CURVDELTA: LOW-CURVATURE DELTA MODELING FOR SURGICAL LEARNING AND UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A central challenge in adapting large language models (LLMs) is capability preservation: an adaptation can achieve its target while degrading general capabilities. We introduce CurvDelta, a scalable framework that incorporates the preservation constraint into the parameterization of the model update. By analyzing the local geometry of the preservation constraint around the pre-adaptation model, CurvDelta constructs a low-curvature space in which parameter updates leave the preservation loss nearly invariant. We make its construction practical for LLMs through Kronecker-factored approximate curvature (K-FAC) and a factorized projection. We also introduce CurvLoRA, a low-rank variant within the same preservation geometry, designed for more efficient adaptation. Both methods apply to preservation-constrained adaptation tasks such as capability learning and selective unlearning. We evaluate our methods on mathematical reasoning, code generation, and knowledge unlearning. For mathematical reasoning and code generation, our approaches achieve competitive downstream performance while better preserving general capabilities, offering a favorable acquisition–retention trade-off relative to the baselines. For knowledge unlearning, our main results on TOFU and MUSE show a better balance between targeted forgetting and knowledge retention than the baselines, with retention performance close to the from-scratch retraining reference.

1 INTRODUCTION

Large language models (LLMs) need to evolve after pretraining. New tasks and domains call for additional capabilities, while privacy, safety, and data-governance requirements can necessitate selective unlearning. Retraining from scratch for every adaptation query is costly, motivating fine-tuning and parameter-efficient methods for capability learning and selective unlearning (Hu et al., 2022; Cao & Yang, 2015; Maini et al., 2024; Shi et al., 2025). Although learning and unlearning change behavior in different ways, both require the update to be selective: progress on the target objective should preserve unrelated knowledge and capabilities.

To preserve existing capabilities while allowing the changes needed for adaptation, prior work has introduced different approaches to limiting unwanted changes. Replay and regularization incorporate preservation into the training objective (Rolnick et al., 2019; Kirkpatrick et al., 2017). Another line of work constrains the space in which adaptation may move, either by projecting task gradients (Chaudhry et al., 2019; Ikram et al., 2026) or by heuristically restricting adaptation to orthogonal or null subspaces of parameters or representations (Fang et al., 2025; Xiong & Xie, 2026; Ju et al., 2026; Liang & Li, 2024; Luo et al., 2026). Despite improvements, these methods either do not guarantee that the parameter update satisfies the intended constraint or impose strong constraints in parameter or representation space that are only indirectly tied to capability preservation. These limitations can lead to an unfavorable adaptation–preservation trade-off.

In this paper, we adopt a first-principles formulation of preservation-constrained adaptation. We seek a model delta that minimizes the task loss subject to a bound on the preservation loss, which measures changes in the model's behavior on a designated preservation set. We therefore ask:

***(Q)** How can we directly incorporate the preservation constraint into the model delta parameterization while retaining the flexibility needed for adaptation?*

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

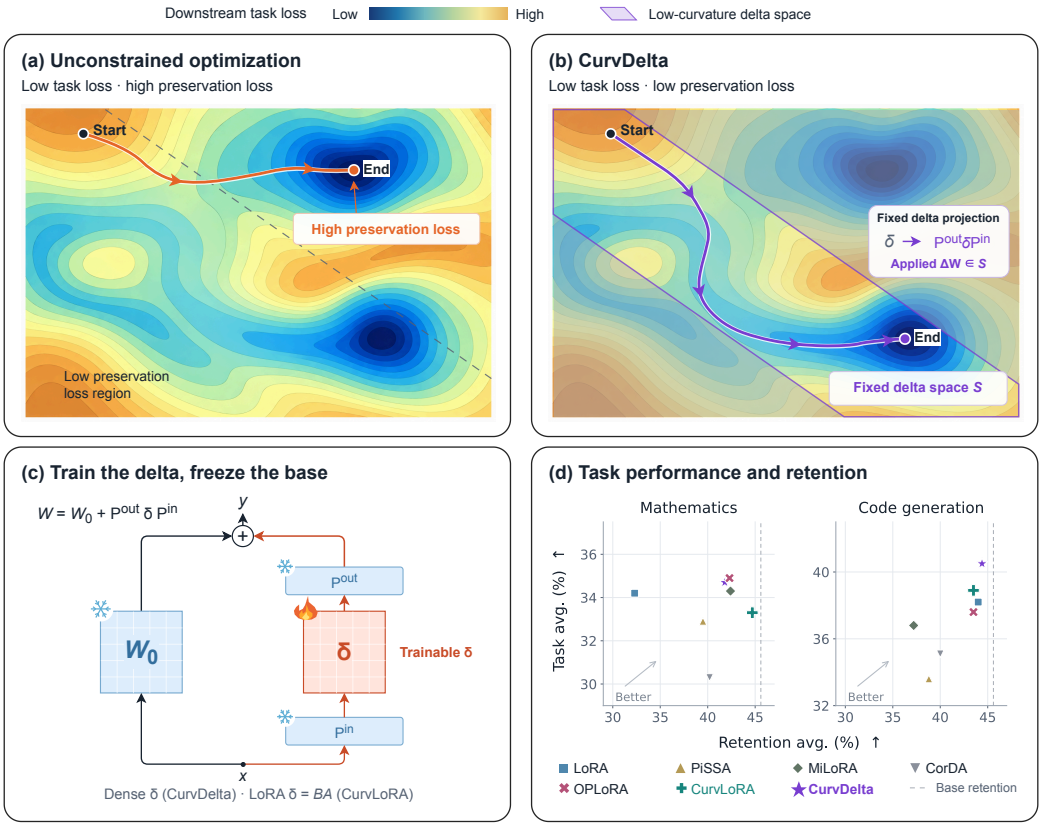


Figure 1: **Overview of CurvDelta. (a)** Unconstrained optimization reaches low task loss but high preservation loss. **(b)** CurvDelta constructs a low-curvature delta space. The applied delta remains in this space throughout optimization, reaching a different solution with comparable task loss. **(c)** The base weights and projections remain frozen. The trainable delta is a dense matrix in CurvDelta or LoRA in CurvLoRA. **(d)** Performance comparison. The dashed line marks the base model's retention average. Our methods achieve the best trade-off between task performance and capability retention.

To address this question, we introduce CurvDelta, a framework for low-curvature delta modeling for surgical learning and unlearning (Figure 1). CurvDelta learns a dense delta within a fixed low-curvature space. We also introduce CurvLoRA, a low-rank variant that uses the same preservation geometry for more efficient adaptation. The method is built around the following three core ideas.

1. Preservation through a task-independent delta parameterization. CurvDelta derives a low-curvature delta space from the local geometry of the preservation constraint around the pre-adaptation model. It constrains a trainable delta into this fixed space before applying it, keeping the realized update constrained throughout optimization. Because the space depends only on the base model and preservation set, the same preservation geometry can be reused across task sets, task objectives, and delta parameterizations.

2. Scalable delta space construction for dense and low-rank adaptation. To make low-curvature delta space construction tractable at LLM scale, CurvDelta approximates preservation curvature with a block-diagonal Kronecker-factored approximate curvature (K-FAC) (Martens & Grosse, 2015). It further applies factorwise spectral truncation to obtain a separable two-sided projection that can be stored and applied efficiently using compact eigenvector bases. The same construction supports both dense adaptation in CurvDelta and low-rank adaptation in CurvLoRA.

3. An explicit preservation guarantee from curvature and displacement. We show that preservation is jointly controlled by the curvature of the retained delta space and the magnitude of the realized update. Specifically, we derive an upper bound on the local preservation loss in which curvature deter-

mines the cost per unit displacement, while the delta norm determines the total displacement. Under the local quadratic and block-diagonal K-FAC models, this bound provides a sufficient condition for both CurvDelta and CurvLoRA to satisfy the preservation constraint.

Experimental results. We evaluate our framework on capability learning and selective unlearning. For capability learning, we study mathematical reasoning and code generation together with retention of general knowledge. Our approach achieves competitive downstream performance while reducing degradation in general capabilities, yielding a favorable acquisition–retention trade-off relative to the evaluated baselines. For selective unlearning, our main results on TOFU and MUSE show a better balance between targeted forgetting and knowledge retention than the evaluated baselines. On TOFU, our approach suppresses targeted knowledge while maintaining utility comparable to the retraining reference. On MUSE, it substantially reduces targeted memorization while retaining most of the retrained model's performance.

2 Problem Formulation: Preservation-Constrained Adaptation

Let $\boldsymbol{\theta}_0$ be the initial parameter vector of a language model. We write $\boldsymbol{\theta} = \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}$ for the adapted parameters, with $\Delta\boldsymbol{\theta}$ the *update*, and $p_{\boldsymbol{\theta}}$ for the distribution induced by $\boldsymbol{\theta}$. Preservation-constrained adaptation seeks an update $\Delta\boldsymbol{\theta}$ that achieves a specified behavior change while leaving unrelated behaviors intact. Each adaptation can be described by a *task set* $\mathcal{D}_{\text{task}}$, which encodes the desired behavior, and a *preservation set* $\mathcal{D}_{\text{pres}}$, which marks behavior that should remain unchanged. For capability learning, $\mathcal{D}_{\text{task}}$ contains data from a new task; for selective unlearning, it contains examples of the behavior to be removed. $\mathcal{D}_{\text{pres}}$ may comprise pretraining data or a representative proxy dataset.

Let $\mathcal{L}_{\text{task}}(\boldsymbol{\theta})$ denote the task loss on $\mathcal{D}_{\text{task}}$. We define the preservation loss as $\mathcal{L}_{\text{pres}}(\boldsymbol{\theta}) = \mathbb{E}_{x\sim\mathcal{D}_{\text{pres}}}[d(\boldsymbol{z}_{\boldsymbol{\theta}}(x), \boldsymbol{z}_{\boldsymbol{\theta}_0}(x))]$, where $\boldsymbol{z}_{\boldsymbol{\theta}}(x)$ denotes the model output vector and d measures its deviation from the original output. We then formulate preservation-constrained adaptation as the following constrained optimization problem:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{task}}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \mathcal{L}_{\text{pres}}(\boldsymbol{\theta}) \le \varepsilon, \tag{1}$$

where $\varepsilon > 0$ specifies the tolerated preservation drift. This formulation covers both acquiring a new capability and removing a targeted behavior, with only the choice of $\mathcal{L}_{\text{task}}$ distinguishing the two.

A standard way to optimize this problem is to minimize the Lagrangian $\mathcal{L}_{\text{task}}(\boldsymbol{\theta}) + \lambda\,\mathcal{L}_{\text{pres}}(\boldsymbol{\theta})$, with $\lambda \ge 0$ controlling the task–preservation trade-off. This relaxation has two limitations. First, λ does not translate into the tolerance ε of Eq. (1): the drift is known only after training. Second, the Lagrangian is itself a multi-task objective, and gradient conflict makes the two objectives difficult to satisfy jointly (Yu et al., 2020; Liu et al., 2021; Chen et al., 2020), especially in unlearning (Xiao et al., 2026a).

3 CurvDelta: Low-Curvature Delta Modeling

We now present CurvDelta, which turns the preservation constraint in Problem (1) into a restriction on the model update $\Delta\boldsymbol{\theta}$. The key idea is to optimize the realized delta within a low-curvature subspace of $\mathcal{L}_{\text{pres}}$ at the original parameters $\boldsymbol{\theta}_0$.

3.1 Local Curvature of the Preservation Loss

To justify restricting $\Delta\boldsymbol{\theta}$ to low-curvature directions, we first relate curvature to the preservation cost induced by an update. Since $\mathcal{L}_{\text{pres}}$ measures the output discrepancy from the original model, we analyze its local variation around $\boldsymbol{\theta}_0$ under the following assumption on d.

