UNMERGE: EFFICIENT MACHINE UNLEARNING VIA TASK ARITHMETIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Approximate machine unlearning seeks to remove the influence of a forget set from a trained model without full retraining. Existing gradient-based methods require data-dependent hyperparameter search, struggle when forget and retain knowledge are entangled, and offer little insight into where unlearning actually happens inside the network. We re-cast unlearning through the lens of task arithmetic: if finetuning produces a *merged* task vector $\boldsymbol{\tau}_{\mathrm{m}}$ that combines learning on forget and retain sets, unlearning is the inverse operation that subtracts a learned forget component $\boldsymbol{\tau}_{\mathcal{F}}$ to recover the retain task vector $\boldsymbol{\tau}_{\mathcal{R}}$. The forget signal is concentrated: at every layer, forget activations lie in a subspace spanned by only a handful of dominant directions. We exploit this with a low-rank factorization $\boldsymbol{\tau}_{\mathcal{F}} = \mathbf{B}Q_{\mathcal{F}}^{\top}$, which is faithful up to a small tail-eigenvalue residual and bounds how far the correction can perturb retain. We then optimize three intuitive goals—match the merged vector inside the forget span, suppress leakage into the retain span, and bound the correction size—that provably bound forget leakage and retain damage in activation space. The resulting algorithm UNMERGE is fast and powerful: on class-level unlearning with ResNet-50 on CIFAR-100 and Tiny ImageNet, it improves Tug-of-War by up to $\sim$24% over a baseline of comparable runtime and by up to $\sim$18% over stronger baselines that run $\sim 5\times$ slower, keeps membership-inference exposure at the level of retraining, and shrinks the feature-distribution gap to the retrained model, where relabeling methods leave forget features cleanly separable. Further studies show that UNMERGE also applies to ViT-S/16 and scales to Llama-3.2-3B. The per-layer basis geometry that drives the algorithm also serves as a layerwise diagnostic for when and where unlearning becomes structurally hard.

1 INTRODUCTION

Machine unlearning, which aims to remove the influence of a subset of training data from an already-trained model, has moved from an abstract ideal to an operational requirement. Regulatory frameworks such as the EU GDPR (Voigt & von dem Bussche, 2017) codify a right to be forgotten, and emerging concerns around copyright, privacy, and safety in foundation models (Li et al., 2024) demand mechanisms that delete the footprint of specific data without retraining from scratch. Exact unlearning by partition-and-retrain (Bourtoule et al., 2021) is the gold standard but is computationally prohibitive for modern deep networks, motivating a large body of *approximate* methods that edit the trained weights post-hoc. We focus on *concept-level* requests—removing a class, a semantic group or a behavior, of which the forget set is a sample—rather than the deletion of arbitrary individual records, which is a distinct problem with its own reference and metrics (§4.5).

The dominant approximate methods can be grouped into four families. *Gradient-ascent* methods (Kurmanji et al., 2023) reverse-train on the forget set with a retain regularizer; *relabeling* methods (Graves et al., 2021) replace forget labels with random or noise targets; *teacher-student* methods (Chundawat et al., 2023) distill an "incompetent teacher" into the forget region; and *saliency- or Fisher-based* methods (Golatkar et al., 2020; Fan et al., 2024; Foster et al., 2024) dampen weights estimated to encode the forget signal. All four ultimately drive a forget loss to reshape the weights, and trade forget completeness against retain damage through hyperparameter tuning.

Despite rapid progress, this line of work shares three limitations: **(i) Fragility**: gradient-ascent and relabeling variants operate on a knife-edge, where one extra epoch can flip a model from under-forgetting to catastrophic collapse, forcing per-task tuning; **(ii) Entanglement blindness**: scalar

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

forget/retain losses give the optimizer no language for reasoning about *where* forget and retain knowledge coexist, so when the two sets share features (the common case) retain damage is an unavoidable byproduct; **(iii) Opacity**: methods are evaluated almost entirely through scalar accuracy and privacy numbers and expose nothing about *why* a layer or a direction was (un)successfully unlearned, leaving diagnosis, comparison, and trust on shaky ground.

We aim to address all three by revisiting unlearning from a new angle: task arithmetic (Ilharco et al., 2023). If finetuning on $\mathcal{S} = \mathcal{R} \cup \mathcal{F}$ for retain set $\mathcal{R}$ and forget set $\mathcal{F}$ is read as implicit multi-task learning that produces a *merged* task vector $\boldsymbol{\tau}_{\mathrm{m}}$, then unlearning $\mathcal{F}$ becomes the inverse problem: identify the forget component $\boldsymbol{\tau}_{\mathcal{F}}$ inside $\boldsymbol{\tau}_{\mathrm{m}}$ and subtract it. We call this *unmerge*, and view it as the natural counterpart to recent data-free model merging (Cheng et al., 2025): the activation-subspace geometry that governs cross-task interference when combining experts also governs forget-retain leakage when separating them, with the optimization direction reversed. This single conceptual shift addresses each of the three limitations above. *Fragility* is replaced with a convex quadratic per layer, solved in a k-dimensional projected space without any gradient on data, so there is no ascent-descent collapse to guard against, and the one remaining scale α can be usually set from forget-set accuracy alone (App. D.1). *Entanglement* becomes a first-class object: forget and retain subspaces are explicitly constructed and their overlap is measurable per layer and per direction. *Opacity* dissolves because the algorithm's internal state—bases, overlaps, per-layer task-vector magnitudes—is directly inspectable.

The implications go beyond a single algorithm: unmerging recasts unlearning from a tuning problem on top of opaque weight edits into a structured decomposition problem with a principled subspace vocabulary. That vocabulary *predicts* unlearning difficulty before any update is applied, *localizes* where in a network the forget signal lives (late layers are the high-leverage region), and delimits the method's scope: a forget request with no separable activation subspace—random instance subsets being the extreme case—cannot be unmerged, and UNMERGE flags this before editing (§4.5). The diagnostics are method-agnostic, the formulation extends to concept removal and modular unlearning in larger models, and the gradient-free edit is amortizable across forget requests on the same checkpoint. Empirically, our instantiation improves the retain/forget tradeoff over gradient-based and task-vector baselines on ResNet-50 (CIFAR-100, TinyImageNet) and ViT-S (CIFAR-100), avoids the collapse mode of gradient-ascent unlearning, and matches retraining closely under membership-inference attacks on the forget set where relabeling methods leave a larger signature on the forget set. We summarize our **contributions** as follows:

- **A new formulation.** We recast approximate unlearning as the inverse of model merging and introduce UNMERGE: learning a per-layer forget task vector and subtracting it from $\boldsymbol{\tau}_{\mathrm{m}}$ so that the remainder behaves as a retain-only task vector in activation space (§3).
- **Theoretical motivation.** Building on the activation-subspace assumption of Cheng et al. (2025), we give an activation-space analysis that connects each term of our objective to an interpretable quantity: forget leakage, retain damage, and tail-direction over-expansion, positioning unmerging as the counterpart of their merging framework and inviting more rigorous theoretical work (§3.3).
- **Gradient-free unlearning.** We instantiate a low-rank procedure—a convex per-layer objective solved in a k-dimensional projected space—whose only data access is forward passes for activation statistics. It improves the retain/forget tradeoff over gradient-based and task-vector baselines on ResNet-50 (CIFAR-100, TinyImageNet) and ViT-S (CIFAR-100), avoids the collapse failure mode of gradient-ascent methods, and recovers a model closer to retraining in output space (§4). Ablations show that the retain-leakage term carries this gain, and a preliminary study carries the method over to a 3B-parameter language model (App. E.4).
- **Interpretability tools.** We introduce per-layer, per-direction diagnostics—task-vector magnitudes, forget-retain overlap, and their depth-wise profiles—that make unlearning diagnosable layer by layer. A separability score computed *before* unlearning predicts which forget requests lie beyond the reach of subspace editing; we characterize that boundary with instance-level experiments (§4.5).

2 RELATED WORK

Task arithmetic and model merging. Task vectors (Ilharco et al., 2023) are weight-space deltas induced by finetuning that support simple arithmetic for composing or suppressing behaviors without retraining. Subsequent *merging* work refines this view: weight averaging across finetuned experts (Wortsman et al., 2022; Matena & Raffel, 2022), conflict-aware pruning before averaging (Yadav et al., 2023; Yu et al., 2024), closed-form layer-activation objectives (Jin et al., 2023), and

tangent-space analyses of when task-vector linearity holds (Ortiz-Jiménez et al., 2023). Most closely, Cheng et al. (2025) show that task-vector interference is governed by the span of layer inputs and derive a data-free merging objective. Our work builds directly on this view but *reverses* it: instead of combining task vectors while minimizing cross-interference, we *unmerge*—decompose an already-merged vector by learning the sub-component that lives in a forget subspace and subtracting it. The handful of papers that compose task vectors for removal (e.g., subtracting a toxicity vector (Ilharco et al., 2023; 2022; Zhang et al., 2023)) assume an explicit forget task vector is available, whereas we *construct and learn* one from data-free bases on the finetuned model itself. Recent *task-vector unlearning* obtains that vector by finetuning on the forget set and negating it, merging several such vectors under a sign-consensus mask (Kim et al., 2025), or constraining finetuning so that merged models admit exact unlearning at scale (Kuo et al., 2026). §4.2 compares against the negated-vector and NegMerge baselines.

