_t)$ and $\boldsymbol{H}_t = \nabla^2\mathcal{L}(\boldsymbol{\theta}_t)$. Standard ℓ_2 SAM minimizes the largest loss in a parameter neighborhood of radius ρ (Foret et al., 2021). If the gradient is L-Lipschitz in this neighborhood, Taylor's theorem gives

$$\max_{\|\boldsymbol{h}\|_2\leq\rho} \mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{h}) = \mathcal{L}(\boldsymbol{\theta}_t) + \rho\|\boldsymbol{g}_t\|_2 + \mathcal{O}(\rho^2). \tag{27}$$

Indeed, the remainder in the linear expansion of the loss is bounded uniformly by $L\rho^2/2$, and the maximum of $\boldsymbol{g}_t^\top\boldsymbol{h}$ over the ball is $\rho\|\boldsymbol{g}_t\|_2$. Thus the first-order neighborhood objective adds a gradient-norm penalty.

For $\boldsymbol{g}_t \neq \boldsymbol{0}$, SAM uses the perturbation $\rho\boldsymbol{g}_t/\|\boldsymbol{g}_t\|_2$. With an L_H-Lipschitz Hessian, a separate Taylor expansion of the gradient gives

$$\nabla\mathcal{L}\left(\boldsymbol{\theta}_t + \rho\frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|_2}\right) = \boldsymbol{g}_t + \rho\frac{\boldsymbol{H}_t\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|_2} + \mathcal{O}(\rho^2), \tag{28}$$

where the remainder has norm at most $L_H\rho^2/2$. This is the gradient evaluated at SAM's perturbed point, with the perturbation treated as fixed during that evaluation. The chain rule identifies the added direction:

$$\nabla_{\boldsymbol{\theta}}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_2|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} = \frac{\boldsymbol{H}_t\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|_2}. \tag{29}$$

These expansions describe SAM's local objective and perturbation direction.

Relation to SAM. The correction in Proposition 1 is proportional to $-\boldsymbol{H}_t\boldsymbol{g}_t$, whereas the additional descent direction in SAM's expansion, Eq. 28, is proportional to $-\boldsymbol{H}_t\boldsymbol{g}_t/\|\boldsymbol{g}_t\|_2$. For a nonzero gradient, both therefore act along the negative gradient of a gradient-norm penalty, with different scaling: SAM uses $\|\nabla\mathcal{L}\|_2$ at first order in ρ, while the crossing model yields $\frac{1}{2}\|\nabla\mathcal{L}\|_2^2$ at second order in η. The latter term arises from gradients at two successive iterates, without an additional perturbed evaluation within either step. This shared local direction motivates the intervention. Appendix A.4 analyzes the additional contributions of Adam and projection in practical DAPR.

A.2 Sign Switching and Adam's Moment History

The crossing effect depends on the realized update directions. We next show why sending the scalar model's signed gradient through Adam need not realize those directions, motivating a correction outside moment estimation.

For a threshold-based signed-gradient update, let $\sigma_t = +1$ when $L_t > c_t$ and $\sigma_t = -1$ otherwise, and feed $\sigma_t\boldsymbol{g}_t$ into Adam. A superscript F distinguishes the states of this direct combination:

$$\begin{aligned}\boldsymbol{m}_t^{\mathrm{F}} &= \beta_1\boldsymbol{m}_{t-1}^{\mathrm{F}} + (1-\beta_1)\sigma_t\boldsymbol{g}_t,\\ \boldsymbol{v}_t^{\mathrm{F}} &= \beta_2\boldsymbol{v}_{t-1}^{\mathrm{F}} + (1-\beta_2)(\sigma_t\boldsymbol{g}_t)^2 = \beta_2\boldsymbol{v}_{t-1}^{\mathrm{F}} + (1-\beta_2)\boldsymbol{g}_t^2.\end{aligned}$$

A pure sign reversal changes the first-moment input but not the squared-gradient input at the same parameter point. States along separate trajectories need not coincide, since later parameters and gradients differ. From zero initialization,

$$\boldsymbol{m}_t^{\mathrm{F}} = (1-\beta_1)\sum_{k=1}^{t}\beta_1^{t-k}\sigma_k\boldsymbol{g}_k,$$

so the direction depends on the history of loss-dependent signs rather than the current sign alone.

Write $\widehat{\boldsymbol{m}}_t^{\mathrm{F}} = \boldsymbol{m}_t^{\mathrm{F}}/(1-\beta_1^t)$ and $\boldsymbol{P}_t^{\mathrm{F}} = \mathrm{diag}((\sqrt{\widehat{\boldsymbol{v}}_t^{\mathrm{F}}} + \epsilon\mathbf{1})^{-1})$. Consider the adaptive-gradient part of the update, with $0 \le \beta_1 < 1$ and $\eta > 0$. Below the threshold, $\sigma_t = -1$, and its first-order change in the task loss is

$$\boldsymbol{g}_t^\top(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) = -\frac{\eta}{1-\beta_1^t}\Big[\beta_1 \boldsymbol{g}_t^\top \boldsymbol{P}_t^{\mathrm{F}} \boldsymbol{m}_{t-1}^{\mathrm{F}} \quad -(1-\beta_1)\boldsymbol{g}_t^\top \boldsymbol{P}_t^{\mathrm{F}} \boldsymbol{g}_t\Big].$$

If

$$\beta_1 \boldsymbol{g}_t^\top \boldsymbol{P}_t^{\mathrm{F}} \boldsymbol{m}_{t-1}^{\mathrm{F}} > (1-\beta_1)\boldsymbol{g}_t^\top \boldsymbol{P}_t^{\mathrm{F}} \boldsymbol{g}_t,$$

the directional derivative is negative: historical momentum makes the adaptive update a first-order descent direction even though the current sign requests ascent. The reversed gradient then enters the moment history and affects later updates. AdamW adds a separate decoupled weight-decay displacement; this does not remove the coupling in the first-moment recurrence.