Assumption 1. The map $\boldsymbol{\theta} \mapsto \boldsymbol{z}_{\boldsymbol{\theta}}(x)$ is twice continuously differentiable, and the discrepancy d satisfies $d(\boldsymbol{v}, \boldsymbol{v}) = 0$, $\nabla_{\boldsymbol{u}} d(\boldsymbol{u}, \boldsymbol{v})|_{\boldsymbol{u}=\boldsymbol{v}} = 0$, and $\nabla^2_{\boldsymbol{u}} d(\boldsymbol{u}, \boldsymbol{v})|_{\boldsymbol{u}=\boldsymbol{v}} \succeq 0$ for every model output $\boldsymbol{v}$.

These conditions are mild: a twice-differentiable nonnegative d with $d(\boldsymbol{v}, \boldsymbol{v}) = 0$ is minimized at $\boldsymbol{u} = \boldsymbol{v}$, so the remaining two conditions follow. Bregman divergences provide a broad family of such discrepancies (Bregman, 1967); in particular, log-sum-exp on logits gives $d(\boldsymbol{z}_{\boldsymbol{\theta}}, \boldsymbol{z}_{\boldsymbol{\theta}_0}) = D_{\text{KL}}(p_{\boldsymbol{\theta}_0} \| p_{\boldsymbol{\theta}})$, which we use throughout. Appendix B.1 gives the derivation.

Under Assumption 1, the second-order Taylor expansion around $\boldsymbol{\theta}_0$ yields the approximation:

$$\mathcal{L}_{\text{pres}}(\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}) \approx \frac{1}{2}\Delta\boldsymbol{\theta}^\top \boldsymbol{H}_{\text{pres}} \Delta\boldsymbol{\theta}. \tag{2}$$

where $\boldsymbol{H}_{\text{pres}} := \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{\text{pres}}(\boldsymbol{\theta}_0)$. By the second-order chain rule, $\boldsymbol{H}_{\text{pres}}$ consists of the generalized Gauss–Newton (GGN) term below and a residual of the form $\mathbb{E}_x[\sum_k (\partial_{z_k} d)\nabla^2_{\boldsymbol{\theta}} z_{\boldsymbol{\theta},k}(x)]$. At $\boldsymbol{\theta}_0$, the two arguments of d coincide, and Assumption 1 gives $\partial_{z_k} d = 0$ for every k. The residual therefore vanishes, yielding

$$\boldsymbol{G}_{\text{pres}} := \mathbb{E}_{x\sim\mathcal{D}_{\text{pres}}}\left[\boldsymbol{J}_0(x)^\top \boldsymbol{H}_z(x) \boldsymbol{J}_0(x)\right] = \boldsymbol{H}_{\text{pres}} \succeq \boldsymbol{0}, \tag{3}$$

Here, $\boldsymbol{J}_0(x)$ and $\boldsymbol{H}_z(x)$ are respectively the parameter Jacobian of the outputs and the output-space Hessian of d, both evaluated at $\boldsymbol{\theta}_0$. Since $\boldsymbol{H}_z(x) \succeq \boldsymbol{0}$, the resulting GGN matrix is positive semidefinite. Appendix B.2 gives the full derivation.

Substituting the quadratic approximation into the constraint $\mathcal{L}_{\text{pres}} \le \varepsilon$ gives

$$\min_{\Delta\boldsymbol{\theta}} \mathcal{L}_{\text{task}}(\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}) \quad \text{s.t.} \quad \Delta\boldsymbol{\theta}^\top \boldsymbol{G}_{\text{pres}} \Delta\boldsymbol{\theta} \le 2\varepsilon. \tag{4}$$

Let $(\lambda_i, \boldsymbol{u}_i)$ denote the eigenpairs of $\boldsymbol{G}_{\text{pres}}$, and write $\Delta\boldsymbol{\theta} = \sum_i \alpha_i \boldsymbol{u}_i$. Then the constraint becomes $\sum_i \lambda_i \alpha_i^2 \le 2\varepsilon$, whereas $\|\Delta\boldsymbol{\theta}\|_2^2 = \sum_i \alpha_i^2$. Thus, for a fixed update norm, concentrating the update on eigenvectors with smaller eigenvalues incurs less preservation loss. We therefore construct the delta space from the trailing eigenspace of $\boldsymbol{G}_{\text{pres}}$, which spans its low-curvature subspace. Appendix B.3 gives the full derivation.

3.2 Constructing the Low-Curvature Delta Space

Computing the trailing eigenspace of $\boldsymbol{G}_{\text{pres}}$ is intractable at LLM scale. We therefore approximate $\boldsymbol{G}_{\text{pres}}$ with block-diagonal K-FAC (Martens & Grosse, 2015), using one block for each adapted linear weight. For $\boldsymbol{W}_l \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the corresponding block is

$$\boldsymbol{G}^{(l)}_{\text{pres}} \approx \boldsymbol{K}^{\text{in}}_l \otimes \boldsymbol{K}^{\text{out}}_l, \qquad \boldsymbol{K}^{\text{in}}_l = \mathbb{E}[\boldsymbol{a}_{l-1}\boldsymbol{a}_{l-1}^\top], \quad \boldsymbol{K}^{\text{out}}_l = \mathbb{E}[\boldsymbol{g}_l \boldsymbol{g}_l^\top]. \tag{5}$$

Here, $\boldsymbol{a}_{l-1}$ is the input activation and $\boldsymbol{g}_l$ is the layer output gradient. We estimate the factors once at $\boldsymbol{\theta}_0$ on $\mathcal{D}_{\text{pres}}$ using the Monte Carlo Fisher estimator in Appendix C.1.

For clarity, we omit the layer index below. Let $(\lambda^{\text{out}}_i, \boldsymbol{u}^{\text{out}}_i)$ and $(\lambda^{\text{in}}_j, \boldsymbol{u}^{\text{in}}_j)$ denote the factor eigenpairs, with eigenvalues in descending order. By the Kronecker eigenstructure, the matrix direction $\boldsymbol{u}^{\text{out}}_i (\boldsymbol{u}^{\text{in}}_j)^\top$ has curvature $\lambda^{\text{out}}_i \lambda^{\text{in}}_j$ under the K-FAC block. We construct a low-curvature delta space by retaining directions formed from trailing eigenvectors of both factors.

Factorwise spectral truncation. For $\gamma \in (0, 1)$, let r_{out} and r_{in} be the smallest ranks at which the cumulative spectral energy reaches γ for the respective factors. Let $\boldsymbol{U}^{\text{out}}_h$ and $\boldsymbol{U}^{\text{in}}_h$ contain these leading eigenvectors. The orthogonal projectors onto the complementary trailing eigenspaces are

$$\boldsymbol{P}^{\text{out}} = \boldsymbol{I} - \boldsymbol{U}^{\text{out}}_h (\boldsymbol{U}^{\text{out}}_h)^\top, \qquad \boldsymbol{P}^{\text{in}} = \boldsymbol{I} - \boldsymbol{U}^{\text{in}}_h (\boldsymbol{U}^{\text{in}}_h)^\top. \tag{6}$$

Given a trainable pre-projection delta $\boldsymbol{\delta}(\boldsymbol{\phi})$ with the same shape as $\boldsymbol{W}$, we define the applied delta as

$$\Delta\boldsymbol{W} := \boldsymbol{P}^{\text{out}}\, \boldsymbol{\delta}(\boldsymbol{\phi})\, \boldsymbol{P}^{\text{in}}. \tag{7}$$

This projection retains exactly the span of $\boldsymbol{u}^{\text{out}}_i (\boldsymbol{u}^{\text{in}}_j)^\top$ with $i > r_{\text{out}}$ and $j > r_{\text{in}}$. The resulting delta space has dimension $(d_{\text{out}} - r_{\text{out}})(d_{\text{in}} - r_{\text{in}})$. Appendix B.4 gives the derivation.

Curvature of the retained space. The largest eigenvalue in the retained space is the product of the largest retained factor eigenvalues. Expanding $\Delta\boldsymbol{W}$ in the corresponding orthonormal basis therefore yields

$$\kappa := \lambda^{\text{out}}_{r_{\text{out}}+1} \lambda^{\text{in}}_{r_{\text{in}}+1},$$
$$\text{vec}(\Delta\boldsymbol{W})^\top (\boldsymbol{K}^{\text{in}} \otimes \boldsymbol{K}^{\text{out}}) \,\text{vec}(\Delta\boldsymbol{W}) \le \kappa \|\Delta\boldsymbol{W}\|_F^2,$$

Thus, under the K-FAC model, the layer's contribution to the quadratic preservation loss is bounded by $\frac{1}{2}\kappa\|\Delta\boldsymbol{W}\|_F^2$. Appendix B.6 gives the proof. As γ increases, both κ and the dimension of the retained

space are nonincreasing, so the threshold jointly controls the curvature bound and the available update space.

Our factorwise construction yields a subspace of the full K-FAC eigenspace with eigenvalues at most κ. It may also discard potentially useful low-curvature directions. Nevertheless, empirical evidence of abundant flat directions (Sagun et al., 2017; Ghorbani et al., 2019) and low-dimensional optimization (Li et al., 2018; Aghajanyan et al., 2021) suggests that the retained space may still provide sufficient capacity for adaptation. Appendix B.5 characterizes the excluded directions and their effect on task optimization.

3.3 Delta Parameterizations and Optimization

Having constructed the fixed low-curvature update space in Section 3.2, we now turn to representing and optimizing task-specific updates within it. The projected form supports two complementary parameterizations: a dense delta that spans the entire retained space and a low-rank LoRA delta that reduces the number of trainable parameters, yielding CurvDelta and CurvLoRA, respectively.

Algorithm 1 Low-Curvature Delta Modeling

Require: Initial parameters $\boldsymbol{\theta}_0$; task and preservation sets $\mathcal{D}_{\text{task}}$, $\mathcal{D}_{\text{pres}}$; task loss $\mathcal{L}_{\text{task}}$.
Require: Target weights $\mathcal{T}$; threshold γ; delta parameterization $\boldsymbol{\delta}_l(\boldsymbol{\phi}_l)$; learning rates $\{\eta_t\}_{t=1}^{T}$.
Ensure: Adapted parameters $\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}$.
1: Estimate the K-FAC factors at $\boldsymbol{\theta}_0$ on $\mathcal{D}_{\text{pres}}$ (Appendix C.1).
2: Cache the leading factor eigenbases selected by γ.
3: Initialize $\boldsymbol{\phi}_l$; keep $\boldsymbol{\theta}_0$ fixed.
4: **for** $t = 1$ to T **do**
5: For each $l \in \mathcal{T}$, apply the projected delta $\Delta\boldsymbol{W}_l = \boldsymbol{P}_l^{\text{out}}\boldsymbol{\delta}_l(\boldsymbol{\phi}_l)\boldsymbol{P}_l^{\text{in}}$.
6: Take an optimizer step on $\{\boldsymbol{\phi}_l\}_{l\in\mathcal{T}}$ loss $\mathcal{L}_{\text{task}}(\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta})$.
7: **end for**
8: Recompute $\Delta\boldsymbol{\theta}$ from the final delta parameters.
9: **return** $\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}$.

Dense parameterization. CurvDelta parameterizes the delta as a dense matrix $\boldsymbol{\delta}_l \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$ initialized at zero. Under the projection in Eq. (7), this parameterization spans the entire low-curvature space. It has $d_{\text{out}}d_{\text{in}}$ trainable parameters per weight and $(d_{\text{out}} - r_{\text{out}})(d_{\text{in}} - r_{\text{in}})$ effective degrees of freedom after projection.