Machine unlearning. Motivated by right-to-be-forgotten regulations (Voigt & von dem Bussche, 2017) and the desire to remove poisoned or copyrighted data, machine unlearning (Cao & Yang, 2015; Nguyen et al., 2025; Wang et al., 2024) aims to produce a model indistinguishable from one retrained on $\mathcal{R} = \mathcal{S} \setminus \mathcal{F}$. Exact methods partition training so forgetting a point invalidates one shard (Bourtoule et al., 2021) but require training-time instrumentation. Approximate methods operate post-hoc on $\boldsymbol{W}_{\mathrm{FT}}$: influence- or Fisher-based scrubbing (Golatkar et al., 2020; 2021), gradient-ascent variants such as NegGrad+ and SCRUB (Kurmanji et al., 2023), saliency-masked updates (Fan et al., 2024), synaptic dampening (Foster et al., 2024), boundary shifting (Chen et al., 2023), amnesiac relabeling (Graves et al., 2021; Tarun et al., 2024), and distillation-based erasure that separates forgetting from retention through a teacher (Chundawat et al., 2023; Zhou et al., 2025) or adds a masked-distillation penalty against over-unlearning (Ha et al., 2026). Compared to these, UNMERGE is *data-free at optimization* (data enters only when collecting second-moment statistics), *layerwise and interpretable*, and motivated by a merging-theoretic objective with activation-space bounds (Prop. 3.3); it performs no training on $\mathcal{F}$ or $\mathcal{R}$ at unlearning time (App. B.3).

Subspace projection and representation editing. A separate thread exploits low-dimensional structure to intervene on behaviors: continual-learning subspace methods project gradients orthogonal to previous tasks (Saha et al., 2021; Farajtabar et al., 2020), while concept-erasure and knowledge-editing methods intervene on activation directions or low-rank weight subspaces (Kim et al., 2018; Belrose et al., 2023; Ravfogel et al., 2022; Meng et al., 2022). Closest to our edit are gradient-free class-unlearning methods that project weights away from forget activation subspaces (Kodge et al., 2024; Chen et al., 2024); with $\boldsymbol{W}_{\mathrm{init}}{=}0$ our update is a soft member of this family, from which it differs by acting on the task vector, by solving an explicit forgetting-vs-leakage objective instead of a hard projection, and by its supervised bases and diagnostics (App. A). We adopt Concept Activation Vectors (Kim et al., 2018) as one of our basis constructions but lift them from activation-level probing to weight-level subtraction, and unlike hard orthogonal projection we keep the natural overlap between $\boldsymbol{Q}_{\mathcal{F}}$ and $\boldsymbol{Q}_{\mathcal{R}}$ and softly penalize retain leakage via Term 2 of equation 2 (see App. A for method-specific comparisons).

3 UNLEARNING VIA UNMERGING

3.1 PRELIMINARIES

Let $\boldsymbol{W}_{\mathrm{init}} \in \{\boldsymbol{W}_0, \boldsymbol{W}_{\mathrm{PT}}\}$ be the starting weights: a public pretrained checkpoint $\boldsymbol{W}_{\mathrm{PT}}$, or training from scratch ($\boldsymbol{W}_0$), for which we take $\boldsymbol{W}_{\mathrm{init}} = 0$ rather than the random draw so that $\boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{W}_{\mathrm{FT}}$ (ablated in App. D.4). Finetuning on a train set $\mathcal{S}$ gives $\boldsymbol{W}_{\mathrm{FT}}$. The forget set $\mathcal{F} \subset \mathcal{S}$ is revealed after finetuning, and $\mathcal{R} = \mathcal{S} \setminus \mathcal{F}$ is the retain set with $|\mathcal{R}| > |\mathcal{F}|$. An unlearning algorithm maps $\boldsymbol{W}_{\mathrm{FT}}$ to $\boldsymbol{W}_{\mathrm{UL}}$ using $\mathcal{F}$ and $\mathcal{R}$; the ideal target is the model $\boldsymbol{W}_{\mathrm{RT}}$ retrained from $\boldsymbol{W}_{\mathrm{init}}$ on $\mathcal{R}$ only.

Following Task Arithmetic Ilharco et al. (2023), a task vector is the weight difference induced by training, and we define

$$\boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{W}_{\mathrm{FT}} - \boldsymbol{W}_{\mathrm{init}}, \qquad \boldsymbol{\tau}^{\star}_{\mathcal{R}} = \boldsymbol{W}_{\mathrm{RT}} - \boldsymbol{W}_{\mathrm{init}}, \qquad \boldsymbol{\tau}_{\mathcal{R}} \coloneqq \boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{\tau}_{\mathcal{F}}, \tag{1}$$

where $\boldsymbol{\tau}_{\mathrm{m}}$ is the merged task vector (since $\mathcal{S} = \mathcal{R} \cup \mathcal{F}$) and $\boldsymbol{\tau}^{\star}_{\mathcal{R}}$ the ideal retrained one. We learn a parameterized forget task vector $\boldsymbol{\tau}_{\mathcal{F}}$ such that the remainder $\boldsymbol{\tau}_{\mathcal{R}}$ behaves like a retain-only task vector in activation space; §4.5 measures how close $\boldsymbol{\tau}_{\mathcal{R}}$ lands to $\boldsymbol{\tau}^{\star}_{\mathcal{R}}$.

Why $\boldsymbol{W}_{\mathrm{FT}}$ can be treated as merged. Joint finetuning on $\mathcal{R} \cup \mathcal{F}$ is not additive, and we never assume $\boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{\tau}_{\mathcal{F}} + \boldsymbol{\tau}_{\mathcal{R}}$ for separately trained experts: $\boldsymbol{\tau}_{\mathcal{R}}$ is *defined* as the remainder, and the

objective selects the $\boldsymbol{\tau}_{\mathcal{F}}$ that makes it approximately orthogonal to the forget span. Assumption 3.1 makes this meaningful for a jointly trained vector, since the rows of $\boldsymbol{\tau}_{\mathrm{m}}$ lie in the span of the joint input activations. "Unmerging" thus names the *objective* rather than the training provenance, and unlike task arithmetic it needs no separately trained forget expert.

Parameterizing forget task vector. Cheng et al. (2025) show that the task vector at layer l of a multi-layer neural network is spanned by l-th layer's inputs. This motivates us to construct a parameterized forget task vector $\boldsymbol{\tau}_{\mathcal{F}}^l \in \mathbb{R}^{m \times d_l}$ at layer l based on forget inputs $\boldsymbol{X}_{\mathcal{F}}^l \in \mathbb{R}^{N_{\mathcal{F}} \times d_l}$ at that layer. Denote $\boldsymbol{\tau}_{\mathcal{F}}^l = \boldsymbol{B}^l {\boldsymbol{Q}_{\mathcal{F}}^l}^\top$ with a learnable matrix $\boldsymbol{B}^l \in \mathbb{R}^{m \times k}$ and a forget basis matrix $\boldsymbol{Q}_{\mathcal{F}}^l \in \mathbb{R}^{d_l \times k}$. We can construct $\boldsymbol{Q}_{\mathcal{F}}^l$ either by singular value decomposition of $\boldsymbol{X}_{\mathcal{F}}^l$ (unsupervised, high-energy basis) or by ridge-probe concept-activation vectors Kim et al. (2018) (supervised, discriminative basis). The forget projector is $\boldsymbol{P}_{\mathcal{F}}^l = \boldsymbol{Q}_{\mathcal{F}}^l {\boldsymbol{Q}_{\mathcal{F}}^l}^\top$; the retain basis $\boldsymbol{Q}_{\mathcal{R}}^l \in \mathbb{R}^{d_l \times k}$ and projector $\boldsymbol{P}_{\mathcal{R}}^l$ are built likewise from the retain inputs $\boldsymbol{X}_{\mathcal{R}}^l \in \mathbb{R}^{N_{\mathcal{R}} \times d_l}$.

3.2 UNMERGING

The objective decouples across layers. For each layer we learn $\boldsymbol{\tau}_{\mathcal{F}}^l$ such that the remainder $\boldsymbol{\tau}_{\mathcal{R}}^l = \boldsymbol{\tau}_{\mathrm{m}}^l - \boldsymbol{\tau}_{\mathcal{F}}^l$ is approximately orthogonal to the forget span and $\boldsymbol{\tau}_{\mathcal{F}}^l$ approximately orthogonal to the retain span, and we regularize the size of $\boldsymbol{\tau}_{\mathcal{F}}^l$ against over-expansion. The per-layer UNMERGE objective, and the update that applies the learned $\boldsymbol{\tau}_{\mathcal{F}}^l = \boldsymbol{B}^l {\boldsymbol{Q}_{\mathcal{F}}^l}^\top$ with a scaling $\alpha > 0$, are

$$\min_{\boldsymbol{\tau}_{\mathcal{F}}^l} \| \left(\boldsymbol{\tau}_{\mathrm{m}}^l - \boldsymbol{\tau}_{\mathcal{F}}^l\right) \boldsymbol{P}_{\mathcal{F}}^l \|_F^2 + \lambda \|\boldsymbol{\tau}_{\mathcal{F}}^l \boldsymbol{P}_{\mathcal{R}}^l\|_F^2 + \gamma \|\boldsymbol{\tau}_{\mathcal{F}}^l\|_F^2, \tag{2}$$

$$\boldsymbol{W}_{\mathrm{UL}}^l = \boldsymbol{W}_{\mathrm{FT}}^l - \alpha\, \boldsymbol{\tau}_{\mathcal{F}}^l. \tag{3}$$

With orthonormal bases, equation 2 can be reduced to a strictly convex quadratic in $\boldsymbol{B}^l$ alone, $\|\boldsymbol{\tau}_{\mathrm{m}}^l \boldsymbol{Q}_{\mathcal{F}}^l - \boldsymbol{B}^l\|_F^2 + \lambda \|\boldsymbol{B}^l \boldsymbol{C}^l\|_F^2 + \gamma \|\boldsymbol{B}^l\|_F^2$ with $\boldsymbol{C}^l = {\boldsymbol{Q}_{\mathcal{F}}^l}^\top \boldsymbol{Q}_{\mathcal{R}}^l \in \mathbb{R}^{k \times k}$: mk parameters per layer, no data access, and a retain term that acts only along the directions $\boldsymbol{Q}_{\mathcal{F}}^l$ shares with $\boldsymbol{Q}_{\mathcal{R}}^l$.