If the scalar surrogate in Appendix A.1 is passed directly to Adam, the input is $(\alpha_t + \sigma_t)\boldsymbol{g}_t$ and

$$\boldsymbol{m}_t^{\mathrm{F}} = \beta_1 \boldsymbol{m}_{t-1}^{\mathrm{F}} + (1-\beta_1)(\alpha_t + \sigma_t)\boldsymbol{g}_t.$$

Below the center, the input is $-(1-\alpha_t)\boldsymbol{g}_t$, but the actual direction still depends on historical momentum. The squared-gradient input is also scaled by $(\alpha_t + \sigma_t)^2$, so the pure-sign invariance of the second-moment input no longer applies.

Proposition 1 requires an actual descent–partial-ascent pair. The recurrences above show why reversing Adam's input does not guarantee that pair, so its regularization interpretation cannot be transferred directly to this signed-gradient Adam update. Historical momentum can therefore weaken or delay the intended intervention. DAPR keeps both moment estimators driven by task gradients and applies the correction afterward, preventing its sign from being directly accumulated. Its retained Adam channel and projection also change the scalar recursion; Appendix A.4 gives the corresponding practical decomposition.

A.3 Geometry of the Practical Correction

Separating the correction from the moment inputs leaves a second question: how should an ascent correction coexist with the current Adam direction? Proposition 2 characterizes the projection used for this purpose, separately for each trainable tensor. Fix a block and suppress (j) in the proof: $\boldsymbol{P} \succ 0$, $\boldsymbol{g}$, and $\boldsymbol{u}$ are its current preconditioner, gradient, and Adam direction. Let $\epsilon_{\mathrm{proj}} \ge 0$ denote the projection stabilizer only; the numerical stabilizer in $\boldsymbol{P}$ remains positive. Put $a = \boldsymbol{g}^\top \boldsymbol{P}\boldsymbol{u}$, $b = \boldsymbol{u}^\top \boldsymbol{P}\boldsymbol{u}$, and

$$\boldsymbol{q} = \boldsymbol{P}\boldsymbol{g} - \frac{a}{b + \epsilon_{\mathrm{proj}}}\boldsymbol{P}\boldsymbol{u}.$$

For exact projection assume $\boldsymbol{u} \neq 0$; when $\boldsymbol{u} = 0$, set $\boldsymbol{q} = \boldsymbol{P}\boldsymbol{g}$.

Proof. Substitution gives $\boldsymbol{u}^\top \boldsymbol{q} = \epsilon_{\mathrm{proj}} a/(b + \epsilon_{\mathrm{proj}})$ and $\boldsymbol{g}^\top \boldsymbol{q} = \boldsymbol{g}^\top \boldsymbol{P}\boldsymbol{g} - a^2/(b + \epsilon_{\mathrm{proj}}) \ge 0$, where the inequality follows from weighted Cauchy–Schwarz. For exact projection, stationarity of $\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{P}\boldsymbol{g}\|^2_{\boldsymbol{P}^{-1}}$ subject to $\boldsymbol{u}^\top \boldsymbol{z} = 0$ gives $\boldsymbol{z} = \boldsymbol{P}\boldsymbol{g} - \nu \boldsymbol{P}\boldsymbol{u}$ and $\nu = a/b$. With positive stabilization, the objective is

$$\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{P}\boldsymbol{g}\|^2_{\boldsymbol{P}^{-1}} + \frac{(\boldsymbol{u}^\top \boldsymbol{z})^2}{2\epsilon_{\mathrm{proj}}}.$$

Its stationarity equation gives the same $\boldsymbol{q}$. Positive definiteness of $\boldsymbol{P}^{-1}$ ensures uniqueness. □

For the concatenated parameters, the exact projection therefore solves the separable problem

$$\min_{\{\boldsymbol{z}^{(j)}\}} \sum_j \frac{1}{2}\|\boldsymbol{z}^{(j)} - \boldsymbol{P}^{(j)}\boldsymbol{g}^{(j)}\|^2_{(\boldsymbol{P}^{(j)})^{-1}} \quad \text{subject to} \quad (\boldsymbol{u}^{(j)})^\top \boldsymbol{z}^{(j)} = 0 \quad \text{for every } j.$$

There is one constraint per block. A single constraint on the sum of block inner products would allow cancellation between blocks and generally yield a different projection. Positive stabilization replaces each constraint with its own quadratic penalty.

Using the block-diagonal coefficient $\boldsymbol{A}_t$ from Lemma 1, exact block projection implies

$$\boldsymbol{u}_t^\top \boldsymbol{A}_t \boldsymbol{q}_t = \sum_j \alpha_t^{(j)} (\boldsymbol{u}_t^{(j)})^\top \boldsymbol{q}_t^{(j)} = 0, \qquad \boldsymbol{g}_t^\top \boldsymbol{A}_t \boldsymbol{q}_t \ge 0.$$

With positive stabilization, the first expression equals $\sum_j \alpha_t^{(j)} \epsilon_{\text{proj}} a_t^{(j)} / (b_t^{(j)} + \epsilon_{\text{proj}})$. For the below-center update, with $\boldsymbol{r} = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$,

$$\boldsymbol{g}_t^\top (\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) = -\eta \boldsymbol{g}_t^\top \boldsymbol{u}_t + \eta \boldsymbol{g}_t^\top \boldsymbol{A}_t \boldsymbol{q}_t - \eta \lambda_{\text{base}} \boldsymbol{g}_t^\top \boldsymbol{r}.$$

The projected correction contributes nonnegatively to the first-order loss change, while the full update combines this contribution with Adam and anchoring. With exact projection, a block correction vanishes when its gradient is parallel to its nonzero Adam direction; positive stabilization generally leaves a residual.

A.4 The Crossing Effect in the Practical DAPR Update

We now return to the motivating two-step effect with Adam, projection, adaptive strengths, and anchoring included. Lemma 1 gives the full local decomposition; Corollary 1 then identifies conditions for a gradient-norm interpretation. We use the lemma's concatenated notation. The matrices $\boldsymbol{P}$, $\boldsymbol{Q}_t$, and $\boldsymbol{A}_t$ are block diagonal, but the full loss Hessian $\boldsymbol{H}_t$ need not be. The coefficient $\boldsymbol{A}_t$ contains the tensor-specific adaptive strengths. The analysis fixes the preconditioner and step size over a crossing pair and evaluates both gradients on the same loss; Adam directions, coefficients, and projection subspaces may change.