Low-rank parameterization. CurvLoRA reduces the number of trainable parameters through a LoRA factorization (Hu et al., 2022),

$$\boldsymbol{\delta}_l = \boldsymbol{B}_l\boldsymbol{A}_l, \qquad \Delta\boldsymbol{W}_l = (\boldsymbol{P}_l^{\text{out}}\boldsymbol{B}_l)(\boldsymbol{A}_l\boldsymbol{P}_l^{\text{in}}),$$

where $\boldsymbol{B}_l \in \mathbb{R}^{d_{\text{out}}\times r}$ and $\boldsymbol{A}_l \in \mathbb{R}^{r\times d_{\text{in}}}$, with $\boldsymbol{B}_l$ initialized at zero. This parameterization has $r(d_{\text{out}} + d_{\text{in}})$ trainable parameters per weight and restricts the applied delta to rank at most r within the same low-curvature space. The factorized form avoids materializing a dense weight matrix.

Both variants apply the projection implicitly through the cached leading eigenvector bases. The corresponding computations and their time and memory complexity are detailed in Appendix C.2.

Optimization. The preservation geometry is estimated once before training, and the resulting projectors and base parameters remain fixed. For both parameterizations, we minimize $\mathcal{L}_{\text{task}}(\boldsymbol{\theta}_0+\Delta\boldsymbol{\theta})$ over the delta parameters—$\boldsymbol{\delta}_l$ for CurvDelta and $(\boldsymbol{B}_l, \boldsymbol{A}_l)$ for CurvLoRA—by differentiating through the projection. Since the projection is part of the parameterization, the applied delta remains in the prescribed space throughout optimization, independently of the optimizer. For the dense parameterization, ordinary SGD from a zero delta is equivalent to gradient projection onto the same fixed space. With gradient projection, however, optimizers such as Adam rescale gradient coordinates by different adaptive factors, which can move the parameter increment outside that space. Algorithm 1 summarizes the shared training procedure.

3.4 Preservation Guarantee

Under the block-diagonal K-FAC model, the quadratic preservation loss decomposes over the adapted weights. Summing the layerwise bounds from Section 3.2 therefore gives the following guarantee.

Theorem 1 (ε-feasible preservation)**.** *Under the local quadratic model of Section 3.1 and the block-diagonal K-FAC model of Assumption 2, let $\Delta\boldsymbol{\theta}$ be supported on $\mathcal{T}$ with layer deltas given by Eq.* (7)*. Then*

$$\mathcal{L}_{\mathrm{pres}}(\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}) \approx \frac{1}{2}\Delta\boldsymbol{\theta}^\top \boldsymbol{G}_{\mathrm{pres}}\Delta\boldsymbol{\theta} \leq \frac{1}{2}\sum_{l\in\mathcal{T}} \kappa_l \|\Delta\boldsymbol{W}_l\|_F^2, \tag{8}$$

where κ_l is the curvature coefficient for layer l from Section 3.2. The update is feasible under the local quadratic model whenever $\sum_{l\in\mathcal{T}} \kappa_l \|\Delta\boldsymbol{W}_l\|_F^2 \leq 2\varepsilon$.

The proof applies Proposition 2 to each block (Appendix B.7). The geometry fixes κ_l, while the delta norms depend on the optimization trajectory, including the learning-rate schedule and number of training steps.

4 Experiments

We summarize the setup and main results here, and provide the experimental details in Appendix E.

4.1 Capability Learning

4.1.1 Experimental Setup

Tasks, datasets, and evaluation. We study capability acquisition in mathematical reasoning and code generation. For mathematical reasoning, we fine-tune on 100K examples sampled from MetaMathQA (Yu et al., 2024) and evaluate on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b). For code generation, we fine-tune on 104K examples sampled from CodeFeedback (Zheng et al., 2024) and evaluate on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). To measure knowledge retention, we evaluate on TriviaQA (Joshi et al., 2017), NQ-Open (Kwiatkowski et al., 2019), and MMLU (Hendrycks et al., 2021a). For the *preservation set* $\mathcal{D}_{\mathrm{pres}}$, we use 256 examples sampled from the FineFineWeb corpus (M-A-P et al., 2024).

Models. We evaluate both tasks using the pretrained base checkpoints of Gemma-3-1B (Gemma Team et al., 2025) and Llama-2-7B (Touvron et al., 2023). For PEFT methods and CurvLoRA, we set the rank to 128 for Gemma-3-1B and 32 for Llama-2-7B, holding it fixed across tasks.

Baselines. We include standard fine-tuning (FT) as an unconstrained adaptation reference and experience replay (Replay) (Rolnick et al., 2019) as a data-based baseline. Replay mixes task examples with examples sampled from $\mathcal{D}_{\mathrm{pres}}$ during fine-tuning. We additionally include CrispEdit (Ikram et al., 2026) as a curvature-based gradient-projection baseline and OPLoRA (Xiong & Xie, 2026) as an orthogonal delta-projection baseline. We further compare against LoRA (Hu et al., 2022), PiSSA (Meng et al., 2024), MiLoRA (Wang et al., 2025a), and the knowledge-preserved version of CorDA (Yang et al., 2024).

4.1.2 Main Results

Tables 1 and 2 compare the acquisition–retention trade-off in mathematical reasoning and code generation. Full fine-tuning provides strong capability acquisition but consistently reduces general-knowledge retention, whereas CurvDelta keeps downstream performance competitive while preserving substantially more general knowledge across both domains. This pattern is strongest for code generation: CurvDelta attains the strongest joint averages on both backbones, with downstream performance close to or above full fine-tuning and the highest average retention among the reported adaptation methods. On mathematical reasoning, CurvDelta remains close to full fine-tuning in downstream performance while substantially narrowing the retention gap. The low-rank trade-off varies by backbone: CurvLoRA achieves the highest average retention among adaptation methods on Llama, though with lower downstream performance than CurvDelta, while CurvDelta provides

Table 1: **Capability learning on mathematical reasoning.** Within each backbone, the best and second-best results among adaptation methods are shown in bold and underlined, respectively.

Method	Downstream tasks (%)			Retention tasks (%)				ALL
	GSM8K	MATH	Avg.	TriviaQA	NQ-Open	MMLU	Avg.	
Gemma-3-1B	—	—	—	35.7	10.7	26.1	24.2	—
↪ FT	**36.9**	5.7	21.3	22.5	6.7	25.2	18.1	19.4
↪ Replay	36.8	**6.3**	**21.6**	13.9	4.9	24.4	14.4	17.3
↪ CrispEdit	32.8	5.2	19.0	**34.0**	10.1	26.1	**23.4**	21.6
↪ LoRA	34.8	5.9	20.4	15.8	5.3	25.6	15.6	17.5
↪ PiSSA	35.4	5.4	20.4	11.6	3.3	26.0	13.6	16.3
↪ MiLoRA	36.0	5.9	21.0	15.4	5.5	24.8	15.2	17.5
↪ CorDA	30.0	5.1	17.6	25.7	8.2	**26.2**	20.0	19.0
↪ OPLoRA	33.7	5.1	19.4	27.3	8.6	26.0	20.6	20.1
↪ CurvLoRA	33.7	4.7	19.2	30.1	8.7	25.0	21.3	20.4
↪ CurvDelta	**36.9**	5.5	21.2	32.5	**10.2**	26.1	22.9	**22.2**
Llama-2-7B	—	—	—	64.4	26.6	45.9	45.6	—
↪ FT	59.3	**12.6**	36.0	44.9	15.2	35.1	31.7	33.4
↪ Replay	**61.2**	12.0	**36.6**	59.8	24.4	43.4	42.5	**40.2**
↪ CrispEdit	60.1	11.9	36.0	51.0	19.2	40.6	36.9	36.6
↪ LoRA	57.5	10.8	34.2	47.0	16.6	33.3	32.3	33.0
↪ PiSSA	55.8	9.9	32.9	56.7	21.6	40.3	39.5	36.9
↪ MiLoRA	58.0	10.5	34.3	59.2	23.8	44.3	42.4	39.2
↪ CorDA	52.8	7.8	30.3	57.0	21.7	41.8	40.2	36.2
↪ OPLoRA	58.6	11.2	34.9	60.2	23.6	43.2	42.3	39.4
↪ CurvLoRA	56.6	10.0	33.3	**62.9**	**26.1**	**45.1**	**44.7**	40.1
↪ CurvDelta	58.4	11.0	34.7	58.7	24.5	42.2	41.8	39.0

Table 2: **Capability learning on code generation.** Within each backbone, the best and second-best results among adaptation methods are shown in bold and underlined, respectively.

Method	Downstream tasks (%)			Retention tasks (%)				ALL
	HumanEval	MBPP	Avg.	TriviaQA	NQ-Open	MMLU	Avg.	
Gemma-3-1B	-	-	-	35.8	10.8	26.1	24.2	-
↪ FT	**20.7**	19.3	**20.0**	29.2	10.1	25.4	21.6	20.9
↪ Replay	16.5	21.7	19.1	32.3	10.1	**26.9**	23.1	21.5
↪ CrispEdit	15.9	**22.5**	19.2	**34.7**	10.5	25.7	23.6	21.9
↪ LoRA	15.2	19.3	17.3	32.4	**10.8**	26.2	23.1	20.8
↪ PiSSA	17.1	20.9	19.0	29.8	9.2	25.8	21.6	20.6
↪ MiLoRA	14.6	19.3	17.0	30.7	9.8	25.7	22.1	20.0
↪ CorDA	15.2	18.3	16.8	29.7	9.5	25.9	21.7	19.7
↪ OPLoRA	17.5	18.7	18.1	32.5	**10.8**	25.9	23.1	21.1
↪ CurvLoRA	16.5	20.6	18.6	32.7	10.3	26.3	23.1	21.3
↪ CurvDelta	17.1	22.2	19.7	34.5	10.5	26.1	**23.7**	**22.1**
Llama-2-7B	—	—	—	64.4	26.6	45.9	45.6	—
↪ FT	37.2	39.7	38.5	58.1	22.4	40.8	40.4	39.6
↪ Replay	**39.6**	41.0	40.3	58.3	22.3	41.8	40.8	40.6
↪ CrispEdit	37.8	40.7	39.3	57.8	19.5	43.5	40.3	39.9
↪ LoRA	34.1	**42.3**	38.2	62.7	**25.7**	43.5	44.0	41.7
↪ PiSSA	31.1	36.0	33.6	56.3	20.2	39.9	38.8	36.7
↪ MiLoRA	35.4	38.1	36.8	53.6	19.5	38.6	37.2	37.0
↪ CorDA	29.9	40.2	35.1	57.2	21.7	41.2	40.0	38.0
↪ OPLoRA	35.4	39.7	37.6	61.9	24.7	43.9	43.5	41.1
↪ CurvLoRA	36.0	41.8	38.9	62.4	24.8	43.3	43.5	41.7
↪ CurvDelta	**39.6**	41.3	**40.5**	**63.1**	24.9	**45.1**	**44.4**	**42.8**

the stronger joint result on Gemma. Overall, restricting updates to the low-curvature space shifts the acquisition–retention trade-off toward better general-knowledge retention; the dense variant delivers stronger joint results in most task–backbone pairs, while CurvLoRA remains competitive with fewer trainable parameters.