Why low rank suffices. Replacing $\boldsymbol{P}_{\mathcal{F}}^l$ and $\boldsymbol{P}_{\mathcal{R}}^l$ with the empirical activation second-moments $\frac{1}{N_{\mathcal{F}}} {\boldsymbol{X}_{\mathcal{F}}^l}^\top \boldsymbol{X}_{\mathcal{F}}^l$ and $\frac{1}{N_{\mathcal{R}}} {\boldsymbol{X}_{\mathcal{R}}^l}^\top \boldsymbol{X}_{\mathcal{R}}^l$ recovers the data-space form of the same objective, at md_l instead of mk parameters per layer and with the activations revisited at every step. The low-rank form is both tighter and safer. By Assumption 3.1, the rows of $\boldsymbol{\tau}_{\mathcal{F}}^l$ lie approximately in the span of the forget activations, so the top-k eigenbasis is faithful up to a tail residual bounded by Lemma 3.2, and the forget eigenspectrum decays sharply in the layers we consider. The rank-k form also caps how far $\boldsymbol{\tau}_{\mathcal{F}}^l$ can perturb directions outside the forget subspace, which disproportionately carry retain information: an overly large k hurts retain accuracy at the same α (App. D.1).

Implementation. UNMERGE runs in three phases—(1) one forward pass over $\mathcal{F} \cup \mathcal{R}$ builds $\boldsymbol{Q}_{\mathcal{F}}^l$ and $\boldsymbol{Q}_{\mathcal{R}}^l$; (2) $\boldsymbol{B}^l$ is initialized at the projection $\boldsymbol{B}_0^l = \boldsymbol{\tau}_{\mathrm{m}}^l \boldsymbol{Q}_{\mathcal{F}}^l$ and moved toward the minimizer of the projected quadratic under a fixed budget of T=100 Adam steps ($<$ 1s); (3) equation 3 is applied, optionally only to deep layers (`skip-layers`, §4.3)—and is *optimization-time data-free*: no gradient is ever taken on a training sample (pseudocode and details in App. B.1 and B.3).

Solver and step budget. The projected quadratic has the unique minimizer

$$\boldsymbol{B}^{l\star} = \boldsymbol{\tau}_{\mathrm{m}}^l \boldsymbol{Q}_{\mathcal{F}}^l \big((1+\gamma)\boldsymbol{I} + \lambda\, \boldsymbol{C}^l {\boldsymbol{C}^l}^\top\big)^{-1}, \tag{4}$$

a single $k \times k$ solve per layer (App. B.2). We do not run Phase 2 to convergence: the fixed budget is part of the estimator, and because Adam's normalized steps move each entry by at most $\delta = \mathrm{lr} \times T$, this motivates the approximation $\boldsymbol{B}_\delta^l = \boldsymbol{B}^{l\star} + S_\delta(\boldsymbol{B}_0^l - \boldsymbol{B}^{l\star})$ with S_δ the entrywise soft-threshold: every coefficient reaches its regularized optimum except the few largest projection coefficients, which are shrunk only by δ. Early stopping thus acts as an implicit regularizer (Yao et al., 2007); for the low-overlap CAV bases the iterate coincides with the minimizer, for PCA the residual carries part of the edit (see App. D.3 for comparison with actual iterate). Removing terms with α re-swept shows that Term 2 accounts for essentially all of the gain over the plain projection (App. D.2).

3.3 THEORETICAL MOTIVATION

We motivate the objective with activation-space bounds, adapting recent model-merging theory Cheng et al. (2025) to unlearning; the analysis is per layer and explains what each term controls.

Assumption 3.1 (Input Subspace Property Cheng et al. (2025)). *For a model finetuned with learning rate η under Lipschitz continuity, the task vector $\boldsymbol{\tau} \in \mathbb{R}^{m\times d}$ at a given layer satisfies $\boldsymbol{\tau} = \boldsymbol{C}\boldsymbol{X} + \boldsymbol{E}$ where $\boldsymbol{C} \in \mathbb{R}^{m\times N}$ is a coefficient matrix, $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ collects the layer inputs during training, and $\|\boldsymbol{E}\|_F = \mathcal{O}(\eta^2)$.*

Every SGD update of a linear layer is an outer product with that layer's inputs, so the span property is exact up to the drift of those inputs during training; the $\mathcal{O}(\eta^2)$ rate is a finetuning idealization that we use heuristically for from-scratch training (App. B.2). It motivates our low-rank parameterization:

Lemma 3.2 (Low-Rank Approximation). *Let $\boldsymbol{\Sigma}_{\mathcal{F}} = \frac{1}{N_{\mathcal{F}}}\boldsymbol{X}_{\mathcal{F}}^\top\boldsymbol{X}_{\mathcal{F}}$ be the second-moment matrix of forget inputs with eigenvalues $\nu_1 \geq \cdots \geq \nu_d \geq 0$, and let $\boldsymbol{Q}_{\mathcal{F}} \in \mathbb{R}^{d\times k}$ be the top-k eigenvectors with projector $\boldsymbol{P}_{\mathcal{F}} = \boldsymbol{Q}_{\mathcal{F}}\boldsymbol{Q}_{\mathcal{F}}{}^\top$. Under Assumption 3.1, any task vector $\boldsymbol{\tau}$ whose rows lie in $\operatorname{span}(\boldsymbol{X}_{\mathcal{F}}^\top)$ admits the decomposition $\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{Q}_{\mathcal{F}}{}^\top + \boldsymbol{E}$ with $\boldsymbol{B} = \boldsymbol{\tau}\boldsymbol{Q}_{\mathcal{F}} \in \mathbb{R}^{m\times k}$ and*

$$\|\boldsymbol{E}\|_F^2 \leq \|\boldsymbol{C}\|_{\mathrm{op}}^2 \cdot N_{\mathcal{F}} \sum_{i=k+1}^{d} \nu_i. \tag{5}$$

The approximation error of $\boldsymbol{\tau}_{\mathcal{F}} = \boldsymbol{B}\boldsymbol{Q}_{\mathcal{F}}{}^\top$ thus decays with the tail eigenvalues of the forget covariance, so rank k suffices when the forget signal is low-dimensional. The three terms of our objective, in turn, control the activation-space quantities that matter for unlearning.

Proposition 3.3 (Activation-Space Bounds). *Let $\boldsymbol{\Sigma}_{\mathcal{F}} \in \mathbb{R}^{d\times d}$ be the second-moment matrix of forget inputs with eigenvalues $\nu_1 \geq \cdots \geq \nu_d$ and top-k projector $\boldsymbol{P}_{\mathcal{F}}$. Similarly, let $\boldsymbol{\Sigma}_{\mathcal{R}}$ have eigenvalues $\mu_1 \geq \cdots \geq \mu_d$ with top-k projector $\boldsymbol{P}_{\mathcal{R}}$. Let $\alpha > 0$ be the scaling used in the update and denote $\boldsymbol{\tau}_{\mathcal{R}} = \boldsymbol{\tau}_{\mathrm{m}} - \alpha\boldsymbol{\tau}_{\mathcal{F}}$. Then:*

$$\mathbb{E}_{x\sim\mathcal{F}}\left[\|\boldsymbol{\tau}_{\mathcal{R}}\, x\|^2\right] \leq \nu_1 \left\|(\boldsymbol{\tau}_{\mathrm{m}} - \alpha\boldsymbol{\tau}_{\mathcal{F}})\boldsymbol{P}_{\mathcal{F}}\right\|_F^2 + \nu_{k+1}\left\|\boldsymbol{\tau}_{\mathrm{m}} - \alpha\boldsymbol{\tau}_{\mathcal{F}}\right\|_F^2, \tag{6}$$

$$\mathbb{E}_{x\sim\mathcal{R}}\left[\|\alpha\boldsymbol{\tau}_{\mathcal{F}}\, x\|^2\right] \leq \alpha^2\Big(\mu_1\left\|\boldsymbol{\tau}_{\mathcal{F}}\boldsymbol{P}_{\mathcal{R}}\right\|_F^2 + \mu_{k+1}\left\|\boldsymbol{\tau}_{\mathcal{F}}\right\|_F^2\Big). \tag{7}$$

Proofs are in App. B.2. Each term of the objective controls one component of these bounds:

- **Term 1** $\|(\boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{\tau}_{\mathcal{F}})\boldsymbol{P}_{\mathcal{F}}\|_F^2$ controls the dominant (ν_1-weighted) component of *forget leakage* equation 6—the response of the unlearned model on forget inputs.
- **Term 2** $\|\boldsymbol{\tau}_{\mathcal{F}}\boldsymbol{P}_{\mathcal{R}}\|_F^2$ controls the dominant (μ_1-weighted) component of *retain damage* equation 7—how much the forget correction disrupts retain performance.
- **Term 3** $\|\boldsymbol{\tau}_{\mathcal{F}}\|_F^2$ controls the tail (μ_{k+1}-weighted) component of retain damage, providing uniform protection across all directions beyond the top-k retain subspace.