Proof. For $L_t > c_t$, set $\boldsymbol{d}_t = \boldsymbol{u}_t + \boldsymbol{A}_t \boldsymbol{P} \boldsymbol{g}_t$ and update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{d}_t$. For $L_{t+1} \le c_{t+1}$,

$$\boldsymbol{\theta}_{t+2} = \boldsymbol{\theta}_{t+1} - \eta \boldsymbol{u}_{t+1} + \eta \boldsymbol{A}_{t+1} \boldsymbol{P} \boldsymbol{Q}_{t+1} \boldsymbol{g}_{t+1} - \eta \lambda_{\text{base}} (\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_0).$$

Taylor's theorem for the full gradient gives

$$\boldsymbol{g}_{t+1} = \boldsymbol{g}_t - \eta \boldsymbol{H}_t \boldsymbol{d}_t + \boldsymbol{r}_t, \qquad \|\boldsymbol{r}_t\|_2 \le \frac{L_H}{2} \eta^2 \|\boldsymbol{d}_t\|_2^2. \tag{30}$$

In block j, the Hessian contribution is $\sum_k \boldsymbol{H}_t^{(jk)} \boldsymbol{d}_t^{(k)}$, so this expansion retains cross-block interactions. Multiplying by $\boldsymbol{Q}_{t+1} = \boldsymbol{Q}_t + \Delta \boldsymbol{Q}_t$ yields

$$\begin{aligned} \boldsymbol{g}_{\perp,t+1} = \boldsymbol{g}_{\perp,t} &- \eta \boldsymbol{Q}_t \boldsymbol{H}_t \boldsymbol{d}_t + \Delta \boldsymbol{Q}_t \boldsymbol{g}_t \\ &- \eta \Delta \boldsymbol{Q}_t \boldsymbol{H}_t \boldsymbol{d}_t + (\boldsymbol{Q}_t + \Delta \boldsymbol{Q}_t) \boldsymbol{r}_t. \end{aligned} \tag{31}$$

The first-order corrections combine as

$$-\eta \boldsymbol{A}_t \boldsymbol{P} \boldsymbol{g}_t + \eta \boldsymbol{A}_{t+1} \boldsymbol{P} \boldsymbol{g}_{\perp,t} = -\eta \boldsymbol{A}_t \boldsymbol{P} \boldsymbol{g}_{\parallel,t} + \eta \Delta \boldsymbol{A}_t \boldsymbol{P} \boldsymbol{g}_{\perp,t}.$$

Substitution proves the identity in Lemma 1, with

$$\widetilde{\mathcal{R}}_t = \eta \boldsymbol{A}_{t+1} \boldsymbol{P} \left[-\eta \Delta \boldsymbol{Q}_t \boldsymbol{H}_t \boldsymbol{d}_t + (\boldsymbol{Q}_t + \Delta \boldsymbol{Q}_t) \boldsymbol{r}_t \right]. \tag{32}$$

Consequently,

$$\|\widetilde{\mathcal{R}}_t\|_2 \le \eta^2 \|\boldsymbol{A}_{t+1} \boldsymbol{P}\|_2 \|\Delta \boldsymbol{Q}_t\|_2 \|\boldsymbol{H}_t\|_2 \|\boldsymbol{d}_t\|_2 + \frac{L_H}{2} \eta^3 \|\boldsymbol{A}_{t+1} \boldsymbol{P}\|_2 \|\boldsymbol{Q}_{t+1}\|_2 \|\boldsymbol{d}_t\|_2^2.$$

All products retain their displayed order; in particular, $\boldsymbol{A}_t$ need not commute with $\boldsymbol{H}_t$. □

With the other factors bounded, the remainder is $O(\eta^2 \|\Delta \boldsymbol{Q}_t\|_2 + \eta^3)$, and becomes $O(\eta^3)$ when $\|\Delta \boldsymbol{Q}_t\|_2 = O(\eta)$.

The term $\Delta \boldsymbol{A}_t$ captures changes in the block-specific adaptive strengths. When all blocks share the same coefficient, $\boldsymbol{A}_t = \alpha_t \boldsymbol{I}$, recovering the scalar-coefficient decomposition.

Corollary 1 (Conditional Orthogonal Gradient-Norm Interpretation). In addition to the lemma's assumptions, suppose exact projection is used and, over the pair, (i) $\boldsymbol{A}_{t+1} = \boldsymbol{A}_t = \boldsymbol{A}$ and $\boldsymbol{Q}_{t+1} = \boldsymbol{Q}_t = \boldsymbol{Q}$, (ii) $\boldsymbol{g}_{\parallel,t} = 0$, (iii) $\boldsymbol{Q} \boldsymbol{H}_t \boldsymbol{u}_t = 0$, and (iv) $\boldsymbol{Q} \boldsymbol{H}_t \boldsymbol{A} \boldsymbol{P} \boldsymbol{g}_{\perp,t} = \boldsymbol{H}_t \boldsymbol{A} \boldsymbol{P} \boldsymbol{g}_{\perp,t}$. Apart from Adam and anchoring, the leading correction is

$$-\eta^2 \boldsymbol{A} \boldsymbol{P} \boldsymbol{H}_t \boldsymbol{A} \boldsymbol{P} \boldsymbol{g}_{\perp,t}.$$

Treat $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{Q}$ as locally fixed and define

$$\Phi(\boldsymbol{\theta}) = \frac{1}{2} (\boldsymbol{Q} \nabla \mathcal{L}(\boldsymbol{\theta}))^\top \boldsymbol{A} \boldsymbol{P} (\boldsymbol{Q} \nabla \mathcal{L}(\boldsymbol{\theta})).$$

Exact block projection gives $\boldsymbol{Q}^2 = \boldsymbol{Q}$ and $\boldsymbol{Q}^\top \boldsymbol{P} = \boldsymbol{P}\boldsymbol{Q}$. Since $\boldsymbol{A}$ is scalar within each block, it commutes with $\boldsymbol{P}$ and $\boldsymbol{Q}$. Hence

$$\nabla\Phi(\boldsymbol{\theta}_t) = \boldsymbol{H}_t \boldsymbol{Q}^\top \boldsymbol{A}\boldsymbol{P}\boldsymbol{Q}\boldsymbol{g}_t = \boldsymbol{H}_t \boldsymbol{A}\boldsymbol{P}\boldsymbol{g}_{\perp,t}.$$