4.2 Selective Unlearning

4.2.1 Experimental Setup

Benchmarks and models. We conduct experiments on two widely-used LLM unlearning benchmarks: TOFU (Maini et al., 2024) and MUSE (Shi et al., 2025). TOFU provides a controlled unlearning setting based on synthetic author-profile QA pairs, whereas MUSE evaluates verbatim and knowledge memorization on real-world text corpora. For TOFU, we evaluate the Forget 5% and Forget 10% settings on three instruction-tuned backbones spanning 3B–8B scales: Llama-3.2-3B-Instruct

Table 3: Unlearning under SimNPO. CurvLoRA is the low-rank variant of CurvDelta. Left: TOFU with Llama-3.1-8B-Instruct. Right: MUSE Books with Llama-2-7B.

Method	TOFU						MUSE		
	Forget 5%			Forget 10%			Forget		Retain
	FQ ↑	FP ↓	MU ↑	FQ ↑	FP ↓	MU ↑	VerbMem ↓	KnowMem ↓	KnowMem ↑
Retrain	–	10.8	63.2	–	10.6	64.6	14.2	29.5	68.6
Original	−11.8	99.0	62.7	−26.8	99.2	62.7	99.7	46.6	69.1
↪ FT	−0.5	13.0	54.6	−1.4	14.4	38.9	23.3	27.6	46.8
↪ CrispEdit	−0.2	12.7	59.3	−1.9	16.9	42.0	14.3	29.2	53.8
↪ LoRA	−7.4	22.6	63.1	−13.5	18.2	64.6	31.6	31.1	53.4
↪ VILA	−2.2	19.8	62.9	−8.4	17.7	66.8	29.8	34.8	59.0
↪ OPLoRA	−4.5	21.6	64.5	−15.5	27.4	62.9	18.5	29.3	58.2
↪ CurvLoRA	−1.3	7.6	62.6	−3.7	3.4	62.5	20.0	31.2	57.6
↪ CurvDelta	−0.1	6.3	63.3	−1.1	6.5	63.2	11.6	32.1	55.8

(Grattafiori et al., 2024), Qwen3-4B-Instruct-2507 (Yang et al., 2025), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We omit the Forget 1% setting because it is too weak to distinguish methods. For MUSE, we use the benchmark's Llama-2-7B target model on the Books corpus. Further details are provided in Appendix E.2.

Baselines and objectives. We compare CurvLoRA and CurvDelta against five baselines: full fine-tuning (FT), CrispEdit (Ikram et al., 2026), LoRA (Hu et al., 2022), VILA (Kim et al., 2025), and OPLoRA (Xiong & Xie, 2026). FT, LoRA, VILA, and OPLoRA represent full-parameter, standard parameter-efficient, unlearning-specific parameterizations, and orthogonal delta-projection adaptation, respectively, while CrispEdit is a curvature-based gradient-projection method. Each method is evaluated with three unlearning objectives: Gradient Difference (GradDiff) (Liu et al., 2022; Maini et al., 2024), Negative Preference Optimization (NPO) (Zhang et al., 2024), and SimNPO (Fan et al., 2025). Appendices D and E.2 provide the objective definitions and optimization settings, respectively.

Evaluation. For TOFU, we report three core metrics: Forget Quality (FQ), QA probability (FP), and model utility (MU). FQ is based on a KS-test p-value that checks whether the unlearned model and a model retrained from scratch without the forget data behave similarly on the forget examples. We report the base-10 logarithm of this p-value in the TOFU tables and denote the retrained model Retrain. For MUSE, we report forget verbatim memorization (VerbMem), forget knowledge memorization (KnowMem), and retain knowledge memorization (Retain KnowMem), all measured by ROUGE.

4.2.2 Main Results

Table 3 reports the main results using SimNPO, with Llama-3.1-8B-Instruct on TOFU and Llama-2-7B on MUSE. Additional TOFU results cover two more backbones (Llama-3.2-3B-Instruct and Qwen3-4B-Instruct-2507) and two alternative objectives (GradDiff and NPO) in Appendix Tables 8–10. MUSE results with these alternative objectives appear in Appendix Table 11.

In general, CurvDelta improves the forgetting–retention balance on both TOFU and MUSE. On TOFU, CurvDelta attains the strongest Forget Quality in both forget splits and combines low forget-answer probability with model utility close to Retrain. CurvLoRA further lowers forget-answer probability in the 10% split, although its Forget Quality is lower than that of the dense variant. On MUSE, CurvDelta achieves the lowest verbatim memorization, whereas CurvLoRA retains more knowledge.

The additional results show how CurvDelta performs under GradDiff and NPO as well as SimNPO, across the three TOFU backbones and MUSE (Appendix Tables 8–10 and 11).

4.3 Ablation Studies

We study three choices in delta-space construction: the projection basis, the spectral-energy threshold γ, and the amount of preservation data used for curvature estimation. Figure 2 reports these ablations for CurvLoRA on TOFU Forget 10% with Llama-3.2-3B-Instruct and SimNPO.

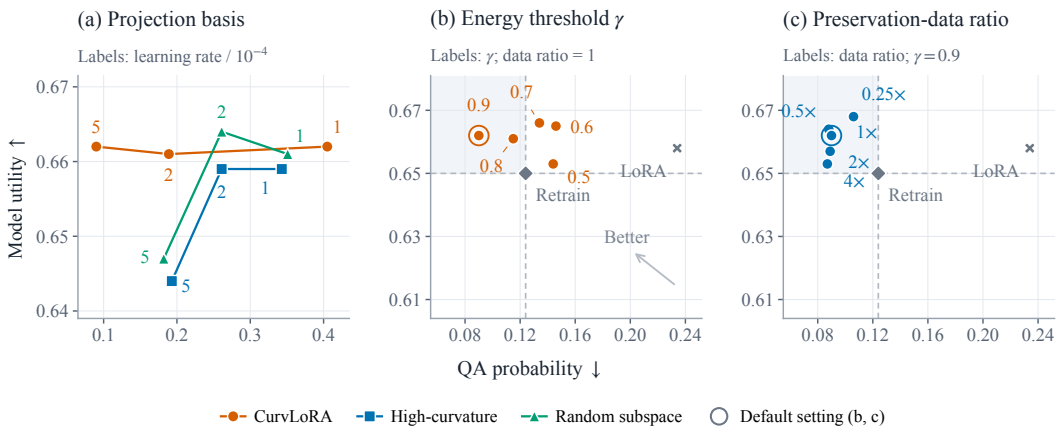


Figure 2: **Ablations of the delta-space construction.** CurvLoRA on TOFU Forget 10% with Llama-3.2-3B-Instruct and SimNPO. (a) Projection-basis choices with matched dimensions; lines connect learning-rate settings, not training steps. (b) Spectral-energy threshold γ at preservation-data ratio 1. (c) Preservation-data ratio at $\gamma = 0.9$. Rings mark the default settings in (b, c). Upper left is better; (a) uses a different axis range. Dashed lines and shading mark Retrain and the region improving on it in both metrics.

Projection basis. To isolate the effect of direction selection, we compare our low-curvature delta space with two dimension-matched variants constructed from high-curvature and random directions. As shown in Figure 2 (a), across learning rates, updates in the low-curvature space strengthen forgetting while maintaining model utility, whereas both variants lose utility at the largest learning rate while still producing weaker forgetting. These comparisons show that constructing the delta space from preservation curvature improves the forgetting–utility trade-off beyond controlling its dimension.

Spectral-energy threshold. The threshold γ jointly controls the maximum retained curvature and the dimension of the available update space. As shown in Figure 2 (b), all thresholds strengthen forgetting relative to LoRA with comparable model utility; larger γ generally further reduces QA probability while utility remains stable.

Preservation-data ratio. We estimate curvature from subsets of the retain set, with sizes from $0.25\times$ to $4\times$ that of the forget set. As shown in Figure 2 (c), the performance of CurvLoRA is robust to the amount of preservation data used for curvature estimation, indicating that our method does not require a large preservation set to estimate an effective low-curvature space.

5 Conclusion

We studied how to incorporate a constraint derived from the preservation loss into the model delta parameterization while retaining the flexibility needed for adaptation. Our answer is CurvDelta, which estimates the local curvature of the preservation loss at the pre-adaptation model with block-diagonal K-FAC, retains the trailing eigendirections of both Kronecker factors by factorwise spectral truncation, and applies a two-sided projection so that the realized delta stays in this fixed low-curvature space throughout optimization. The same geometry supports both dense and low-rank adaptation and is reusable across tasks and objectives. Because the space depends only on the base model and the preservation set, one construction is reused across task sets, task objectives, and delta parameterizations. We derive a local preservation bound of CurvDelta that depends jointly on the retained curvature and delta magnitude. In our experiments, CurvDelta attains competitive target performance on mathematical reasoning and code generation with less degradation of general capabilities than the evaluated baselines, and improves the forgetting–retention balance on TOFU and MUSE, with retention close to the retraining reference.

AI USE STATEMENT

We used generative AI tools to draft portions of the manuscript and to edit the writing for clarity, grammar, and style. The authors provided the scientific content, checked all AI-assisted text against the underlying analysis and experimental results, and revised it as needed. We did not use generative AI for research tasks requiring disclosure under the ICLR 2027 AI Policy for Authors. All authors reviewed the final manuscript and take responsibility for its content, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568/.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. doi: 10.1016/0041-5553(67)90040-7. URL https://doi.org/10.1016/0041-5553(67)90040-7.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015. doi: 10.1109/SP.2015.35. URL https://doi.org/10.1109/SP.2015.35.

Sungmin Cha, Sungjun Cho, Dasol Hwang, and Moontae Lee. Towards robust and parameter-efficient knowledge unlearning for LLMs. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1ExfUpmIW4.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2039–2050, 2020.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/02443e4e008231e0af0855f3cc70ed17-Abstract-Conference.html.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, pp. 16366–16396, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/29c8c615b3187ee995029284702d3f43-Abstract-Conference.html.

Arpit Garg, Hemanth Saratchandran, Ravi Garg, and Simon Lucey. Stable forgetting: Bounded parameter-efficient unlearning in foundation models. *arXiv preprint arXiv:2509.24166*, 2025. URL https://arxiv.org/abs/2509.24166.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju-yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2232–2241. PMLR, 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan,

Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings,

Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.77. URL https://aclanthology.org/2024.acl-long.77/.

Zarif Ikram, Arad Firouzkouhi, Stephen Tu, Mahdi Soltanolkotabi, and Paria Rashidinejad. CrispEdit: Low-curvature projections for scalable non-destructive LLM editing. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62453.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. SOUL: Unlocking the power of second-order optimization for LLM unlearning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4276–4292, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.245. URL https://aclanthology.org/2024.emnlp-main.245/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Bocheng Ju, Jianhua Wang, Chengliang Liu, and Xiaolin Chang. Null-space constrained low-rank adaptation for response-specified large language model unlearning. arXiv:2606.10989, 2026. URL https://arxiv.org/abs/2606.10989.

Yejin Kim, Eunwon Kim, Buru Chang, and Junsuk Choe. Improving fisher information estimation and efficiency for LoRA-based LLM unlearning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=mTJW8Y1nd8.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla LoRA may suffice for LLM fine-tuning. In *High-dimensional Learning Dynamics Workshop at the International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=fQ79zHBsjm.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryup8-WCW.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liang_InfLoRA_Interference-Free_Low-Rank_Adaptation_for_Continual_Learning_CVPR_2024_paper.html.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890, 2021.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pp. 243–254. PMLR, 2022. URL https://proceedings.mlr.press/v199/liu22a.html.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeepLoRA: Continual learning with residual gradient adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T3Vc5fkTzV.

M-A-P, Ge Zhang, Xinrun Du, Zhimiao Yu, Zili Wang, Zekun Wang, Shuyue Guo, Tianyu Zheng, Kang Zhu, Jerry Liu, Shawn Yue, Binbin Liu, Zhongyuan Peng, Yifan Yao, Jack Yang, Ziming Li, Bingni Zhang, Minghao Liu, Tianyu Liu, Yang Gao, Wenhu Chen, Xiaohuan Zhou, Qian Liu, Taifeng Wang, and Wenhao Huang. FineFineWeb: A comprehensive study on fine-grained domain web corpus. https://huggingface.co/datasets/m-a-p/FineFineWeb, 2024. Version 0.1.0.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=P8seBluN3c.