Both bounds hold for any $\boldsymbol{\tau}_{\mathcal{F}}$, hence for the budgeted iterate we apply, and Terms 2 and 3 carry the factor α^2, so α trades forgetting strength against retain damage (§4.3).

Corollary 3.4 (The minimizer). *Let $\sigma_1 \geq \cdots \geq \sigma_k \in [0,1]$ be the singular values of $\boldsymbol{C} = \boldsymbol{Q}_{\mathcal{F}}^\top\boldsymbol{Q}_{\mathcal{R}}$, the cosines of the principal angles between the forget and retain spans, with left singular vectors $\boldsymbol{V}$. Then $\boldsymbol{B}^\star = \boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}}\,\boldsymbol{V}\operatorname{diag}(g_i)\boldsymbol{V}^\top$ with $g_i = 1/(1+\gamma+\lambda\sigma_i^2)$: the minimizer keeps forget-private directions ($\sigma_i \approx 0$) at gain $1/(1+\gamma)$ and shrinks directions shared with retain ($\sigma_i \approx 1$) by up to $1/(1+\gamma+\lambda)$, and the leading term of equation 7 shrinks by the factor g_i^2 along each direction.*

Term 2 therefore acts exactly where the two spans meet, which is the mechanism behind the ablation of App. D.2. The selected α exceeds the unit gain at which the first term of equation 6 is smallest, which we attribute to forget signal in its tail term; the scope of the bounds is discussed in App. B.2.

Remark 3.5 (Relation to Model Merging). *Our unmerging objective mirrors the model-merging objective of Cheng et al. (2025). For two tasks $\mathcal{F}$ and $\mathcal{R}$ with task vectors $\boldsymbol{\tau}_{\mathcal{F}}, \boldsymbol{\tau}_{\mathcal{R}}$, their merging objective seeks $\boldsymbol{\tau}_{\mathrm{m}}$ minimizing $\sum_i \|(\boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{\tau}_i)\boldsymbol{\tau}_i^\top\|_F^2$, i.e., bringing the merged vector close to each expert in its own subspace. Substituting $\boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{\tau}_{\mathcal{F}} + \boldsymbol{\tau}_{\mathcal{R}}$ yields the cross-interference terms $\|\boldsymbol{\tau}_{\mathcal{R}}\boldsymbol{\tau}_{\mathcal{F}}^\top\|_F^2 + \|\boldsymbol{\tau}_{\mathcal{F}}\boldsymbol{\tau}_{\mathcal{R}}^\top\|_F^2$. By Assumption 3.1, $\boldsymbol{\tau}_i^\top$ spans approximately the same subspace as $\boldsymbol{P}_i$, recovering our Terms 1 and 2. Merging minimizes interference when combining tasks; unmerging maximizes separation when removing one—the same subspace vocabulary with the direction reversed; the correspondence is an analogy at the level of the objectives, not a formal duality.*

4 EXPERIMENTS

4.1 SETUP

Models, data, and protocol. We use ResNet-50 (He et al., 2016) on CIFAR-100 (Krizhevsky, 2009) and TinyImageNet (Le & Yang, 2015), and ViT-S/16 (Dosovitskiy et al., 2021) on CIFAR-100 to probe architectural generality (App. E.3). Each model is finetuned from both initializations $\boldsymbol{W}_{\text{init}} \in \{\boldsymbol{W}_0, \boldsymbol{W}_{\text{PT}}\}$ to obtain $\boldsymbol{W}_{\text{FT}}$, which lets us probe how task-vector magnitude affects unmerging. Forget requests are class-level: a five-class superclass on CIFAR-100 (Trees, Aquatic Mammals, Vehicles) and a ten-class semantic group on TinyImageNet (Dogs & Cats, Arthropods); $\mathcal{F}$ is the union of the chosen classes' training images and $\mathcal{R} = \mathcal{S} \setminus \mathcal{F}$. Two instance-level protocols are introduced in §4.5; class indices, training schedules, and all configurations are in App. C.

Baselines. We compare with the retrained model $\boldsymbol{W}_{\text{RT}}$ (gold standard), with NegGrad+ (Kurmanji et al., 2023), Random Label (RL) (Graves et al., 2021), SalUn (Fan et al., 2024), the task-vector family (negated task vector (NegTV) (Ilharco et al., 2023) and NegMerge (Kim et al., 2025)), and among gradient-free subspace methods, with the activation-space projection of Kodge et al. (2024). Hyperparameters are tuned per (method, init) on Trees and Dogs & Cats and transferred to the other tasks with minimal changes, which stresses transferability (App. C). Our variants are UNMERGE-PCA and UNMERGE-CAV in KMEANS and CLASS modes (§3, Alg. 3).

Metrics. We use three evaluation pipelines, each read as a gap to retraining (the gold standard) rather than in absolute terms. **(1) Performance.** Forget, retain and held-out test accuracy, and the composite *tug-of-war* score of Zhao et al. (2024)

$$\text{ToW} = \big(1 - |\text{Acc}_{\mathcal{F}} - \text{Acc}_{\mathcal{F}}^{\text{RT}}|\big) \cdot \big(1 - |\text{Acc}_{\mathcal{R}} - \text{Acc}_{\mathcal{R}}^{\text{RT}}|\big) \cdot \big(1 - |\text{Acc}_{\text{test}} - \text{Acc}_{\text{test}}^{\text{RT}}|\big), \tag{8}$$

which equals 1 when $\boldsymbol{W}_{\text{RT}}$ is recovered exactly and penalizes over- and under-unlearning alike. **(2) Entanglement** (§4.4). Distribution distances between forget and retain features at the penultimate layer, as a relative gap to $\boldsymbol{W}_{\text{RT}}$. **(3) Privacy** (§4.5). The gap in membership-inference attack accuracy on the forget set between $\boldsymbol{W}_{\text{UL}}$ and $\boldsymbol{W}_{\text{RT}}$, following Zhao et al. (2024); Fan et al. (2024); Shokri et al. (2017); Song & Mittal (2021) (App. E.6); a gap far above retraining signals residual memorization, where far below signals over-unlearning (§4.5). App. E.2 reports three-seed μ and σ on Trees.

4.2 MAIN UNLEARNING RESULTS

Tables 1 (CIFAR-100, mean $\pm$ std across three superclasses) and 2 (TinyImageNet, Dogs & Cats) summarize the main comparisons; detailed per-superclass tables and the TinyImageNet Arthropods replication are deferred to App. E.1. UNMERGE consistently recovers retain and test accuracy close to retraining while driving forget accuracy toward the retrained level, and does so *without any gradient updates on the training data*. We highlight several observations:

Cross-task consistency. On both datasets and both initializations UNMERGE variants lead on tug-of-war (ToW); on CIFAR-100 at least two of the three variants sit above every gradient-based baseline, on TinyImageNet/$\boldsymbol{W}_0$ only CAV-CLASS does. The dispersion across superclasses (Tab. 1) is small for the leading UNMERGE cells (± 0.005–0.006 ToW) and large for the most disruptive baselines (Random Label forget accuracy $9.2 \pm 5.1\%$ on $\boldsymbol{W}_{\text{PT}}$). Over three seeds on Trees every ordering of the UNMERGE variants relative to SalUn and RL is preserved, with 1.7–$4.5\times$ smaller ToW standard deviations on $\boldsymbol{W}_0$ (App. E.2); NegTV and NegMerge trail the best variant by 0.12–0.24 ToW on every task because the vector they subtract, finetuned on $\mathcal{F}$, overlaps retain directions, which our leakage Term 2 of equation 2 penalizes; the projection of Kodge et al. (2024) is the most competitive baseline on CIFAR-100 (mean ToW $0.962/0.944$, within 0.02–0.05 of our best variant) but trails it by 0.17–0.35 on TinyImageNet, where it leaves 18% forget accuracy on $\boldsymbol{W}_0$; and on ViT-S/16 (retrained test 86.1%; App. E.3) UNMERGE-CAV-KMEANS leads every axis (ToW 0.9975 vs. 0.9949 for SalUn at $14\times$ lower runtime, MIA gap and entanglement $1.8\times$ and $7\times$ closer to retraining). On $\boldsymbol{W}_0$ where $\boldsymbol{\tau}_{\text{m}}$ is larger in deeper layers, supervision pays off more: UNMERGE-CAV-CLASS leads on both datasets (0.9909 on CIFAR-100, 0.9152 on TinyImageNet), and PCA is most exposed because its top-k eigenvectors absorb shared forget/retain directions.

Efficiency. UNMERGE takes 8–105s on one H100: the PCA variant is about $2\times$ faster than NegGrad+, the CAV variants are on par with it, and all are $\sim 5\times$ faster than Random Label and SalUn (110–113s on CIFAR-100, 666–679s on TinyImageNet), *without any retain-loop fine-tuning*. Combined with the ToW gains this is up to a $\sim 24\%$ relative lift over NegGrad+, the comparable-runtime gradient

Table 1: CIFAR-100 unlearning across three superclasses (Trees, Aquatic Mammals, Vehicles). Cells report mean ± 1 std across superclasses (sample std, $N=3$; one seed per cell, three-seed variance in App. E.2); $\boldsymbol{W}_{\mathrm{RT}}$ shows plain means. Best non-retrained in **bold**. UNMERGE variants achieve best ToW performance with noticeable short runtime. Per-superclass detail tables are in App. E.1.