The correction is thus $-\eta^2 \boldsymbol{A}\boldsymbol{P}\nabla\Phi(\boldsymbol{\theta}_t)$. Since $\boldsymbol{A}\boldsymbol{P} \succ 0$ for the positive block strengths used here,

$$\nabla\Phi(\boldsymbol{\theta}_t)^\top \left[-\eta^2 \boldsymbol{A}\boldsymbol{P}\nabla\Phi(\boldsymbol{\theta}_t)\right] = -\eta^2 \nabla\Phi(\boldsymbol{\theta}_t)^\top \boldsymbol{A}\boldsymbol{P}\nabla\Phi(\boldsymbol{\theta}_t) \leq 0.$$

It is therefore a first-order descent direction for this frozen-state weighted transverse gradient norm, strictly so when $\nabla\Phi(\boldsymbol{\theta}_t) \neq \mathbf{0}$. The potential weights differ between blocks. When $\boldsymbol{A} = \alpha \boldsymbol{I}$, its correction reduces to $-\eta^2\alpha^2 \boldsymbol{P}\boldsymbol{H}_t\boldsymbol{P}\boldsymbol{g}_{\perp,t}$.

Under conditions (i)–(iv), the exact-projection case therefore admits a frozen-state gradient-norm interpretation. Lemma 1 describes the general update, including stabilized projection, changing coefficients, and projection rotation. The reported branch-switch milestones (Table 5) are consistent with the crossing scenario analyzed here; the gradient-norm interpretation remains conditional on the assumptions above.

B Experimental Details and Additional Results

This appendix provides protocols and supplementary evidence for Sections 3.2–3.4: checkpoint trajectories and optimizer diagnostics, component and hyperparameter analyses, and additional training settings. Trajectory plots supplement the main-text endpoints; additional tables report results not shown in the main text.

B.1 Data and Evaluation

Table 6 records the training data and evaluation benchmarks. The evaluation sources are ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), GSM8K (Cobbe et al., 2021), and XNLI (Conneau et al., 2018); the table specifies the subsets and metrics used here.

Training data

Dataset	Recorded split / subset	Reported examples
UltraChat-200k	`train_sft`	207,865
UltraFeedback-Binarized	`train_prefs`	61,135 pairs
Aya	`train`; `zh`, `de`, `es`, `fr`	100,000
LIMA	`train`	1,000

Evaluation inventory

Benchmark	Reported metric	Shots	Evaluation size
ARC Challenge	Normalized accuracy	25	1,172
MMLU	Macro-average accuracy	5	14,042
HellaSwag	Accuracy	10	10,042
GSM8K	Strict CoT exact match	5	1,319
XNLI-fr	Three-way accuracy	0	2,490
Aya target languages	Instruction-following accuracy	0	2,500/language

Table 6: Reported datasets and evaluation settings. Counts reflect evaluation splits from standard lm-evaluation-harness benchmarks.

Training and optimization hyperparameters. All post-training fine-tuning uses PEFT LoRA with rank $r = 16$, scaling factor $\alpha = 32$, dropout 0.05, and zero bias, targeting attention query and value linear projections (W_q, W_v). Standard base optimizers use AdamW with $\beta_1 = 0.9, \beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay 0.01, scheduled via cosine decay with 5% linear warmup. For SFT on UltraChat, learning rates are $\eta = 2 \times 10^{-4}$ for Gemma-3-4B, $\eta = 1 \times 10^{-4}$ for Llama-3-8B, and $\eta = 2 \times 10^{-5}$ for Gemma-3-12B and Qwen-2.5-14B. For standard DPO on UltraFeedback,

$\beta = 0.10$ and $\eta = 5 \times 10^{-6}$ ($\eta = 2 \times 10^{-5}$ on 12B/14B), while the stress-testing configuration jointly sets $\beta = 0.01$ and $\eta = 2 \times 10^{-5}$. The per-device batch size and gradient accumulation yield an effective batch size of 8 with max sequence length 2,048. SAM baselines evaluate perturbation radius $\rho = 0.05$. DAPR uses default hyperparameters: baseline anchor $\alpha_0 = 0.5$, offset multiplier $\lambda = 1.0$, smoothing factors $\tau = 0.1, \tau_s = 0.1$, and base anchoring weight $\lambda_{\text{base}} = 0.01$.

B.2 Extended Results and Diagnostics

B.2.1 Capability Trajectories under SFT and Stress DPO

Figure 1 in Section 3.2 provides the complete SFT (UltraChat) and stress-DPO (UltraFeedback) trajectories across all four model configurations (Gemma-3-4B, Llama-3-8B, Gemma-3-12B, and Qwen-2.5-14B). Error bars at 10K represent empirical standard deviations over three random seeds for 4B and 8B models. Tabular comparisons across all models and intermediate milestones are reported in Table 2.

B.2.2 DPO Stress-Test Trajectories and Parameter Displacement

Under the aggressive DPO stress configuration ($\beta = 0.01, \eta = 2 \times 10^{-5}$), Table 7 tracks the multi-milestone capability progression across ARC Challenge, MMLU, HellaSwag, and GSM8K, alongside cumulative parameter displacement $\|\Delta W\|_F$, evaluated at 2K, 4K, 6K, 8K, and 10K training steps on Gemma-3-4B and Llama-3-8B. Continuous trajectory curves across all benchmarks are presented in Figure 1 in the main text.

B.2.3 Curvature Diagnostics and Estimator Convention

Local loss-landscape geometry is evaluated in the trainable LoRA parameter subspace across scales, as summarized in Table 1 (top eigenvalue $\lambda_{\max}$, trace $\mathrm{Tr}(\boldsymbol{H})$, gradient norm, and relative runtime). Furthermore, Figure 2 compares the final training and held-out validation cross-entropy loss on UltraChat at 10K steps, confirming that DAPR's capability retention coexists with lower held-out loss (0.431 vs. 0.449 on Gemma-3-4B; 0.526 vs. 0.570 on Llama-3-8B).