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

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 121038–121072, 2024. doi: 10.52202/079017-3846. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/db36f4d603cc9e3a2a5e10b93e6428f2-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0DcZxeWfOPt.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the Hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017. URL https://arxiv.org/abs/1706.04454.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4823–4836, Albuquerque, New Mexico, 2025a. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.248. URL https://aclanthology.org/2025.naacl-long.248/.

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

Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. GRU: Mitigating the trade-off between unlearning and retention for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64690–64710. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wang25de.html.

Zeguan Xiao, Siqing Li, Yong Wang, Xuetao Wei, Jian Yang, Yun Chen, and Guanhua Chen. Modeling LLM unlearning as an asymmetric two-task learning problem. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19467–19477, San Diego, California, United States, 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.890. URL https://aclanthology.org/2026.acl-long.890/.

Zeguan Xiao, Lang Mo, Yun Chen, Lei Yang, Jiehui Zhao, Lili Yang, and Guanhua Chen. Representation-guided parameter-efficient LLM unlearning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14602–14616, San Diego, California, United States, 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.717. URL https://aclanthology.org/2026.findings-acl.717/.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 71768–71791, 2024. doi: 10.52202/079017-2292. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/83f95bb0ac5046338ea2afe3390e9f4b-Abstract-Conference.html.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

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

volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. OpenCodeInterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12834–12859, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.762. URL https://aclanthology.org/2024.findings-acl.762/.

A Related Works

Preservation-aware task adaptation. Preservation-aware task adaptation addresses the challenge of acquiring capabilities required by new tasks while limiting degradation of unrelated capabilities previously acquired. Existing approaches differ in both the signal used to identify what should be preserved and the stage of adaptation they constrain. Replay-based methods train on stored examples from earlier tasks, while self-synthesized rehearsal generates proxy examples when the original data are unavailable (Rolnick et al., 2019; Huang et al., 2024). Optimization-based methods instead regularize parameters or reconcile competing objectives: EWC penalizes changes to parameters estimated to be important for earlier tasks (Kirkpatrick et al., 2017), whereas PCGrad, CAGrad, and GradDrop can mediate conflicts between task and preservation gradients when both losses are optimized jointly (Yu et al., 2020; Liu et al., 2021; Chen et al., 2020). PEFT methods constrain the update geometry in different ways. O-LoRA separates successive tasks through orthogonal low-rank subspaces (Wang et al., 2023); MiLoRA adapts minor singular components of pretrained weights; and CorDA provides a knowledge-preserved decomposition intended to limit degradation of pretrained knowledge (Wang et al., 2025a; Yang et al., 2024). A further line of work explicitly constrains LoRA updates using subspaces derived from pretrained weights, task representations, or both (Xiong & Xie, 2026; Liang & Li, 2024; Luo et al., 2026). These constructions use orthogonality to protect selected parameter or representation subspaces during adaptation. Model editing addresses a related preservation problem. ROME and MEMIT directly update selected MLP weights, while MEND learns an auxiliary editor that maps fine-tuning gradients to parameter updates (Meng et al., 2022; 2023; Mitchell et al., 2022). AlphaEdit limits collateral changes by projecting editing perturbations into the null space of preserved-knowledge representations (Fang et al., 2025). CrispEdit projects task gradients into a low-curvature space estimated with K-FAC (Ikram et al., 2026). CurvDelta derives its delta space from the local curvature of the preservation loss, relating the allowed update directions to their local preservation cost. It incorporates the projection into the delta parameterization, so that the realized delta remains in the prescribed space throughout optimization, independently of the optimizer. This combines a preservation-loss-based choice of geometry with an explicit constraint on the applied delta, supporting both dense and low-rank adaptation.

Selective unlearning. LLM unlearning seeks to suppress knowledge or behavior associated with a designated forget set while preserving utility on unrelated data, providing a practical alternative to full retraining (Cao & Yang, 2015). Existing methods control this trade-off by modifying different objects in the update process. GradDiff couples ascent on the forget-set loss with training on retained data, whereas NPO and SimNPO regulate forgetting through bounded, preference-inspired losses (Liu et al., 2022; Zhang et al., 2024; Fan et al., 2025). GRU and SAGO instead reshape the instantaneous update direction using first-order forget–retain signals: GRU projects the unlearning gradient away from directions estimated to harm retention, while SAGO treats unlearning as an asymmetric two-task problem and gates the two gradients coordinate-wise according to their sign agreement (Wang et al., 2025b; Xiao et al., 2026a). LoRA-specific methods control the content or dynamics of the low-rank update. FILA and VILA use parameter-level forget-to-retain importance to initialize adapters, whereas ReGLU uses representation geometry both to initialize a discriminative low-rank subspace and to regularize the adapter update away from retained representations (Cha et al., 2025; Kim et al., 2025; Xiao et al., 2026b). Null-space-constrained low-rank adaptation also uses benign hidden representations to restrict updates while combining safe-target response learning

with suppression of undesired responses (Ju et al., 2026). Stable Forgetting instead applies bounded transformations to LoRA adapters to control their update dynamics (Garg et al., 2025). SOUL is the second-order counterpart, using second-order information to shape each unlearning step (Jia et al., 2024). CurvDelta derives its delta space from preservation-loss curvature, providing a local loss bound in terms of the retained curvature and delta magnitude. Both CurvDelta and CurvLoRA can optimize GradDiff, NPO, or SimNPO within this space.

B Theoretical Analysis

B.1 Bregman Divergences Satisfy Assumption 1

For a twice-differentiable convex function ϕ on the output space, the associated Bregman divergence is

$$d_\phi(\boldsymbol{u}, \boldsymbol{v}) = \phi(\boldsymbol{u}) - \phi(\boldsymbol{v}) - \nabla\phi(\boldsymbol{v})^\top(\boldsymbol{u} - \boldsymbol{v}). \tag{9}$$

Substituting $\boldsymbol{u} = \boldsymbol{v}$ gives $d_\phi(\boldsymbol{v}, \boldsymbol{v}) = 0$. Differentiating in the first argument gives $\nabla_{\boldsymbol{u}} d_\phi(\boldsymbol{u}, \boldsymbol{v}) = \nabla\phi(\boldsymbol{u}) - \nabla\phi(\boldsymbol{v})$, which vanishes at $\boldsymbol{u} = \boldsymbol{v}$, and $\nabla^2_{\boldsymbol{u}} d_\phi(\boldsymbol{u}, \boldsymbol{v}) = \nabla^2\phi(\boldsymbol{u})$, which is positive semidefinite at $\boldsymbol{u} = \boldsymbol{v}$ by convexity of ϕ. Equation (9) therefore satisfies Assumption 1 for any such ϕ, with no requirement that $\boldsymbol{\theta}_0$ be a stationary point of a pretraining or proxy loss.

Two instantiations are relevant here. The quadratic function $\phi(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u}\|_2^2$ yields $d_\phi(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|_2^2$. For categorical outputs with $\boldsymbol{z}$ the logit vector and $p_{\boldsymbol{z}} = \mathrm{softmax}(\boldsymbol{z})$, the log-sum-exp function $\phi(\boldsymbol{z}) = \log\sum_k \exp(z_k)$ yields $d_\phi(\boldsymbol{z}_{\boldsymbol{\theta}}, \boldsymbol{z}_{\boldsymbol{\theta}_0}) = D_{\mathrm{KL}}(p_{\boldsymbol{\theta}_0} \| p_{\boldsymbol{\theta}})$, which is the preservation discrepancy used in all reported experiments.

B.2 The Preservation Hessian Coincides with the GGN Matrix

Write $\mathcal{L}_{\mathrm{pres}}(\boldsymbol{\theta}) = \mathbb{E}_x[d(\boldsymbol{z}_{\boldsymbol{\theta}}(x), \boldsymbol{z}_{\boldsymbol{\theta}_0}(x))]$ and abbreviate $\boldsymbol{J}(x) = \nabla_{\boldsymbol{\theta}} \boldsymbol{z}_{\boldsymbol{\theta}}(x)$. Differentiating twice in $\boldsymbol{\theta}$ and exchanging differentiation with the expectation gives

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{pres}}(\boldsymbol{\theta}) = \mathbb{E}_x\Big[\boldsymbol{J}(x)^\top \big(\nabla^2_{\boldsymbol{z}} d\big) \boldsymbol{J}(x) + \textstyle\sum_k \big(\partial_{z_k} d\big) \nabla^2_{\boldsymbol{\theta}} z_{\boldsymbol{\theta},k}(x)\Big]. \tag{10}$$

The first term is the generalized Gauss–Newton term. Evaluating Eq. (10) at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, the inner derivative $\partial_{z_k} d$ is taken at $\boldsymbol{z} = \boldsymbol{z}_{\boldsymbol{\theta}_0}(x)$, where Assumption 1 gives $\nabla_{\boldsymbol{z}} d(\boldsymbol{z}, \boldsymbol{z}_{\boldsymbol{\theta}_0}(x))|_{\boldsymbol{z}=\boldsymbol{z}_{\boldsymbol{\theta}_0}(x)} = \mathbf{0}$. The entire sum therefore vanishes regardless of the curvature of the model outputs, leaving $\boldsymbol{H}_{\mathrm{pres}} = \mathbb{E}_x[\boldsymbol{J}_0(x)^\top \boldsymbol{H}_z(x) \boldsymbol{J}_0(x)] = \boldsymbol{G}_{\mathrm{pres}}$, which is positive semidefinite because $\boldsymbol{H}_z(x) \succeq \mathbf{0}$.

B.3 Spectral Form of the Local Preservation Loss

Because $\boldsymbol{G}_{\mathrm{pres}}$ is symmetric and positive semidefinite, it admits an orthonormal eigendecomposition $\boldsymbol{G}_{\mathrm{pres}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$. Let $(\lambda_i, \boldsymbol{u}_i)$ denote its eigenpairs, ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$. Every update can be expanded in this eigenbasis as

$$\Delta\boldsymbol{\theta} = \sum_i \alpha_i \boldsymbol{u}_i, \qquad \alpha_i = \boldsymbol{u}_i^\top \Delta\boldsymbol{\theta}.$$

Using $\boldsymbol{G}_{\mathrm{pres}} \boldsymbol{u}_j = \lambda_j \boldsymbol{u}_j$ and $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \mathbf{1}[i = j]$, the quadratic form becomes

$$\begin{aligned}
\Delta\boldsymbol{\theta}^\top \boldsymbol{G}_{\mathrm{pres}} \Delta\boldsymbol{\theta} &= \sum_i \sum_j \alpha_i \alpha_j \boldsymbol{u}_i^\top \boldsymbol{G}_{\mathrm{pres}} \boldsymbol{u}_j \\
&= \sum_i \sum_j \alpha_i \alpha_j \lambda_j \boldsymbol{u}_i^\top \boldsymbol{u}_j \\
&= \sum_i \lambda_i \alpha_i^2.
\end{aligned}$$

Orthonormality also gives $\|\Delta\boldsymbol{\theta}\|_2^2 = \sum_i \alpha_i^2$. Hence, among updates with the same norm, components along larger-eigenvalue directions contribute more to the local preservation loss. In particular,

$$\lambda_{\min} \|\Delta\boldsymbol{\theta}\|_2^2 \leq \Delta\boldsymbol{\theta}^\top \boldsymbol{G}_{\mathrm{pres}} \Delta\boldsymbol{\theta} \leq \lambda_{\max} \|\Delta\boldsymbol{\theta}\|_2^2,$$

with the upper and lower bounds attained by updates aligned with leading and trailing eigenvectors, respectively.