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{\mathrm{PT}}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.00	0.31 ± 0.28	9.24 ± 5.09	6.12 ± 2.16	2.27 ± 2.19	3.43 ± 3.47	0.32 ± 0.14	0.11 ± 0.08	0.08 ± 0.11	0.69 ± 0.65
Retain Acc ↑	99.98	96.64 ± 1.13	99.94 ± 0.00	99.94 ± 0.04	95.10 ± 4.08	95.83 ± 3.96	99.39 ± 0.49	99.80 ± 0.09	99.63 ± 0.20	99.22 ± 0.36
Test Acc ↑	80.52	73.82 ± 0.66	80.31 ± 0.29	80.52 ± 0.60	72.81 ± 4.02	73.52 ± 4.09	77.62 ± 1.35	79.31 ± 0.25	78.33 ± 0.74	76.90 ± 0.96
ToW ↑	1.0000	0.8992 ± 0.0199	0.9053 ± 0.0485	0.9365 ± 0.0226	0.8581 ± 0.0562	0.8604 ± 0.0479	0.9622 ± 0.0150	**0.9851 ± 0.0046**	**0.9739 ± 0.0077**	0.9498 ± 0.0070
Runtime ↓		23 ± 0	110 ± 0	112 ± 0	**4 ± 0**	11 ± 0	8 ± 0	8 ± 0	26 ± 0	22 ± 0

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{0}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.00	4.53 ± 1.13	9.40 ± 6.02	3.92 ± 0.73	7.32 ± 6.86	7.84 ± 6.77	0.87 ± 0.87	0.51 ± 0.34	0.41 ± 0.36	0.43 ± 0.61
Retain Acc ↑	99.97	96.89 ± 0.75	99.96 ± 0.01	99.95 ± 0.02	95.73 ± 1.14	96.21 ± 1.15	98.35 ± 1.14	97.05 ± 1.53	99.70 ± 0.11	99.85 ± 0.03
Test Acc ↑	72.47	67.54 ± 1.49	73.95 ± 0.63	74.18 ± 0.50	66.72 ± 1.89	66.95 ± 1.88	69.19 ± 2.07	68.01 ± 2.00	71.79 ± 0.95	72.66 ± 0.82
ToW ↑	1.0000	0.8798 ± 0.0235	0.8925 ± 0.0564	0.9443 ± 0.0111	0.8370 ± 0.0722	0.8387 ± 0.0755	0.9435 ± 0.0317	0.9230 ± 0.0347	**0.9865 ± 0.0029**	**0.9909 ± 0.0062**
Runtime ↓		23 ± 1	110 ± 1	113 ± 2	**4 ± 0**	11 ± 0	8 ± 0	8 ± 2	21 ± 0	18 ± 0

baseline (the task-vector and projection baselines are as fast or faster but trail by more), and ∼18% over Random Label and SalUn at 5× their speed. Over 95% of the runtime is the forward pass and basis construction; the speedup is set by the baselines' epochs, not by model size (App. B.3).

Scaling up for Language models. Though not the focus of our study, we conduct preliminary experiments on TOFU with Llama-3.2-3B-Instruct (App. E.4) and find that UNMERGE also works on a 3B-parameter LLM: without any training (a retain-deflated PCA basis and the closed-form solver replace the vision configuration), the edited model reaches a forget quality of 0.13 at a model utility of 0.630 (retain-only model: 0.650), against 0.09 and 0.633 for RMU, the strongest default-setting baseline we ran, indicating promising potential for future extension.

Table 2: TinyImageNet unlearning results (Dogs & Cats). Best non-retrained in **bold**. UNMERGE-CAV-CLASS achieves the best ToW for both $\boldsymbol{W}_{\mathrm{init}}$ at a runtime comparable to NegGrad+; the task-vector and projection baselines are faster but trail on ToW. See Arthropods results in Tab. 13.

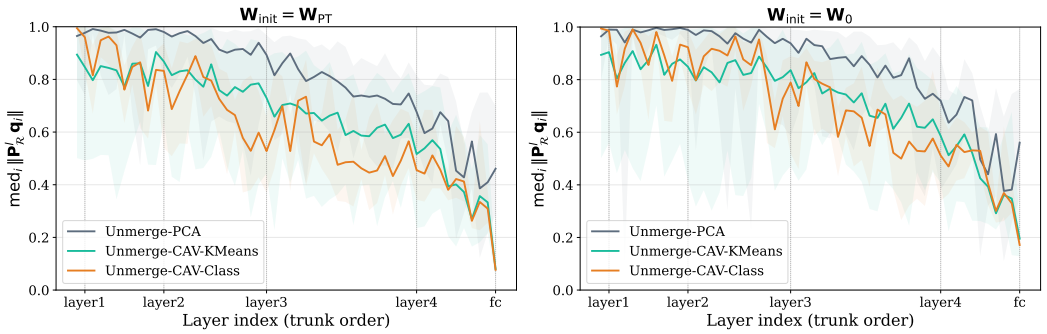
$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{\mathrm{PT}}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	1.44	16.26	15.46	0.26	0.86	1.90	0.68	0.00	0.04
Retain Acc ↑	99.98	96.36	99.98	99.98	89.70	91.62	93.30	99.37	99.96	99.94
Test Acc ↑	74.46	67.98	74.65	74.62	62.57	64.29	63.91	72.07	73.41	73.58
ToW ↑	1	0.8883	0.8358	0.8440	0.7885	0.8161	0.8189	0.9635	**0.9893**	**0.9904**
Runtime ↓		93s	666s	671s	**18s**	54s	22s	71s	105s	99s

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{0}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	0.26	9.44	8.92	8.50	9.48	17.74	4.60	2.82	0.52
Retain Acc ↑	99.98	85.64	99.98	99.97	93.29	94.06	96.39	83.72	97.87	97.88
Test Acc ↑	61.78	48.08	61.29	61.12	52.18	52.45	54.46	46.10	55.31	55.75
ToW ↑	1	0.7373	0.9011	**0.9047**	0.7718	0.7722	0.7350	0.6736	0.8898	**0.9152**
Runtime ↓		101s	670s	679s	**18s**	53s	23s	56s	86s	84s

4.3 WHERE DOES UNLEARNING HAPPEN?

Because UNMERGE builds per-layer bases $\boldsymbol{Q}^l_{\mathcal{F}}, \boldsymbol{Q}^l_{\mathcal{R}}$ and extracts $\boldsymbol{\tau}^l_{\mathcal{F}}$, we can quantify each layer's contribution. Empirically, the useful unmerging budget concentrates in the deeper layers: skipping the first 35/40 conv layers in ResNet-50 retains the forget drop while *improving* retain/test accuracy by 2–5 points. This matches the feature hierarchy of vision networks—generic edges and textures early, class-specific parts and objects late (Zeiler & Fergus, 2014; Yosinski et al., 2014; Bau et al., 2017)—and the depth-aware edits of recent class-unlearning methods (Hatami et al., 2026; Chang et al., 2026). It is a property of the request rather than of the method: a class- or concept-level forget set is separable where semantics are encoded, whereas a request defined by a low-level attribute such as a colour or texture would call for earlier layers, which the same diagnostics would indicate. Theory predicts the depth profile: early layers have larger tail eigenvalues $\sum_{i>k} \nu_i$ (Lemma 3.2) and a larger retain-damage constant μ_1 (equation 7) because generic edge/texture features dominate their second-moment spectrum, so any forget correction leaks heavily into retain; deep layers encode class-specific representations, μ_{k+1} decays, and the forget/retain spectra are more separable.

Per-layer geometry. Figs. 1 and 2 (App. D.1) make this visible on Trees. *Magnitude* (Fig. 2): the cross-init gap is depth-dependent—in layer1–2 $\|\boldsymbol{\tau}_{\mathrm{m}}\|$ is slightly larger on $\boldsymbol{W}_{\mathrm{PT}}$ (pretrained features must be shifted away from ImageNet), but at layer4 and the classifier it is ∼1.5–2× larger on $\boldsymbol{W}_0$ (peak 15.4/15.8 vs. 10.0/8.8), where the deepest task-specific weights are learned from scratch. *Entanglement* (Fig. 1): the mean per-direction overlap $\mathrm{mean}_i \, \|\boldsymbol{P}^l_{\mathcal{R}} \boldsymbol{q}_i\|$ of the forget directions with



(a) $\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$ (b) $\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$

Figure 1: Per-layer forget–retain entanglement on CIFAR-100 (Trees), ResNet-50: mean over the forget basis directions $\boldsymbol{q}_i$ of the overlap $\|\boldsymbol{P}^l_{\mathcal{R}} \boldsymbol{q}_i\|$ with the retain subspace, all bases at rank 16, seed 1 (the band is the spread over directions). This per-direction overlap is the rank-invariant form of the basis cross-norm $\|\boldsymbol{Q}^{l\top}_{\mathcal{F}} \boldsymbol{Q}^l_{\mathcal{R}}\|_F$, the structural analogue of the merging cross-interference term $\|\boldsymbol{\tau}_{\mathcal{F}} \boldsymbol{\tau}^\top_{\mathcal{R}}\|_F$ (Asm. 3.1, Remark 3.5); smaller means less entangled.

the retain subspace, all bases at rank 16; it is the quantity that enters the separability score of §4.5. It decays with depth for every basis, from 0.8–1.0 in layer1–2 to 0.4–0.6 at layer4; PCA is the most entangled at every depth (0.54/0.58 at layer4, 0.46/0.56 at the classifier on $\boldsymbol{W}_{\text{PT}}$/$\boldsymbol{W}_0$), the discriminative CAV probes suppress the shared variance that PCA's eigenvectors absorb (0.40–0.46 and 0.08–0.20), and all three bases are nearly init-invariant (block means within 0.13, 0.05 at layer4): what changes between regimes is the magnitude of $\boldsymbol{\tau}_{\text{m}}$, not the entanglement of the bases.