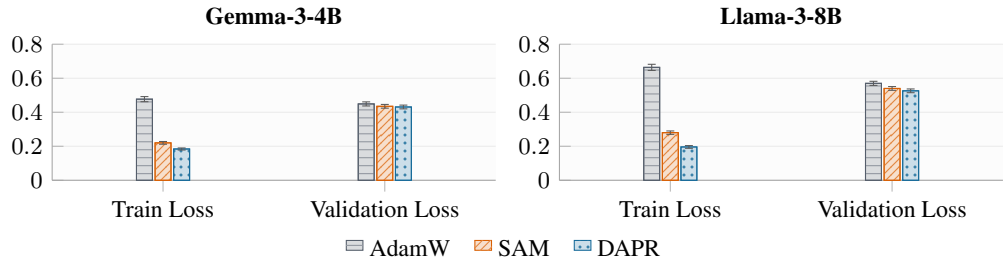

Figure 2: **UltraChat SFT training and validation cross-entropy loss at 10K steps** (mean ± SD across three seeds; lower is better) for Gemma-3-4B and Llama-3-8B.

For a fixed validation loss $\mathcal{L}$ on the trainable parameter vector θ, a Hessian–vector product can be evaluated without materializing the Hessian (Pearlmutter, 1994):

$$Hv = \nabla_\theta \Big(\nabla_\theta \mathcal{L}(\theta)^\top v \Big), \qquad H = \nabla_\theta^2 \mathcal{L}(\theta). \tag{33}$$

For an indefinite Hessian, the eigenvalue of largest magnitude need not be the largest algebraic eigenvalue. In particular, a norm-based power-iteration estimate alone does not determine the eigenvalue sign. The signed Rayleigh quotient is $v^\top Hv/(v^\top v)$. In our measurements, the dominant spectral radius and dominant curvature direction $\boldsymbol{v}_1$ are extracted via $K = 50$ iterations of normalized power iteration initialized from a random Gaussian probe $\boldsymbol{v}^{(0)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, terminating when the relative change in the eigenvalue estimate satisfies $|\lambda^{(k)} - \lambda^{(k-1)}|/\lambda^{(k)} < 10^{-4}$; the signed Rayleigh quotient is then evaluated to confirm non-negativity ($\lambda_{\max} \geq 0$). For independent Rademacher probes $\boldsymbol{z}_m \in \{-1, +1\}^d$, Hutchinson's stochastic trace estimator satisfies (Hutchinson, 1989):

$$\widehat{\mathrm{Tr}}(\boldsymbol{H}) = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{z}_m^\top \boldsymbol{H} \boldsymbol{z}_m, \qquad \mathbb{E}[\widehat{\mathrm{Tr}}(\boldsymbol{H})] = \mathrm{Tr}(\boldsymbol{H}). \tag{34}$$

Model	Metric	Optimizer	Milestone Step				
			2,000	4,000	6,000	8,000	10,000
Gemma-3-4B ($\beta = 0.01$)	ARC Challenge (%)	AdamW	**58.28**	**59.98**	**60.58**	55.97	56.66
		SAM	54.86	55.97	58.36	58.79	59.98
		DAPR	58.11	**59.98**	59.90	**60.24**	**60.49**
	GSM8K flex (%)	AdamW	**48.75**	**49.96**	**49.28**	41.17	44.73
		SAM	38.89	43.06	47.69	**50.64**	**50.11**
		DAPR	46.25	47.99	47.99	49.81	49.36
	MMLU (%)	AdamW	58.31	**58.52**	**58.48**	58.23	58.50
		SAM	58.05	58.11	58.45	**58.54**	**58.52**
		DAPR	**58.48**	58.35	58.42	58.29	58.21
	HellaSwag (%)	AdamW	**78.89**	79.73	79.96	76.90	77.60
		SAM	76.27	76.79	78.43	78.97	80.20
		DAPR	78.85	**79.80**	**80.16**	**80.35**	**80.40**
	$\|\Delta W\|_F$	AdamW	2.095	2.586	2.874	0.721	0.911
		SAM	2.030	3.505	5.237	6.363	7.385
		DAPR	2.798	3.691	4.146	4.342	4.409
Llama-3-8B ($\beta = 0.01$)	ARC Challenge (%)	AdamW	**56.23**	58.45	**58.28**	58.11	58.62
		SAM	54.01	55.46	56.66	**58.36**	**60.24**
		DAPR	**56.23**	**59.22**	57.59	58.11	57.76
	GSM8K flex (%)	AdamW	**56.41**	52.31	52.62	**51.18**	**54.59**
		SAM	51.86	**53.68**	54.59	50.95	50.64
		DAPR	54.59	47.69	**54.81**	49.36	50.80
	MMLU (%)	AdamW	**63.20**	62.85	63.15	**63.16**	63.00
		SAM	62.26	62.68	62.95	63.11	**63.11**
		DAPR	62.93	**63.15**	**63.16**	62.68	62.54
	HellaSwag (%)	AdamW	**82.46**	**83.08**	**83.36**	83.43	83.35
		SAM	79.86	81.10	82.43	**83.87**	**84.89**
		DAPR	82.23	82.95	83.12	83.43	83.44
	$\|\Delta W\|_F$	AdamW	1.749	2.410	2.764	2.912	2.932
		SAM	2.449	4.316	5.835	7.422	8.694
		DAPR	2.606	3.905	4.582	4.933	4.986

Table 7: **Multi-milestone DPO stress trajectories and parameter displacement** under $\beta = 0.01, \eta = 2 \times 10^{-5}$ on UltraFeedback across benchmarks and Frobenius parameter displacement $\|\Delta W\|_F$. Bold marks the highest capability score per milestone; parameter displacement is unranked. DAPR preserves stability across milestones while tracking two-pass SAM.

We set $M = 100$ independent Rademacher probe vectors. All curvature diagnostics are evaluated strictly over the trainable LoRA parameter subspace $\boldsymbol{\theta}$ (with pre-trained base parameters frozen) on the cross-entropy loss across 500 fixed mini-batches sampled from the UltraChat validation split. Hessian–vector products $\boldsymbol{Hv}$ are accumulated over the 500 batches before performing power iterations and trace probes, avoiding the invalid approximation of averaging per-batch dominant eigenvalues. Reported error bars (±SD) capture variation across 3 independent random evaluation seeds for probe initialization and validation subsampling.