B.4 Kronecker Structure of the Low-Curvature Delta Space

This appendix derives the delta space of Section 3.2 from the Kronecker eigenstructure of a K-FAC block and establishes the properties stated there. Consider the positive semidefinite factors with eigendecompositions $\boldsymbol{K}^{\text{in}} = \boldsymbol{U}^{\text{in}}\boldsymbol{\Lambda}^{\text{in}}(\boldsymbol{U}^{\text{in}})^\top \in \mathbb{R}^{d_{\text{in}}\times d_{\text{in}}}$ and $\boldsymbol{K}^{\text{out}} = \boldsymbol{U}^{\text{out}}\boldsymbol{\Lambda}^{\text{out}}(\boldsymbol{U}^{\text{out}})^\top \in \mathbb{R}^{d_{\text{out}}\times d_{\text{out}}}$, with both sets of eigenvalues nonnegative and ordered nonincreasingly. Let $\boldsymbol{u}_j^{\text{in}}$ and $\boldsymbol{u}_i^{\text{out}}$ denote the corresponding columns of $\boldsymbol{U}^{\text{in}}$ and $\boldsymbol{U}^{\text{out}}$. Each Kronecker eigenpair satisfies

$$(\boldsymbol{K}^{\text{in}} \otimes \boldsymbol{K}^{\text{out}})(\boldsymbol{u}_j^{\text{in}} \otimes \boldsymbol{u}_i^{\text{out}}) = \lambda_j^{\text{in}}\lambda_i^{\text{out}}(\boldsymbol{u}_j^{\text{in}} \otimes \boldsymbol{u}_i^{\text{out}}). \tag{11}$$

The factorwise energy cutoffs of Section 3.2 retain the index set and induce the curvature coefficient

$$\mathcal{R} = \{(i,j) : i > r_{\text{out}},\, j > r_{\text{in}}\}, \qquad \kappa = \lambda_{r_{\text{out}}+1}^{\text{out}}\lambda_{r_{\text{in}}+1}^{\text{in}}, \tag{12}$$

with the boundary convention $\lambda_{d_{\text{out}}+1}^{\text{out}} = \lambda_{d_{\text{in}}+1}^{\text{in}} = 0$.

The indicator of $\mathcal{R}$ factors as $\mathbf{1}[i > r_{\text{out}}]\mathbf{1}[j > r_{\text{in}}]$. Because $\boldsymbol{P}^{\text{out}}$ and $\boldsymbol{P}^{\text{in}}$ project onto the corresponding trailing factor eigenspaces,

$$\operatorname{vec}(\boldsymbol{P}^{\text{out}}\boldsymbol{\delta}\boldsymbol{P}^{\text{in}}) = (\boldsymbol{P}^{\text{in}} \otimes \boldsymbol{P}^{\text{out}})\operatorname{vec}(\boldsymbol{\delta}), \tag{13}$$

and $\boldsymbol{P}^{\text{in}} \otimes \boldsymbol{P}^{\text{out}}$ retains exactly the Kronecker eigenvectors indexed by $\mathcal{R}$. This gives Eq. (7). Both factorwise projectors in Eq. (6) are symmetric and idempotent, so $\boldsymbol{P}^{\text{in}} \otimes \boldsymbol{P}^{\text{out}}$ is symmetric and idempotent as well. The map $\boldsymbol{X} \mapsto \boldsymbol{P}^{\text{out}}\boldsymbol{X}\boldsymbol{P}^{\text{in}}$ is therefore the orthogonal projector onto the span of $\{\boldsymbol{u}_j^{\text{in}} \otimes \boldsymbol{u}_i^{\text{out}} : (i,j) \in \mathcal{R}\}$. It also cannot increase matrix rank: $\operatorname{rank}(\boldsymbol{P}^{\text{out}}\boldsymbol{X}\boldsymbol{P}^{\text{in}}) \le \operatorname{rank}(\boldsymbol{X})$.

The largest eigenvalue in the retained span is κ whenever the span is nonempty. Appendix B.6 uses this fact to bound the quadratic cost of an arbitrary delta in the span.

For nonzero-trace factors, define

$$\epsilon_{\text{out}} = \frac{\sum_{i>r_{\text{out}}}\lambda_i^{\text{out}}}{\operatorname{tr}(\boldsymbol{K}^{\text{out}})}, \qquad \epsilon_{\text{in}} = \frac{\sum_{j>r_{\text{in}}}\lambda_j^{\text{in}}}{\operatorname{tr}(\boldsymbol{K}^{\text{in}})},$$

then $\epsilon_{\text{out}}, \epsilon_{\text{in}} \le 1-\gamma$, and the sum of the retained K-FAC eigenvalues is

$$\sum_{(i,j)\in\mathcal{R}} \lambda_i^{\text{out}}\lambda_j^{\text{in}} = \epsilon_{\text{out}}\epsilon_{\text{in}}\ \operatorname{tr}(\boldsymbol{K}^{\text{out}})\operatorname{tr}(\boldsymbol{K}^{\text{in}}). \tag{14}$$

The low-curvature delta space therefore has dimension $(d_{\text{out}} - r_{\text{out}})(d_{\text{in}} - r_{\text{in}})$, and its associated eigenvalues account for at most a $(1-\gamma)^2$ fraction of the block trace.

Both quantities are monotone in γ. The cumulative energy $\sum_{i\le r}\lambda_i^{\text{out}}/\operatorname{tr}(\boldsymbol{K}^{\text{out}})$ is nondecreasing in r, so the minimum rank meeting a larger energy requirement is at least as large: $\gamma' \ge \gamma$ implies $r_{\text{out}}(\gamma') \ge r_{\text{out}}(\gamma)$, and likewise for r_{in}. Because the factor eigenvalues are nonincreasing, neither κ nor the retained dimension can increase with γ. A single threshold therefore controls the available delta capacity and the local preservation bound together.

B.5 Relation to the Full Product-Thresholded Eigenspace

Section 3.2 selects retained directions by truncating each factor spectrum. An alternative, used by CrispEdit (Ikram et al., 2026), thresholds the K-FAC product spectrum directly. For a curvature threshold $\tau \ge 0$, the product-thresholded index set is

$$\mathcal{R}_\tau^{\text{prod}} = \{(i,j) : \lambda_i^{\text{out}}\lambda_j^{\text{in}} \le \tau\}. \tag{15}$$

Let $\boldsymbol{M}_\tau \in \{0,1\}^{d_{\text{out}}\times d_{\text{in}}}$ be its indicator mask. For any matrix $\boldsymbol{X} \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, the orthogonal projection onto the corresponding eigenspace is

$$\boldsymbol{\Pi}_\tau(\boldsymbol{X}) = \boldsymbol{U}^{\text{out}}\Big(\big((\boldsymbol{U}^{\text{out}})^\top\boldsymbol{X}\boldsymbol{U}^{\text{in}}\big) \odot \boldsymbol{M}_\tau\Big)(\boldsymbol{U}^{\text{in}})^\top, \tag{16}$$

which follows by expanding $\operatorname{vec}(\boldsymbol{X})$ in the orthonormal basis of Eq. (11). Every nonzero matrix in the range of $\boldsymbol{\Pi}_\tau$ has Rayleigh quotient at most τ, and the range includes every Kronecker eigendirection

that satisfies the cutoff. Neither rule solves Eq. (4) optimally, because that problem also depends on the task objective.

Setting $\tau = \kappa$ relates the two index sets. Every $(i, j) \in \mathcal{R}$ satisfies $\lambda_i^{\text{out}} \lambda_j^{\text{in}} \leq \kappa$, and hence

$$\mathcal{R} \subseteq \mathcal{R}_\kappa^{\text{prod}}. \tag{17}$$

The inclusion can be strict, because $\mathcal{R}_\kappa^{\text{prod}}$ may contain pairs with $i \leq r_{\text{out}}$ or $j \leq r_{\text{in}}$. The omitted pairs involve at least one leading factor eigenvector; in particular, a sufficiently small eigenvalue in one factor can offset a large eigenvalue in the other. The two index sets are compared here under κ as a common product-eigenvalue threshold; γ is instead applied to the cumulative spectral energy of each factor separately and is not a threshold on the full Kronecker spectrum.

The additional directions come with a different projection structure. A generic $\boldsymbol{M}_\tau$ is not an outer product of one input mask and one output mask, so Eq. (16) uses the complete factor eigenbases, a transform into and out of them, and elementwise masking in between. Hadamard multiplication by this mask can also increase the rank of $\boldsymbol{X}$. In contrast, the factorwise mask admits the two-sided projection of Section 3.2, applied through the leading bases while preserving the delta's rank bound. Table 4 compares the cached state and projection cost.

The number of excluded pairs does not by itself determine the effect on task optimization; the relevant quantity is the task gradient's energy in those directions. Empirical Hessian studies report many flat or near-flat directions in overparameterized networks (Sagun et al., 2017; Ghorbani et al., 2019), and intrinsic-dimension experiments show that neural-network objectives can be optimized within low-dimensional parameter subspaces (Li et al., 2018); pretrained language models can likewise be adapted in low-dimensional subspaces (Aghajanyan et al., 2021) or through low-rank updates (Hu et al., 2022). These findings motivate evaluating whether the retained space carries sufficient adaptation capacity, but they do not imply alignment with a specific task gradient.

For a single step, the excluded energy can be written down exactly. Let $\boldsymbol{Q} = \nabla_{\Delta \boldsymbol{W}} \mathcal{L}_{\text{task}}$ at the current iterate. For one Euclidean gradient step on a dense pre-projection delta, the applied direction is $-\eta \boldsymbol{P}^{\text{out}} \boldsymbol{Q} \boldsymbol{P}^{\text{in}}$, and the resulting first-order change in the task loss is

$$\left\langle \boldsymbol{Q}, -\eta \boldsymbol{P}^{\text{out}} \boldsymbol{Q} \boldsymbol{P}^{\text{in}} \right\rangle_F = -\eta \left\| \boldsymbol{P}^{\text{out}} \boldsymbol{Q} \boldsymbol{P}^{\text{in}} \right\|_F^2 = -\eta \sum_{(i,j) \in \mathcal{R}} \left((\boldsymbol{u}_i^{\text{out}})^\top \boldsymbol{Q} \boldsymbol{u}_j^{\text{in}} \right)^2. \tag{18}$$

Under the same first-order approximation and step size, projection onto $\mathcal{R}_\kappa^{\text{prod}}$ would yield an additional decrease of $\eta \sum_{(i,j) \in \mathcal{R}_\kappa^{\text{prod}} \setminus \mathcal{R}} ((\boldsymbol{u}_i^{\text{out}})^\top \boldsymbol{Q} \boldsymbol{u}_j^{\text{in}})^2$. For dense Euclidean gradient descent, this is the one-step difference between the two selection rules. It does not characterize full training trajectories with adaptive optimizers or optimization over low-rank factors; the main experiments assess the task capacity of the retained space as a whole.

B.6 Low-Curvature Delta Control

The Kronecker structure established in Appendix B.4 bounds the quadratic cost of each projected layer delta. The following result depends only on the two positive semidefinite factors and their retained eigenspaces.