From diagnostics to hyperparameters. The two diagnostics fix the hyperparameters used throughout—`skip-layers` at the overlap elbow, α inversely to the deepest-layer $\|\boldsymbol{\tau}_{\text{m}}\|$, and CAV bases where entanglement is the binding constraint—and explain why UNMERGE-PCA leads on $\boldsymbol{W}_{\text{PT}}$ yet trails on $\boldsymbol{W}_0$ at the same entanglement; α itself can be line-searched on forget-set accuracy alone without $\boldsymbol{W}_{\text{RT}}$ or test data (App. D.1).

4.4 FEATURE-LEVEL ENTANGLEMENT

The per-layer overlap of Fig. 1 exists only for basis-constructing methods. To compare all methods on equal footing we measure entanglement in feature space, as the distance between the forget and retain feature distributions at the penultimate layer. Under retraining, forget images are out-of-distribution and project onto retain class manifolds, so the two clouds mix and the distance is small; in a model that has not unlearned they stay separable. An unlearned model should therefore *match the retrained distance*: Tab. 3 reports the relative gap $|d_{\text{UL}} - d_{\text{RT}}|/d_{\text{RT}}$ under four distances on a same-size retain subsample—Gaussian-RBF MMD^2 (single and multi-bandwidth), the Bures–Wasserstein W_2^2 between Gaussian fits, and Sliced-W_2^2 (App. C).

Table 3: Feature-level entanglement on CIFAR-100 (Trees): penultimate-layer distribution distance between forget and a same-size retain subsample, reported as relative gap to the retrained reference (lower is better; $\boldsymbol{W}_{\text{RT}}$ is 0 by construction). Best non-retrained per row in **bold**.

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
MMD^2	0.000	0.612	1.085	0.951	0.702	0.699	0.627	0.319	**0.255**	0.267
MMD^2-multi	0.000	0.638	2.030	1.917	0.735	0.731	0.638	0.309	**0.217**	0.257
W_2^2-Bures	0.000	0.313	0.488	**0.273**	0.522	0.505	0.548	0.553	0.499	0.611
Sliced-W_2^2	0.000	0.654	1.251	0.916	0.736	0.734	0.722	0.092	**0.046**	0.125
Mean rel. gap ↓	0.000	0.554	1.213	1.014	0.674	0.667	0.634	0.318	**0.254**	0.315

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
MMD^2	0.000	0.271	1.938	1.834	0.512	0.499	0.549	0.219	0.181	**0.038**
MMD^2-multi	0.000	0.340	3.119	3.015	0.546	0.531	0.529	0.252	0.172	**0.052**
W_2^2-Bures	0.000	0.078	1.659	1.661	0.233	0.201	0.384	0.438	**0.009**	0.117
Sliced-W_2^2	0.000	0.204	3.265	3.226	0.403	0.378	0.535	0.340	**0.071**	0.181
Mean rel. gap ↓	0.000	0.223	2.495	2.434	0.423	0.402	0.499	0.312	0.108	**0.097**

UNMERGE is the closest to retraining on both initializations (Tab. 3): the CAV variants land within ∼0.10 relative gap on $\boldsymbol{W}_0$ (0.25–0.32 on $\boldsymbol{W}_{\text{PT}}$), whereas Random Label and SalUn deviate by

1.0–2.5× the retrained reference, 3–25× the CAV gap, and Kodge et al. (2024) by 0.50–0.63. The un-edited $\boldsymbol{W}_{\mathrm{FT}}$ calibrates the axis: it sits at 0.67 on $\boldsymbol{W}_0$ and at 0.45–0.70 on TinyImageNet, and UNMERGE moves the features from there toward retraining in all three settings (0.06–0.31; Tab. 17, App. E.5), whereas relabeling pushes them *away* from retraining in every setting (on TinyImageNet/$\boldsymbol{W}_0$ the projection of Kodge et al. (2024) reaches the retrained distance too, but while still classifying 18% of the forget set and losing 7 test points, which is why this axis is read together with accuracy and privacy); on CIFAR-100/$\boldsymbol{W}_{\mathrm{PT}}$ the pretrained features already sit near retraining for every model (0.22 un-edited) and the axis has no headroom. (W_2^2-Bures on $\boldsymbol{W}_{\mathrm{PT}}$, a mean-shift-dominated distance, favors SalUn.) This exposes a *label/feature dissociation*: Random Label and SalUn drive forget accuracy down but leave the forget representations cleanly separable from retain, whereas UNMERGE lowers it while matching retraining's forget-retain feature separation. The gap measures distributional fidelity, not decodability (App. F).

4.5 MEMBERSHIP PRIVACY, SCOPE AND FIDELITY

Membership privacy. On CIFAR-100 the projection of Kodge et al. (2024) has the smallest mean MIA gap (0.017/0.028 on $\boldsymbol{W}_0$/$\boldsymbol{W}_{\mathrm{PT}}$), followed on $\boldsymbol{W}_0$ by UNMERGE-CAV-KMEANS (0.026) and CAV-CLASS (0.029) ahead of SalUn (0.047), Random Label (0.062) and NegGrad+ (0.109); on $\boldsymbol{W}_{\mathrm{PT}}$ the other methods sit within 0.067–0.093. On TinyImageNet UNMERGE sweeps $\boldsymbol{W}_{\mathrm{PT}}$ at 0.009–0.020 against 0.043–0.261 for every baseline (full details in App. E.6). Under these attacks, UNMERGE remains close to the retrained reference in membership-inference exposure at class level.

Concept-level vs. instance-level requests. UNMERGE removes what the forget samples *share*, so we test two instance-level protocols on Trees, each with its own retrained reference (App. E.7). On *coherent* subsets (half of the images of each Trees class) the aggregate scores favor UNMERGE (ToW 0.97/0.94 vs. 0.94 for a re-tuned NegGrad+ and 0.81–0.90 for relabeling), but a per-class breakdown shows a concept-level edit: the two halves of a class are i.i.d., so the edit lowers both alike, whereas relabeling protects the retained half and under-forgets the other; no method reproduces retraining. On *random* subsets no method does either: UNMERGE and NegGrad+ stay near-inert, and Random Label and SalUn score only by *marking* the forget set (attack accuracy 0.84–0.88 vs. 0.63–0.68 under retraining). The Phase-1 separability score $S = 0.17$ is consistent with both outcomes: it tells whether a request carries a concept the edit can act on. Removing the influence of given samples and removing the concept they represent are distinct problems with different references and metrics (Triantafillou et al., 2026; Cooper et al., 2025; Zhao et al., 2024), which coincide only when $\mathcal{F}$ covers the concept, and UNMERGE targets the latter.

Does unmerging recover the retrained task vector? Beyond accuracy proxies, we compare $\boldsymbol{\tau}_{\mathcal{R}} = \boldsymbol{\tau}_{\mathrm{m}} - \alpha\boldsymbol{\tau}_{\mathcal{F}}$ with $\boldsymbol{\tau}^{\star}_{\mathcal{R}} = \boldsymbol{W}_{\mathrm{RT}} - \boldsymbol{W}_{\mathrm{init}}$ directly on Trees (App. E.8). In weight space two independent retrains differ by 0.72–0.73 relative error and the un-edited $\boldsymbol{W}_{\mathrm{FT}}$ already sits at 0.75; UNMERGE stays at that floor (0.750–0.754: the minimal edit does no weight-space harm), whereas Random Label (0.82) and SalUn (0.91) move *away* from the retrained solution. Output space discriminates more sharply in the same direction: UNMERGE-CAV-KMEANS has the smallest forget-set divergence to $\boldsymbol{W}_{\mathrm{RT}}$ in both regimes (2.9 vs. 4.9–5.2 on $\boldsymbol{W}_0$, 5.3 vs. 5.9–6.6 on $\boldsymbol{W}_{\mathrm{PT}}$), because relabeling pushes forget inputs confidently to wrong classes while subspace subtraction lands closer to a model that never saw them.

5 CONCLUSION

We recast approximate machine unlearning as the inverse of model merging: UNMERGE learns a low-rank forget component of the merged task vector under a subspace objective whose terms bound forget leakage and retain damage in activation space, and subtracts it without any gradient update on the training data. On class-level forgetting with ResNet-50 and ViT-S/16 it improves the tug-of-war trade-off over gradient-based and task-vector baselines at a fraction of their cost, keeps membership-inference exposure at the level of retraining, and, unlike relabeling methods, moves the forget representations toward the retrained model rather than away from it. The per-layer bases and task-vector magnitudes double as a diagnostic that motivates the depth and scale choices and, computed before any edit, delimits the method's scope to concept-level requests; instance-level requests are a different problem that no method we tested solves (further limitations in App. F). Promising directions include language models beyond our preliminary study (App. E.4), continual forget requests, and adaptive per-layer rank and depth selection driven by the same diagnostics.

AI USE STATEMENT

In this work, we used a generative AI (Claude) for the following tasks with required disclosure: debugging, providing feedback on experimental design, implementing utility code such as plotting, and interpreting results. We have not used generative AI tools to generate synthetic data sets, for translation, or to clean or reformat data sets; qualitative and thematic data analysis is not applicable to this work. Additionally, we used the assistant to search and summarize related literature. We have reviewed and verified all AI-assisted work. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only public image-classification benchmarks (CIFAR-100, TinyImageNet) and publicly available pretrained weights, and involves no human subjects or personal data. Machine unlearning serves data-protection goals, but an approximate method must not be presented as a guarantee of deletion: we therefore evaluate against retraining on accuracy, feature and membership-inference axes, and we report the forget requests for which our method does not work (§4.5). Our finding that relabeling-based unlearning can make forget samples *more* identifiable is reported to inform practitioners. We are not aware of other ethical concerns.