Milestone	Optimizer	Max Logit ($z_{\max}$)	Logit Std (σ_z)	Saturation ($\lvert z\rvert > 30$)	Sequence Margin ($\Delta \log \pi$)	Chosen PPL (PPL_w)
Panel A: Gemma-3-4B (Quadruple RMSNorm sandwich, 4B parameters)						
Base (Step 0)	—	34.53	3.95	$< 0.01\%$	-27.44	3.94
Step 2,000	AdamW	185.42	7.94	0.48%	-15.01	21.80
	SAM	37.51	4.11	$< 0.01\%$	-27.88	3.94
	DAPR	156.42	8.20	0.83%	-11.17	28.37
Step 4,000	AdamW	192.92	8.48	0.83%	-7.95	22.10
	SAM	52.70	4.51	$< 0.01\%$	-27.16	4.23
	DAPR	202.36	8.54	0.99%	-12.51	19.83
Step 6,000	AdamW	**218.06**	**8.83**	**1.11%**	$+16.21$	47.88
	SAM	127.58	6.20	0.08%	-18.63	6.98
	DAPR	207.82	8.38	0.88%	$+2.18$	24.01
Step 8,000 *(Collapse)*	AdamW	**37.77**	**4.04**	$< 0.01\%$	$\mathbf{-31.16}$	**3.90**
	SAM	155.59	6.20	0.10%	-8.75	6.25
	DAPR	219.74	8.24	0.90%	$+0.76$	21.83
Step 10,000	AdamW	43.39	4.24	$< 0.01\%$	-34.44	4.67
	SAM	178.99	6.57	0.17%	$+0.82$	7.36
	DAPR	222.08	8.38	0.96%	$+3.91$	25.50
Panel B: Llama-3-8B (Standard pre-layer normalization, 8B parameters)						
Base (Step 0)	—	30.44	2.56	$< 0.01\%$	-31.60	5.16
Step 2,000	AdamW	40.58	2.42	$< 0.01\%$	$+8.10$	31.88
	SAM	30.54	2.58	$< 0.01\%$	-30.58	5.49
	DAPR	38.07	2.43	$< 0.01\%$	$+1.52$	24.52
Step 4,000	AdamW	42.37	2.38	$< 0.01\%$	$+13.42$	59.48
	SAM	32.33	2.56	$< 0.01\%$	-26.77	5.02
	DAPR	41.67	2.39	$< 0.01\%$	-2.55	59.18
Step 6,000	AdamW	42.78	2.43	$< 0.01\%$	$+61.41$	60.97
	SAM	35.03	2.52	$< 0.01\%$	-17.91	4.96
	DAPR	43.15	2.42	$< 0.01\%$	$+40.74$	27.13
Step 8,000	AdamW	44.29	2.43	$< 0.01\%$	$+79.52$	101.61
	SAM	36.42	2.47	$< 0.01\%$	-8.21	8.42
	DAPR	44.85	2.48	$< 0.01\%$	$+52.80$	67.25
Step 10,000	AdamW	44.00	2.42	$< 0.01\%$	$+85.01$	100.49
	SAM	37.09	2.45	$< 0.01\%$	-3.80	7.90
	DAPR	45.81	2.53	$< 0.01\%$	$+73.52$	99.20

Table 8: **Optimization diagnostics under weakened reference anchoring** ($\beta = 0.01, \eta = 2 \times 10^{-5}$) on UltraFeedback validation pairs. Metrics report maximum logit magnitude ($z_{\max}$), vocabulary dispersion (σ_z), saturation fraction ($\lvert z\rvert > 30$), sequence reward margin ($\Delta \log \pi$), and chosen perplexity (PPL_w).

Top eigenvalue and trace measure different aspects of curvature. On Gemma-3-4B, DAPR lowers the reported top eigenvalue from 437.2 to 312.8, while the trace rises from 415 to 1,059 (Table 1). The top eigenvalue measures the strongest positive curvature direction, whereas the trace sums all Hessian eigenvalues. The two statistics therefore describe complementary aspects of the measured geometry. SAM uses an adversarial perturbation in an ℓ_2 neighborhood, while DAPR coordinates its correction with the current Adam direction. Proposition 2 characterizes this direction,

and Corollary 1 gives its conditional two-step interpretation. The reported spectra provide empirical diagnostics in the trainable LoRA parameterization.

B.2.4 Logit and Preference-Margin Diagnostics

Table 8 reports checkpoint-level optimization diagnostics for AdamW, SAM, and DAPR on Gemma-3-4B and Llama-3-8B. Under the joint DPO stress setting ($\beta = 0.01$, $\eta = 2 \times 10^{-5}$), we track token logit scales, the fraction with $|z| > 30$, sequence log-probability margins ($\Delta \log \pi_\theta$), and policy perplexities on UltraFeedback validation pairs at recorded training milestones. The reported margin is the chosen-minus-rejected sequence log-probability difference under the trained policy. Logit magnitudes summarize output scale, while margins and perplexities describe preference ordering and response likelihood. We summarize the key observations below.

Gemma-3-4B. AdamW's maximum logit rises from 34.53 at initialization to 218.06 at 6K, where $\sigma_z = 8.83$ and 1.11% of logits exceed $|z| > 30$. At 8K, its maximum logit falls to 37.77 and the sequence margin changes from $+16.21$ at 6K to -31.16, coinciding with the capability drop in Figure 1. At 10K, DAPR and SAM maintain positive sequence margins of $+3.91$ and $+0.82$, respectively, whereas AdamW's margin is -34.44. Their chosen-response perplexities also differ (25.50, 7.36, and 4.67), so retained capability should be read alongside the preference-task diagnostics.

Llama-3-8B. The logit scale remains substantially smaller: the reported maximum is at most 45.81 across methods and milestones, and fewer than 0.01% of logits exceed $|z| > 30$. At 10K, the sequence margins are $+85.01$ for AdamW, $+73.52$ for DAPR, and -3.80 for SAM. This contrasts with Gemma's trajectory and complements the model-dependent capability results. These cross-model observations do not isolate normalization architecture or capacity.