Proposition 2 (Low-curvature delta control)**.** *Let $\mathcal{R}$ and κ be the retained direction set and curvature coefficient of Eq.* (12)*. Every applied delta $\Delta \boldsymbol{W} = \boldsymbol{P}^{\text{out}} \boldsymbol{\delta} \boldsymbol{P}^{\text{in}}$ satisfies*

$$\operatorname{vec}(\Delta \boldsymbol{W})^\top (\boldsymbol{K}^{\text{in}} \otimes \boldsymbol{K}^{\text{out}}) \operatorname{vec}(\Delta \boldsymbol{W}) \leq \kappa \|\Delta \boldsymbol{W}\|_F^2. \tag{19}$$

Proof. By Appendix B.4, the map $\boldsymbol{X} \mapsto \boldsymbol{P}^{\text{out}} \boldsymbol{X} \boldsymbol{P}^{\text{in}}$ is the orthogonal projector onto the span of the Kronecker eigenvectors $\boldsymbol{u}_j^{\text{in}} \otimes \boldsymbol{u}_i^{\text{out}}$ with $(i, j) \in \mathcal{R}$. Expanding $\operatorname{vec}(\Delta \boldsymbol{W}) = \sum_{(i,j) \in \mathcal{R}} c_{ij} \, (\boldsymbol{u}_j^{\text{in}} \otimes \boldsymbol{u}_i^{\text{out}})$ in this orthonormal basis,

$$\operatorname{vec}(\Delta \boldsymbol{W})^\top (\boldsymbol{K}^{\text{in}} \otimes \boldsymbol{K}^{\text{out}}) \operatorname{vec}(\Delta \boldsymbol{W}) = \sum_{(i,j) \in \mathcal{R}} \lambda_i^{\text{out}} \lambda_j^{\text{in}} \, c_{ij}^2 \leq \kappa \sum_{(i,j) \in \mathcal{R}} c_{ij}^2 = \kappa \|\Delta \boldsymbol{W}\|_F^2,$$

where the inequality uses that the eigenvalues are sorted in descending order, so $\lambda_i^{\text{out}} \leq \lambda_{r_{\text{out}}+1}^{\text{out}}$ for $i > r_{\text{out}}$ and $\lambda_j^{\text{in}} \leq \lambda_{r_{\text{in}}+1}^{\text{in}}$ for $j > r_{\text{in}}$. □

When the retained span is nonempty, the bound is attained by a direction with eigenvalue κ.

B.7 Proof of Theorem 1

To aggregate the layerwise bounds into the full preservation loss, we use the following model of the curvature on the adapted weights.

Assumption 2 (Block-diagonal curvature)**.** The preservation GGN quadratic form on updates supported on $\mathcal{T}$ follows the block-diagonal K-FAC model, with blocks $\boldsymbol{K}_l^{\text{in}} \otimes \boldsymbol{K}_l^{\text{out}}$ for $l \in \mathcal{T}$. All weights outside $\mathcal{T}$ remain unchanged.

Proof. Assumption 1 gives the local quadratic approximation in Eq. (2), and Appendix B.2 identifies its Hessian with $\boldsymbol{G}_{\text{pres}}$. Under Assumption 2,

$$\begin{aligned}\Delta\boldsymbol{\theta}^\top \boldsymbol{G}_{\text{pres}} \Delta\boldsymbol{\theta} &= \sum_{l\in\mathcal{T}} \operatorname{vec}(\Delta\boldsymbol{W}_l)^\top (\boldsymbol{K}_l^{\text{in}} \otimes \boldsymbol{K}_l^{\text{out}}) \operatorname{vec}(\Delta\boldsymbol{W}_l) \\ &\le \sum_{l\in\mathcal{T}} \kappa_l \|\Delta\boldsymbol{W}_l\|_F^2,\end{aligned}$$

where the inequality applies Proposition 2 to each block. Multiplying by $1/2$ gives Eq. (8). Comparing this bound with ε establishes the sufficient condition under the local quadratic model. □

C Curvature Estimation and Computational Cost

C.1 Monte Carlo Fisher Estimation

We instantiate the preservation discrepancy with token-level $D_{\text{KL}}(p_{\boldsymbol{\theta}_0} \| p_{\boldsymbol{\theta}})$. Up to a term independent of $\boldsymbol{\theta}$, this is the expected negative log-likelihood under pseudo-targets drawn from the original model. Following standard Monte Carlo Fisher estimation in K-FAC (Martens & Grosse, 2015), for each preservation context we sample $\widetilde{y}_t \sim p_{\boldsymbol{\theta}_0}(\cdot \mid x_{<t})$ and backpropagate

$$\ell_{\text{MC}}(\boldsymbol{\theta}; x) = -\sum_t \log p_{\boldsymbol{\theta}}(\widetilde{y}_t \mid x_{<t}). \tag{20}$$

At $\boldsymbol{\theta}_0$, the expected outer product of the scores of Eq. (20) is the model Fisher, which for softmax outputs with a KL discrepancy equals the GGN matrix in Eq. (3). Accumulating the input activations and pre-activation pseudo-gradients of this backward pass yields the factors $\boldsymbol{K}_l^{\text{in}}$ and $\boldsymbol{K}_l^{\text{out}}$ of Eq. (5).

C.2 Complexity Accounting

CurvDelta constructs the preservation geometry once and applies the resulting projector during task optimization. Both CurvDelta and CurvLoRA cache the same thin eigenvector bases; their projection costs differ with the delta parameterization. We account for construction and application separately, with all layerwise costs stated for $\boldsymbol{W}_l \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$.

One-time construction and caching. Estimating the K-FAC factors requires one forward and backward pass over $\mathcal{D}_{\text{pres}}$ under Eq. (20). Per target weight, the factors occupy $O(d_{\text{out}}^2 + d_{\text{in}}^2)$ memory and their eigendecompositions cost $O(d_{\text{out}}^3 + d_{\text{in}}^3)$. After selecting the energy cutoffs, we discard the dense factors and full eigenbases, retaining only $\boldsymbol{U}_h^{\text{out}}$, $\boldsymbol{U}_h^{\text{in}}$, and κ_l. With the base model, preservation set, target weights, and threshold γ fixed, this cache is reusable across tasks, optimization hyperparameters, and both delta parameterizations.

For comparison, the exact preservation GGN for an N-parameter model requires $O(N^2)$ storage and $O(N^3)$ eigendecomposition time; even one exact weight block costs $O(d_{\text{out}}^3 d_{\text{in}}^3)$ to diagonalize. The product-thresholded projector of Eq. (16) uses the same K-FAC eigendecompositions as CurvDelta but retains the full factor eigenbases and a product mask. Table 4 compares the resulting caches and application costs.

Projection during optimization. For CurvDelta, we apply Eq. (7) through four multiplications by the cached thin bases:

$$\begin{aligned}\boldsymbol{D}_l &= \boldsymbol{\delta}_l - (\boldsymbol{\delta}_l \boldsymbol{U}_h^{\text{in}})(\boldsymbol{U}_h^{\text{in}})^\top, \\ \Delta\boldsymbol{W}_l &= \boldsymbol{D}_l - \boldsymbol{U}_h^{\text{out}}((\boldsymbol{U}_h^{\text{out}})^\top \boldsymbol{D}_l).\end{aligned}$$

Table 4: Per-weight storage and cost of one projection. Cached geometry excludes trainable deltas and temporary buffers. Application is to a dense delta except for CurvLoRA, which projects rank-r factors.

Projection	Cache storage	Projection cost
Exact GGN eigenspace	$O(d_{\text{out}}^2 d_{\text{in}}^2)$	$O(d_{\text{out}}^2 d_{\text{in}}^2)$
Product threshold	$O(d_{\text{out}}^2 + d_{\text{in}}^2 + d_{\text{out}} d_{\text{in}})$	$\Theta(d_{\text{out}} d_{\text{in}}(d_{\text{out}} + d_{\text{in}}))$
CurvDelta (dense)	$O(d_{\text{out}} r_{\text{out}} + d_{\text{in}} r_{\text{in}})$	$\Theta(d_{\text{out}} d_{\text{in}}(r_{\text{out}} + r_{\text{in}}))$
CurvLoRA (rank r)	$O(d_{\text{out}} r_{\text{out}} + d_{\text{in}} r_{\text{in}})$	$O(r(d_{\text{out}} r_{\text{out}} + d_{\text{in}} r_{\text{in}}))$

This gives the dense projection cost in Table 4. For CurvLoRA, the factorization in Section 3.3 instead applies the same thin-basis operations to $\widetilde{\boldsymbol{B}}_l = \boldsymbol{P}^{\text{out}} \boldsymbol{B}_l$ and $\widetilde{\boldsymbol{A}}_l = \boldsymbol{A}_l \boldsymbol{P}^{\text{in}}$, reducing projection work when $r \ll \min(d_{\text{out}}, d_{\text{in}})$. The projected factors occupy $O(r(d_{\text{out}} + d_{\text{in}}))$ memory, and evaluating $\widetilde{\boldsymbol{B}}_l(\widetilde{\boldsymbol{A}}_l \boldsymbol{a}_{l-1})$ costs $O(r(d_{\text{out}} + d_{\text{in}}))$ per input vector without materializing a dense delta. The table reports projection work only, rather than the cost of a full training step.

D Unlearning Objectives

In this section, we instantiate the general constrained update problem in Eq. (1) for the unlearning experiments. Here $\mathcal{D}_{\text{task}} = \mathcal{D}_f$ is the forget set and $\mathcal{D}_{\text{pres}} = \mathcal{D}_r$ is the retain set. Let $\ell(\boldsymbol{\theta}; x) = -\sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(x_t \mid x_{<t})$ denote the negative log-likelihood of a sequence x.

Gradient Difference (GradDiff). GradDiff (Liu et al., 2022; Maini et al., 2024) combines gradient ascent on the forget-set NLL with gradient descent on the retain-set NLL. Written as a minimization objective, it is

$$\mathcal{L}_{\text{GradDiff}}(\boldsymbol{\theta}) = -\mathbb{E}_{x \sim \mathcal{D}_f}\big[\ell(\boldsymbol{\theta}; x)\big] + \alpha\, \mathbb{E}_{x \sim \mathcal{D}_r}\big[\ell(\boldsymbol{\theta}; x)\big], \tag{21}$$

where $\alpha > 0$ controls the contribution of retain-set supervision.

Negative Preference Optimization (NPO). NPO (Zhang et al., 2024) replaces unbounded ascent with a reference-anchored preference objective:

$$\mathcal{L}_{\text{NPO}}(\boldsymbol{\theta}; \mathcal{D}_f) = -\frac{2}{\beta}\, \mathbb{E}_{x \sim \mathcal{D}_f} \log \sigma\left(-\beta \log \frac{p_{\boldsymbol{\theta}}(x)}{p_{\boldsymbol{\theta}_0}(x)}\right), \tag{22}$$

where σ is the logistic sigmoid and $\beta > 0$ controls the sharpness of the bounded transformation.

SimNPO. SimNPO (Fan et al., 2025) removes the reference-model ratio from NPO and applies the bounded transformation to the length-normalized forget-sequence log-likelihood:

$$\mathcal{L}_{\text{SimNPO}}(\boldsymbol{\theta}; \mathcal{D}_f) = -\frac{2}{\beta}\, \mathbb{E}_{x \sim \mathcal{D}_f} \log \sigma\left(-\frac{\beta}{|x|} \log p_{\boldsymbol{\theta}}(x) - \gamma_{\text{SimNPO}}\right), \tag{23}$$

where $|x|$ is the sequence length and γ_{SimNPO} is the margin controlling the target level of forget-sequence suppression.

E Experimental Details

E.1 Capability Learning

Section 4.1.1 summarizes the datasets and evaluation setup; here, we provide complementary details about preservation set and hyperparameter selection. For the preservation set, we sample 256 preservation examples from the FineFineWeb validation split (M-A-P et al., 2024). For training, we retrain LoRA, PiSSA, and MiLoRA using the configurations selected by Lee et al. (2026). For FT, Replay, CorDA, OPLoRA, CurvDelta, and CurvLoRA, we tune the learning rate over the grid in Table 5; the remaining settings are fixed across methods.