REPRODUCIBILITY STATEMENT

Algorithms 1–3 give complete pseudocode, and App. B.2 contains the assumptions, proofs and the derivation of the closed-form minimizer. App. C lists the forget tasks, training schedules, baseline settings and all UNMERGE configurations; App. D.1 documents how they are chosen; App. E.2 reports seed variance; and App. D.4 identifies the pretrained checkpoint by its hash. All datasets are public, and every experiment runs on a single NVIDIA H100 GPU (runtimes in App. B.3). We will release the code, configurations and evaluation pipeline upon acceptance.

REFERENCES

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3319–3327, 2017. doi: 10.1109/CVPR.2017.354. URL https://doi.org/10.1109/CVPR.2017.354.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d066d21c619d0a78c5b557fa3291a8f4-Abstract-Conference.html.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy*, 2021. URL https://doi.org/10.1109/SP40001.2021.00019.

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

Wenhan Chang, Tianqing Zhu, Ping Xiong, Yufeng Wu, Faqian Guan, and Wanlei Zhou. Zero-shot class unlearning via layer-wise relevance analysis and neuronal path perturbation. *IEEE Transactions on Dependable and Secure Computing*, 23(4):8469–8486, 2026. doi: 10.1109/TDSC.2026.3684581. URL https://doi.org/10.1109/TDSC.2026.3684581.

Huiqiang Chen, Tianqing Zhu, Xin Yu, and Wanlei Zhou. Machine unlearning via null space calibration. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 358–366, 2024. doi: 10.24963/ijcai.2024/40. URL https://doi.org/10.24963/ijcai.2024/40.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: rapid forgetting of deep networks via shifting the decision boundary. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Chen_Boundary_Unlearning_Rapid_Forgetting_of_Deep_Networks_via_Shifting_the_CVPR_2023_paper.html.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/cheng25h.html.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI Conference on Artificial Intelligence*, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/25879.

A. Feder Cooper, Christopher A. Choquette-Choo, Miranda Bogen, Kevin Klyman, Matthew Jagielski, Katja Filippova, Ken Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, et al. Machine unlearning doesn't do what you think: Lessons for generative AI policy and research. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2412.06966.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C. Lipton, J. Zico Kolter, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. *arXiv preprint arXiv:2506.12618*, 2025. URL https://arxiv.org/abs/2506.12618.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ec4d2e436794d1bf55ca83f5ebb31887-Abstract-Conference.html.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI Conference on Artificial Intelligence*, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/29092.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: selective forgetting in deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Golatkar_Eternal_Sunshine_of_the_Spotless_Net_Selective_Forgetting_in_Deep_CVPR_2020_paper.html.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Golatkar_Mixed-Privacy_Forgetting_in_Deep_Networks_CVPR_2021_paper.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI Conference on Artificial Intelligence*, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17371.

SeungBum Ha, Saerom Park, and Sung Whan Yoon. Unlearning's blind spots: Over-unlearning and prototypical relearning attack. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2506.01318. arXiv:2506.01318 (2025).

Arman Hatami, Romina Aalishah, and Ilya E. Monosov. Class unlearning via depth-aware removal of forget-specific directions. *arXiv preprint arXiv:2604.15166*, 2026. URL https://arxiv.org/abs/2604.15166.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Gabriel Ilharco, Mitchell Wortsman, Samir Y Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bc6cddcd5d325e1c0f826066c1ad0215-Abstract-Conference.html.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Hyo Seo Kim, Dongyoon Han, and Junsuk Choe. NegMerge: Sign-consensual weight merging for machine unlearning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/kim25f.html. PMLR 267:30067–30085; arXiv:2410.05583; OpenReview: https://openreview.net/forum?id=RfiPhUB4wP; code: https://github.com/naver-ai/negmerge.

Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. Deep unlearning: Fast and efficient gradient-free class forgetting. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=BmI5p6wBi0.

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

Kevin Kuo, Amrith Setlur, Kartik Srinivas, Aditi Raghunathan, and Virginia Smith. Exact unlearning of finetuning data via model merging at scale. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2026. URL https://arxiv.org/abs/2504.04626. OpenReview: https://openreview.net/forum?id=u89LDBIyDe.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/062d711fb777322e2152435459e6e9d9-Abstract-Conference.html.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N, Stanford University*, 2015. URL https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The WMDP benchmark: measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2401.06121.

Michael S Matena and Colin A Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/70c26937fbf3d4600b69a129031b66ec-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*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025. doi: 10.1145/3749987. URL https://doi.org/10.1145/3749987.

Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d28077e5ff52034cd35b4aa15320caea-Abstract-Conference.html.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/ravfogel22a.html.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017. URL https://doi.org/10.1109/SP.2017.41.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/song.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13046–13055, 2024. doi: 10.1109/TNNLS.2023.3266233. URL https://doi.org/10.1109/TNNLS.2023.3266233.

Eleni Triantafillou, Ahmed Imtiaz Humayun, Monica Ribero, Alexander Matt Turner, Michael C. Mozer, and Georgios Kaissis. Is your algorithm unlearning or untraining? *arXiv preprint arXiv:2604.07962*, 2026. URL https://arxiv.org/abs/2604.07962.

Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR)*. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-57959-7. URL https://doi.org/10.1007/978-3-319-57959-7.

Weiqi Wang, Zhiyi Tian, Chenhan Zhang, and Shui Yu. Machine unlearning: A comprehensive survey. *arXiv preprint arXiv:2405.07406*, 2024. URL https://arxiv.org/abs/2405.07406.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Lulu Xue, Shengshan Hu, Wei Lu, Yan Shen, Dongxu Li, Peijin Guo, Ziqi Zhou, Minghui Li, Yanjun Zhang, and Leo Yu Zhang. Towards reliable forgetting: A survey on machine unlearning verification. *ACM Computing Surveys*, 58(12):1–35, 2026. doi: 10.1145/3807451. URL https://doi.org/10.1145/3807451.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. TIES-Merging: resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007. doi: 10.1007/s00365-006-0663-2. URL https://doi.org/10.1007/s00365-006-0663-2.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/532a2f85b6977104bc93f8580abbb330-Abstract.html.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super Mario: absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833, 2014. doi: 10.1007/978-3-319-10590-1_53. URL https://doi.org/10.1007/978-3-319-10590-1_53.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operation. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/299a08ee712d4752c890938da99a77c6-Abstract-Conference.html.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024. URL https://arxiv.org/abs/2404.05868.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/16e18fa3b3add076c30f2a2598f03031-Abstract-Conference.html.

Yu Zhou, Dian Zheng, Qijie Mo, Renjie Lu, Kun-Yu Lin, and Wei-Shi Zheng. Decoupled distillation to erase: A general unlearning method for any class-centric tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20350–20359, 2025. doi: 10.1109/CVPR52734.2025.01895. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhou_Decoupled_Distillation_to_Erase_A_General_Unlearning_Method_for_Any_CVPR_2025_paper.html.

APPENDIX

A Additional Related Work **16**

B Method Details **17**

B.1 Algorithm and Implementation Details 17

B.2 Proofs and Scope of the Analysis 18

B.3 Requirements, Runtime and Cost 21

C Experimental Details **21**

D Ablations of UNMERGE **23**

D.1 Per-Layer Diagnostics and Hyperparameters 23

D.2 Objective Terms 24

D.3 The Budgeted Solver and the Exact Minimizer 24

D.4 $\boldsymbol{W}_{\text{init}}$ Ablation 26

E Additional Results **26**

E.1 Detailed Per-Task Results 26

E.2 Seed Variance 27

E.3 Unlearning with ViT 27

E.4 Preliminary Study on a Language Model 28

E.5 Feature-Level Entanglement on TinyImageNet 30

E.6 Membership Inference: Detailed Results 30

E.7 Instance-Level Forgetting: Details 31

E.8 Decomposition Validation: Details 34

F Limitations and Future Work **35**

A ADDITIONAL RELATED WORK

We expand the comparison to subspace-projection and representation-editing methods that the main paper Related Work (§2) treats only briefly.

Continual learning via subspace projection. Gradient Projection Memory (Saha et al., 2021) and Orthogonal Gradient Descent (Farajtabar et al., 2020) project gradient updates onto subspaces orthogonal to previously seen tasks, preventing catastrophic forgetting by hard separation. We share their intuition that forget and retain knowledge should occupy separable subspaces, but rather than enforcing separation by hard orthogonal projection, we let the bases $\boldsymbol{Q}_{\mathcal{F}}, \boldsymbol{Q}_{\mathcal{R}}$ retain their natural overlap and penalize retain leakage softly via Term 2 of equation 2, so that partially shared directions are explicitly traded off rather than discarded by construction. This soft trade-off is what makes UNMERGE work in regimes where the forget and retain subspaces are genuinely entangled (e.g., $\boldsymbol{W}_{\text{PT}}$ where shared ImageNet-flavor variance lifts $\|\boldsymbol{Q}_{\mathcal{F}}^{\top}\boldsymbol{Q}_{\mathcal{R}}\|_F$ uniformly; see §4.3).