B.3 Scope and Extensions

B.3.1 Multilingual Instruction Tuning and Localized Center Dynamics

As shown in Table 5 (Section 3.4), language-aware centering (LA-DAPR) outperforms shared-center DAPR across all target languages on Gemma-3-4B and Llama-3-8B after 10,000 steps of joint Aya SFT. Each language maintains its own loss-center statistics: $\mathcal{L}^{(l)}(\boldsymbol{\theta}_t) > c_t^{(l)}$ selects additional preconditioned descent, while $\mathcal{L}^{(l)}(\boldsymbol{\theta}_t) \leq c_t^{(l)}$ selects projected ascent with initialization anchoring. The reported branch-switch milestones occur at 3,650–4,110 steps for European languages (German, Spanish, French) and 6,980–7,240 steps for Chinese, indicating distinct convergence dynamics and branch selection stages across diverse language families.

B.3.2 Additional DPO Stress Results at Larger Scales

Table 9 reports 10K-step DPO stress results on Gemma-3-12B and Qwen-2.5-14B under $\beta = 0.01$ and $\eta = 2 \times 10^{-5}$. Standard DPO and UltraChat SFT endpoints for these models appear in Table 2.

Model	Optimizer	ARC-C	MMLU	HellaSwag	GSM8K
Gemma-3-12B	AdamW	72.01	71.73	84.90	**78.32**
	SAM	**72.78**	**71.79**	**87.37**	77.03
	DAPR	72.27	71.34	84.37	76.72
Qwen-2.5-14B	AdamW	62.46	77.82	83.71	86.58
	SAM	**65.27**	77.67	**86.94**	**89.46**
	DAPR	64.42	**77.89**	83.92	76.50

Table 9: **DPO stress retention after 10K steps on 12B and 14B models** ($\beta = 0.01, \eta = 2 \times 10^{-5}$). Scores are percentages; bold marks highest score among the three optimizers.

C Extended Related Work and Positioning

This section expands upon the foundational and contemporary context introduced in Section 4, comparing DAPR with five dominant paradigms for preserving model capabilities, mitigating alignment tax, and navigating non-convex loss landscapes during post-training.

Parameter-Space Geometry vs. Output-Level Regularization. Standard approaches to preventing alignment tax and capability forgetting typically enforce token distribution constraints in the output space, such as forward/reverse Kullback–Leibler (KL) divergence penalties against a frozen reference model (Ouyang et al., 2022; Rafailov et al., 2023). While effective at tethering policy token distributions, output-space regularizers suffer from two notable shortcomings: (i) *Computational and memory overhead*: Evaluating token-level KL penalties requires executing a synchronized forward pass through an auxiliary frozen reference model at every training iteration, effectively doubling the forward FLOPs and expanding GPU memory footprints. (ii) *Geometric detachment*: Output penalties constrain token logits along the data distribution but remain agnostic to the parameter-space loss landscape geometry. Consequently, the model can drift into razor-sharp loss minima that exhibit high sensitivity to distribution shifts or downstream evaluation prompts. DAPR applies a conditional correction in parameter space. Proposition 1 identifies a squared-gradient-norm term in a simplified descent–ascent pair, and Corollary 1 gives a transverse gradient-norm interpretation for the practical update under its stated assumptions. Table 1 complements this local analysis with empirical geometry measurements. The correction reuses gradients from the selected training objective and adds no reference-model evaluation.

Optimizer-Level Regularization vs. Continual Routing Architectures. A distinct line of research treats capability retention as a continual learning problem addressed via modular or dynamic routing architectures. Dynamic modular routing networks allocate specialized adapter capacity across task domains, while preserving base representations through parameter isolation. While modular architectures prevent cross-task interference, they alter the underlying network architecture, necessitate task/domain identifiers or gating classifiers, and increase inference routing latency. DAPR approaches capability retention through a complementary, purely optimizer-level philosophy. By regularizing the optimization trajectory during backward gradient updates, DAPR produces a single, standard parameter-efficient adapter (or fully merged dense weights) that requires zero routing logic, zero architecture modification, and identical inference latency to conventional AdamW fine-tuning.

Intrinsic Curvature Regularization vs. Replay and Self-Distillation. Experience replay and self-distillation methods mitigate alignment drift by interleaving general pre-training corpora or distilling previous checkpoint generations into the current policy. Similarly, pseudo-rehearsal schemes generate synthetic rollouts to anchor the student policy to pre-adaptation competencies. While conceptually appealing, rehearsal methods require maintaining complex data ingestion pipelines, curate balanced replay mixtures, and introduce secondary drift if synthetic rollouts degrade in quality. DAPR circumvents external replay buffers entirely: its dual mechanism combines intrinsic geometric regularization (via projected ascent) with base parameter anchoring (λ_{base}), thereby anchoring the policy to pre-adaptation competencies without storing or sampling auxiliary data.

Instantaneous Preconditioned Projection vs. Offline Activation Orthogonalization. Projection methods differ in the information used to define their constraints. Activation-based approaches construct projection subspaces from offline calibration representations. DAPR instead uses the current Adam direction and preconditioner, computing its correction separately for each trainable tensor without an activation-space factorization. The geometry follows from Eq. 3 and Proposition 2.