Table 5: Training hyperparameters and search space for capability learning.

Hyperparameter	Value / search space
Training epochs	1
Optimizer	AdamW
Learning-rate schedule	Cosine decay
Warmup ratio	0.03
Weight decay	0
Dropout	0
Maximum sequence length	512
Effective batch size	64 (Gemma-3-1B); 16 (Llama-2-7B)
LoRA rank	128 (Gemma-3-1B); 32 (Llama-2-7B)
LoRA target modules	`{q,k,v,o,gate,up,down}_proj`
Learning rate	$\{1.12, 2.00, 3.56, 6.32\} \times \{10^{-5}, 10^{-4}\}$

E.2 Selective Unlearning

Section 4.2.1 summarizes the benchmarks, models, objectives, and evaluation setup; here, we provide complementary details about data splits, curvature construction, and hyperparameter selection.

E.2.1 TOFU

We use the OpenUnlearning implementation.[1] Table 6 lists the fixed settings and search spaces used for experiments.

Table 6: Training hyperparameters and search space on TOFU. Settings are shared across the three backbones and both forget splits.

Hyperparameter	Value / search space
Training epochs	10
Optimizer	`adamw_8bit`
Effective batch size	32
Warmup epochs	1
Weight decay	0.01
LoRA rank	32
LoRA scaling factor	64
LoRA dropout	0.05
LoRA target modules	`{q,k,v,o,gate,up,down}_proj`
Curvature-energy ratio (γ)	$\{0.5, 0.6, 0.7, 0.8, 0.9\}$

E.2.2 MUSE

Table 7 lists the training settings shared across objectives. We search the learning rate over $\{1, 2, 5\} \times 10^{-5}$ for GradDiff, which has no temperature parameter. For NPO, we search learning rates in $\{2, 5, 10\} \times 10^{-5}$ and temperatures $\beta \in \{0.005, 0.01\}$. For SimNPO, we search learning rates in $\{0.5, 1, 2\} \times 10^{-4}$ and the same temperature range $\beta \in \{0.005, 0.01\}$.

Table 7: Shared training settings for FT, LoRA, and CurvLoRA on MUSE Books.

Shared hyperparameter	Value
Training epochs	5
Optimizer	`adamw_8bit`
Effective batch size	32
LoRA rank	32
LoRA scaling factor	64

[1] `https://github.com/locuslab/open-unlearning`

F Additional Results

This appendix reports the complete results across unlearning objectives and methods underlying the selective-unlearning comparisons in Section 4.2.2.

F.1 Complete TOFU Results

Tables 8–10 report the complete TOFU grid for Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct, and Qwen3-4B-Instruct-2507. Each table covers two forget splits and three losses (GradDiff, NPO, and SimNPO). As in Table 3, Retrain and Original are reference rows, then FT, CrispEdit, LoRA, VILA, OPLoRA, CurvLoRA, and CurvDelta are nested under each objective. Training configurations are specified in Appendix E.2.

Table 8: Complete TOFU results on Llama-3.2-3B-Instruct. Retrain and Original are reference rows; subsequent rows are grouped by unlearning objective.

Method	**Forget 5%**			**Forget 10%**		
	FQ ↑	FP ↓	MU ↑	FQ ↑	FP ↓	MU ↑
Retrain	–	13.1	65.9	–	12.4	65.0
Original	−13.6	95.1	66.6	−26.1	95.1	66.6
GradDiff						
↪ FT	−102.5	0.0	56.9	−220.4	0.0	39.2
↪ CrispEdit	−100.9	0.0	54.8	−202.6	0.0	48.1
↪ LoRA	−12.5	85.1	66.1	−26.4	11.6	51.3
↪ VILA	−11.8	12.4	56.2	−26.1	6.7	54.0
↪ OPLoRA	−99.3	0.0	60.2	−220.4	0.0	64.1
↪ CurvLoRA	−100.8	0.0	67.7	−173.7	3.1	67.1
↪ CurvDelta	−8.2	4.7	66.8	−173.8	0.7	67.1
NPO						
↪ FT	−1.9	10.3	47.4	−2.1	4.7	50.4
↪ CrispEdit	−1.9	4.0	60.4	−1.1	3.6	57.0
↪ LoRA	−14.7	91.2	66.7	−18.4	58.2	59.0
↪ VILA	−2.0	11.2	38.3	−9.0	15.5	41.6
↪ OPLoRA	−0.8	17.2	50.0	−0.7	7.9	64.5
↪ CurvLoRA	−0.4	13.4	66.2	−0.2	2.4	67.4
↪ CurvDelta	−1.2	4.0	66.3	−0.2	9.1	66.5
SimNPO						
↪ FT	−0.1	16.2	56.6	−1.2	16.8	58.1
↪ CrispEdit	−0.5	17.3	60.2	−0.9	19.2	61.5
↪ LoRA	−10.2	53.5	65.4	−10.9	23.4	65.8
↪ VILA	−1.8	25.3	55.7	−7.0	23.2	65.6
↪ OPLoRA	−6.1	28.0	65.3	−11.3	22.9	65.6
↪ CurvLoRA	−1.2	3.7	66.4	−2.3	13.4	66.6
↪ CurvDelta	−0.1	10.9	66.1	−0.6	9.1	65.7

Table 9: Complete TOFU results on Llama-3.1-8B-Instruct. Retrain and Original are reference rows; subsequent rows are grouped by unlearning objective.

Method	Forget 5%			Forget 10%		
	FQ ↑	FP ↓	MU ↑	FQ ↑	FP ↓	MU ↑
Retrain	–	10.8	63.2	–	10.6	64.6
Original	−11.8	99.0	62.7	−26.8	99.2	62.7
GradDiff						
↪ FT	−109.7	0.0	43.8	−23.0	0.0	0.0
↪ CrispEdit	−111.7	0.0	59.8	−190.3	1.2	47.5
↪ LoRA	−14.0	9.3	48.1	−38.7	17.0	47.3
↪ VILA	−11.2	9.9	51.4	−37.4	11.3	52.4
↪ OPLoRA	−100.9	0.0	58.2	−204.2	0.0	60.6
↪ CurvLoRA	−118.7	0.0	62.0	−236.0	0.0	63.3
↪ CurvDelta	−100.9	0.0	63.1	−228.7	0.0	65.5
NPO						
↪ FT	−1.7	7.0	54.1	−1.0	7.3	57.2
↪ CrispEdit	−4.1	2.0	63.3	−1.0	3.5	60.7
↪ LoRA	−12.9	97.2	62.7	−26.1	72.8	58.5
↪ VILA	−4.5	51.5	59.5	−9.8	18.1	26.7
↪ OPLoRA	−0.6	14.5	61.9	−8.6	5.4	68.0
↪ CurvLoRA	−0.9	2.3	61.4	−2.3	2.9	66.9
↪ CurvDelta	−0.3	5.6	62.5	−1.6	4.7	63.7
SimNPO						
↪ FT	−0.5	13.0	54.6	−1.4	14.4	38.9
↪ CrispEdit	−0.2	12.7	59.3	−1.9	16.9	42.0
↪ LoRA	−7.4	22.6	63.1	−13.5	18.2	64.6
↪ VILA	−2.2	19.8	62.9	−8.4	17.7	66.8
↪ OPLoRA	−4.5	21.6	64.5	−15.5	27.4	62.9
↪ CurvLoRA	−1.3	7.6	62.6	−3.7	3.4	62.5
↪ CurvDelta	−0.1	6.3	63.3	−1.1	6.5	63.2

Table 10: Complete TOFU results on Qwen3-4B-Instruct-2507. Retrain and Original are reference rows; subsequent rows are grouped by unlearning objective.

Method	Forget 5%			Forget 10%		
	FQ ↑	FP ↓	MU ↑	FQ ↑	FP ↓	MU ↑
Retrain	–	11.5	58.5	–	11.1	59.7
Original	−10.9	98.8	56.0	−20.8	98.9	56.0
GradDiff						
↪ FT	−3.0	0.1	51.0	−220.4	0.0	39.2
↪ CrispEdit	−104.2	0.0	60.5	−233.5	0.0	58.9
↪ LoRA	−6.3	55.8	56.0	−22.7	1.5	43.9
↪ VILA	−8.2	24.5	54.7	−20.5	26.6	55.6
↪ OPLoRA	−11.2	6.1	46.8	−220.3	0.0	56.9
↪ CurvLoRA	−82.7	4.9	54.6	−147.6	6.4	56.6
↪ CurvDelta	−2.5	1.4	56.7	−30.1	8.1	58.0
NPO						
↪ FT	−2.2	5.1	45.5	−0.1	5.2	47.2
↪ CrispEdit	−0.5	3.6	51.9	−0.6	2.7	52.7
↪ LoRA	−5.9	21.3	53.0	−9.8	10.8	56.3
↪ VILA	−10.2	91.2	56.4	−13.7	0.0	0.3
↪ OPLoRA	−3.4	23.9	50.0	−5.5	11.1	56.5
↪ CurvLoRA	−2.2	12.0	57.7	−1.3	4.7	59.6
↪ CurvDelta	−0.1	3.3	55.3	−0.1	3.9	55.7
SimNPO						
↪ FT	−1.2	18.2	44.7	−2.2	16.5	51.0
↪ CrispEdit	−1.1	14.6	52.3	−7.0	16.1	54.9
↪ LoRA	−8.2	48.1	57.7	−16.4	38.9	58.2
↪ VILA	−8.2	51.5	57.9	−15.3	32.7	57.2
↪ OPLoRA	−7.9	36.6	58.4	−15.3	29.4	59.4
↪ CurvLoRA	−1.4	10.6	57.0	−4.7	15.4	57.0
↪ CurvDelta	−0.3	8.1	55.5	−0.0	8.8	54.9

F.2 COMPLETE MUSE RESULTS

Table 11 reports every combination used to support the MUSE Books comparison in the main paper. As in Table 3, Retrain and Original are reference rows, then FT, CrispEdit, LoRA, VILA, OPLoRA, CurvLoRA, and CurvDelta are nested under each objective.

Table 11: Complete MUSE Books results with Llama-2-7B. Retrain and Original are reference rows; subsequent rows are grouped by unlearning objective.

Method	**Forget**		**Retain**
	VerbMem ↓	KnowMem ↓	KnowMem ↑
Retrain	14.2	29.5	68.6
Original	99.7	46.6	69.1
GradDiff			
↪ FT	0.0	0.0	0.0
↪ CrispEdit	88.7	39.6	60.8
↪ LoRA	89.9	40.3	57.9
↪ VILA	0.6	5.1	21.6
↪ OPLoRA	88.5	37.8	57.3
↪ CurvLoRA	20.9	28.3	61.1
↪ CurvDelta	0.5	39.7	65.5
NPO			
↪ FT	12.3	2.9	2.2
↪ CrispEdit	13.3	29.0	59.2
↪ LoRA	27.7	38.1	60.9
↪ VILA	97.3	40.8	64.0
↪ OPLoRA	26.5	29.9	53.2
↪ CurvLoRA	26.5	29.5	60.5
↪ CurvDelta	6.3	16.2	51.3
SimNPO			
↪ FT	23.3	27.6	46.8
↪ CrispEdit	14.3	29.2	53.8
↪ LoRA	31.6	31.1	53.4
↪ VILA	29.8	34.8	59.0
↪ OPLoRA	18.5	29.3	58.2
↪ CurvLoRA	20.0	31.2	57.6
↪ CurvDelta	11.6	32.1	55.8