Fix a step and a tensor block, suppress its superscript (j), and hold $\boldsymbol{P}_t \succ 0$, $\boldsymbol{g}_t$, and $\boldsymbol{u}_t$ fixed. Using the same notation as Proposition 2, set $a_t = \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{u}_t$ and $b_t = \boldsymbol{u}_t^\top \boldsymbol{P}_t \boldsymbol{u}_t$. The correction is

$$\boldsymbol{q}_t = \boldsymbol{P}_t \boldsymbol{g}_t - \frac{a_t}{b_t + \epsilon_{\text{proj}}} \boldsymbol{P}_t \boldsymbol{u}_t. \tag{35}$$

For $\epsilon_{\text{proj}} = 0$ and $\boldsymbol{u}_t \neq \boldsymbol{0}$, consider the constrained problem

$$\min_{\boldsymbol{z}} \ \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{P}_t \boldsymbol{g}_t\|^2_{\boldsymbol{P}_t^{-1}} \quad \text{subject to} \quad \boldsymbol{u}_t^\top \boldsymbol{z} = 0.$$

Its stationarity equation is $\boldsymbol{P}_t^{-1}(\boldsymbol{z} - \boldsymbol{P}_t\boldsymbol{g}_t) + \nu\boldsymbol{u}_t = 0$, so $\boldsymbol{z} = \boldsymbol{P}_t\boldsymbol{g}_t - \nu\boldsymbol{P}_t\boldsymbol{u}_t$. Substitution into the constraint gives $\nu = a_t/b_t$, recovering Eq. 35. Thus $\boldsymbol{P}_t^{-1}$ measures the distance to the unprojected correction, while the constraint imposes Euclidean orthogonality to $\boldsymbol{u}_t$. Equivalently, $\boldsymbol{g}_{\text{ortho}}$ is orthogonal to $\boldsymbol{u}_t$ in the $\boldsymbol{P}_t$ inner product.

For the implemented $\epsilon_{\text{proj}} > 0$, the same correction uniquely minimizes the soft-constraint objective

$$\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{P}_t\boldsymbol{g}_t\|^2_{\boldsymbol{P}_t^{-1}} + \frac{(\boldsymbol{u}_t^\top \boldsymbol{z})^2}{2\epsilon_{\text{proj}}}.$$

Indeed, stationarity yields $\boldsymbol{z} = \boldsymbol{P}_t\boldsymbol{g}_t - (\boldsymbol{u}_t^\top \boldsymbol{z})\boldsymbol{P}_t\boldsymbol{u}_t/\epsilon_{\text{proj}}$; taking its inner product with $\boldsymbol{u}_t$ and substituting back recovers Eq. 35. Direct calculation gives

$$\begin{aligned} \boldsymbol{u}_t^\top \boldsymbol{q}_t &= \frac{\epsilon_{\text{proj}} a_t}{b_t + \epsilon_{\text{proj}}}, \\ \boldsymbol{g}_t^\top \boldsymbol{q}_t &= \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{g}_t - \frac{a_t^2}{b_t + \epsilon_{\text{proj}}} \geq 0. \end{aligned} \tag{36}$$

The inequality follows from $a_t^2 \leq b_t \boldsymbol{g}_t^\top \boldsymbol{P}_t \boldsymbol{g}_t$. Relative to the unprojected inner product $\boldsymbol{u}_t^\top \boldsymbol{P}_t \boldsymbol{g}_t = a_t$, stabilization leaves the fraction $\epsilon_{\text{proj}}/(b_t + \epsilon_{\text{proj}})$. Consequently, approximate orthogonality requires this fraction to be small; a small absolute stabilizer alone is insufficient when b_t is also small. If $\boldsymbol{u}_t = \boldsymbol{0}$, the correction is $\boldsymbol{P}_t\boldsymbol{g}_t$ and the orthogonality constraint is vacuous.

These identities apply block by block. With nonnegative strengths $\alpha_t^{(j)}$, exact projection gives $\boldsymbol{u}_t^\top \boldsymbol{A}_t \boldsymbol{q}_t = 0$, while stabilization leaves the sum of the block residuals weighted by $\alpha_t^{(j)}$, as detailed in Appendix A.3. Hence exact projection removes the ascent correction's component along the current Adam direction; stabilized projection attenuates that component. The complete step combines this correction with Adam and initialization anchoring. Appendix A.4 analyzes their joint contribution over a crossing pair.

Holistic Retention Diagnostics and Empirical Tracking. Recent critical analyses emphasize that conventional post-training evaluations often exhibit metric myopia, focusing narrowly on target-task loss or reward model scores while failing to diagnose insidious degradation in reasoning, commonsense, or multilingual comprehension (Thede et al., 2026). Thede et al. (2026) advocate for continuous tracking of pre-adaptation competencies. Our experimental evaluation (Section 3.2) adopts this holistic diagnostic ethos: we benchmark checkpoints across ARC Challenge, MMLU, HellaSwag, and GSM8K relative to unadapted base model checkpoints, alongside tracking policy sequence margins ($\Delta \log \pi_\theta$) and token logit dynamics (Appendix B.2.4). The additional optimizer comparisons and curvature diagnostics complement these capability measurements.

D Scope and Limitations

Geometry and architectural variation. The diagnostics describe local geometry for a specified loss and parameter subspace. Their variation across models and interventions motivates further study of architecture and optimization, but does not isolate a causal architectural mechanism. A lower trace or gradient norm alone is not a certificate of generalization.

Training objectives and checkpoint selection. The SFT and standard-DPO results show that the value of additional intervention depends on the training objective and setting. DAPR improves all eight main 4B/8B extended-SFT capability comparisons with AdamW, whereas standard DPO and limited-data SFT yield mixed differences. Weakening the DPO reference weight while increasing the learning rate produces a pronounced late-stage drop on Gemma, but not the same pattern on Llama. The stress test changes both settings jointly and does not isolate the effect of β alone. Stress-test capability trajectories and preference diagnostics appear in Appendices B.2.2 and B.2.4.

Our fixed-horizon protocol evaluates retention without capability-based checkpoint selection. Early stopping is a valid alternative when a validation suite is available; a matched comparison should specify the evaluation budget, selection rule, and resulting target-task quality. Such a comparison is required before claiming an advantage over a selected early-stopping checkpoint.

Language-specific centers. LA-DAPR conditions center statistics on language metadata while sharing the model and Adam state. It coincides with DAPR for a single-language stream with identical initialization, hyperparameters, and batches. The joint multilingual results support this extension for the tested mixture; effects on gradient conflict, other language frequencies, and unseen languages remain open.

Generalization of the evidence. The experiments cover finite model, data, LoRA, and training-budget choices. Retained capabilities must be assessed alongside target-task success and measured cost, and short-run sensitivity needs longer-horizon validation. The scalar analysis motivates DAPR; the practical Adam-based interpretation retains the local assumptions in Appendix A.4. Empirical proximity to SAM should therefore be distinguished from algorithmic equivalence or a guarantee against forgetting.