Gradient-free activation-subspace class unlearning. Kodge et al. (2024) estimate retain and forget spaces by SVD of layer activations on a few samples, remove the shared part from the forget space, and project the weights so that the remaining class-discriminatory directions are suppressed, selecting two scaling hyperparameters on training accuracy alone. Null-space calibration (Chen

et al., 2024) confines the unlearning update to the null space of retain activations, and the concurrent DAMP (Hatami et al., 2026) removes forget directions relative to retain prototypes with a depth-aware scale and argues that class unlearning is often carried by the classifier head. With $\boldsymbol{W}_{\text{init}}{=}0$ our update reads $\boldsymbol{W}_{\text{UL}} = \boldsymbol{W}_{\text{FT}}(\boldsymbol{I} - \alpha\, \boldsymbol{Q}_{\mathcal{F}} \boldsymbol{M} \boldsymbol{Q}_{\mathcal{F}}^{\top})$ with the $k \times k$ filter $\boldsymbol{M} = ((1+\gamma)\boldsymbol{I} + \lambda \boldsymbol{C}\boldsymbol{C}^{\top})^{-1}$ for the exact minimizer, i.e. a soft input-space projection of the same family; the hard projection $\boldsymbol{P}_{\mathcal{F}}(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}})$ of Kodge et al. (2024) corresponds to removing the retain-shared part of the forget directions instead of down-weighting it. Our formulation differs in acting on the task vector, which under $\boldsymbol{W}_{\text{PT}}$ leaves the pretrained weights intact and matters for the unsupervised basis (App. D.4); in deriving the filter from an explicit objective that trades forgetting against retain leakage; in offering supervised CAV bases; and in exposing the same quantities as diagnostics. §4 benchmarks Kodge et al. (2024) under the protocol of the other baselines (App. C).

Linear concept erasure. LEACE (Belrose et al., 2023) and RLACE (Ravfogel et al., 2022) give closed-form linear concept erasure with activation-distribution guarantees, and Concept Activation Vectors (CAV) (Kim et al., 2018) probe concept directions with linear classifiers trained on labeled positive/negative activation pools. We use CAV as one of our basis constructions (Alg. 3) and inherit its supervised, discriminative flavor, but *lift* it from representation-level probing/erasure to weight-level subtraction via task-vector decomposition: instead of projecting activations through a learned operator at inference time, we modify the weights themselves so the forget direction is no longer encoded.

Knowledge editing. ROME (Meng et al., 2022) and similar locate-and-edit methods modify specific low-rank weight subspaces to update factual associations in language models. They share our weight-level intervention philosophy but require localized (s, r, o) associations rather than distributional forget/retain sets, and target editing of specific facts rather than removal of broader behaviors. Our setting (class-level forgetting from distributional samples of $\mathcal{F}$ and $\mathcal{R}$) is complementary.

Unifying view. The retain projector $\boldsymbol{P}_{\mathcal{R}}$ in our objective plays the role of the "preserve" subspace from continual learning, while $\boldsymbol{P}_{\mathcal{F}}$ inherits the concept-direction view from CAV/LEACE. Together, the three terms of equation 2 (forget removal, retain leakage, magnitude regularization) map cleanly onto the activation-space bounds of §3.3, giving a single objective that recovers and softens individual instances of these prior approaches.

B Method Details

B.1 Algorithm and Implementation Details

This subsection expands the implementation sketch in §3 and provides the complete pseudocode. Algorithm 1 realizes equation 2 as a three-phase, *optimization-time data-free* procedure: data enters only through the second-moment statistics gathered in Phase 1, and no gradient is taken with respect to a training sample.

Phase 1: basis construction. A forward pass through $\mathcal{F} \cup \mathcal{R}$ (plus a second pass over $\mathcal{R}$ for the CAV variants) collects the layer inputs $\boldsymbol{X}_{\mathcal{F}}^{l}, \boldsymbol{X}_{\mathcal{R}}^{l}$ via forward hooks. The retain basis $\boldsymbol{Q}_{\mathcal{R}}^{l}$ is built as the top-k eigenbasis of $\frac{1}{N_{\mathcal{R}}} \boldsymbol{X}_{\mathcal{R}}^{l\,\top} \boldsymbol{X}_{\mathcal{R}}^{l}$ (PCA, Alg. 2). The forget basis $\boldsymbol{Q}_{\mathcal{F}}^{l}$ is built either by the same PCA construction or by the discriminative CAV construction in Alg. 3, which learns a ridge probe per cluster against the retain pool and stacks the orthogonalized probe directions as the basis. CAV admits two cluster definitions: *CAV-KMeans* partitions forget activations into k unsupervised clusters via k-means; *CAV-Class* uses the available forget class labels, producing one direction per forget class (so $k' = |\mathcal{Y}_{\mathcal{F}}|$ regardless of k). The two modes share the ridge-probe and orthogonalization machinery and differ only in line 5 of Alg. 3. KMeans is purely activation-driven and applicable when class labels are unavailable or when the forget set is sub-class; Class exploits supervision when one direction per forget class is the natural granularity, and matches PCA in cost since it skips the clustering step. The projection $\boldsymbol{\tau}_{\text{m}}^{l} \boldsymbol{Q}_{\mathcal{F}}^{l}$—the minimizer of Term 1 of equation 2 in isolation—and the overlap $\boldsymbol{C}^{l} = \boldsymbol{Q}_{\mathcal{F}}^{l\,\top} \boldsymbol{Q}_{\mathcal{R}}^{l}$ are computed here; they are all Phase 2 needs.

Phase 2: optimization in projected space. Substituting $\boldsymbol{\tau}_{\mathcal{F}}^{l} = \boldsymbol{B}^{l} \boldsymbol{Q}_{\mathcal{F}}^{l\,\top}$ into equation 2 reduces every term to small matrix products in the projected basis—a strictly convex quadratic in $\boldsymbol{B}^{l}$ whose exact minimizer is equation 4 (App. B.2). With $\boldsymbol{\tau}_{\text{m}}^{l} \boldsymbol{Q}_{\mathcal{F}}^{l}$ and $\boldsymbol{C}^{l}$ pre-computed once, our procedure initializes $\boldsymbol{B}^{l}$ at the projection $\boldsymbol{\tau}_{\text{m}}^{l} \boldsymbol{Q}_{\mathcal{F}}^{l}$ and takes 100 Adam steps at learning rate 10^{-3} (mk parameters per

layer, never revisiting the data). This budget does not reach the exact minimizer; App. D.3 measures the gap and shows that applying $\boldsymbol{B}^{l\star}$ directly at the paper's (λ, γ) over-shrinks $\boldsymbol{\tau}_{\mathcal{F}}$ on high-overlap bases, i.e. the fixed budget functions as an early-stopping regularizer.

Phase 3: application. We apply $\boldsymbol{W}^l_{\mathrm{UL}} = \boldsymbol{W}^l_{\mathrm{FT}} - \alpha\, \boldsymbol{B}^l {\boldsymbol{Q}^l_{\mathcal{F}}}^\top$ with a scalar α that corrects for the rank-truncation residual $\|\boldsymbol{E}\|_F$ of Lemma 3.2 and for imperfect rank. The optional `skip-layers` knob restricts the update to deeper layers, where the retain-damage constant μ_1 of equation 7 is smaller and the forget/retain spectra are more separable; §4.3 reads the α and `skip-layers` choices directly off Figs. 1 and 2.

Implementation notes. Five details of the released code differ from, or are not visible in, the description above. *(i) Centering.* Bases are computed from the *centered* covariance of the layer inputs rather than from the second moment used in §3.3; this removes the shared mean-activation direction of post-ReLU features, and the bounds of Proposition 3.3 then hold for centered inputs, with the mean response $\|\boldsymbol{\tau}_{\mathcal{R}}\, \boldsymbol{\mu}_{\mathcal{F}}\|^2$ as an additional forget term. *(ii) CAV pools.* In Alg. 3 the forget and retain pools are mean-centered *separately* before the ridge probes are fit, so each probe contrasts one cluster (or class) with the remaining activations in centered coordinates; the common forget-vs-retain mean shift is removed by construction. With $|\mathcal{Y}_{\mathcal{F}}|{=}5$ classes the class probes are nearly linearly dependent (effective rank 4), so the fifth orthogonalized direction of CAV-CLASS carries almost no signal. *(iii) Subsampling.* PCA statistics use every sample, but the CAV probes are fit on at most `max-points` activation vectors per layer (8192 on CIFAR-100, 16384 on TinyImageNet), taken from the first shuffled batches with a fixed loader seed; for a convolutional layer with L spatial positions this is `max-points`$/L$ images (e.g. 512 images at layer4), and CAV makes a second pass over the retain set. *(iv) Biases.* For an edited layer with a bias, the code shrinks the bias task vector uniformly, $b_{\mathrm{UL}} = b_{\mathrm{init}} + \big(1 - \alpha\gamma/(1+\gamma)\big)(b_{\mathrm{FT}} - b_{\mathrm{init}})$. In ResNet-50 only the classifier has a bias ($\|b_{\mathrm{FT}} - b_{\mathrm{init}}\| \le 0.15$, a logit shift below 0.02), so the rule is immaterial there; ViT-S/16 has a bias in every edited layer, and App. E.3 reports the ViT cells with and without the rule. *(v) Normalization layers.* Only convolutional and linear weights are edited. BatchNorm (LayerNorm for ViT) parameters and running statistics are kept from $\boldsymbol{W}_{\mathrm{FT}}$, activations are collected in evaluation mode, and no recalibration pass is run after the edit.

B.2 Proofs and Scope of the Analysis

Derivation of the closed form (equation 4). Write $\boldsymbol{\tau}_{\mathcal{F}} = \boldsymbol{B}\boldsymbol{Q}_{\mathcal{F}}^\top$ with $\boldsymbol{Q}_{\mathcal{F}}^\top \boldsymbol{Q}_{\mathcal{F}} = \boldsymbol{I}$ and $\boldsymbol{Q}_{\mathcal{R}}^\top \boldsymbol{Q}_{\mathcal{R}} = \boldsymbol{I}$, and let $\boldsymbol{C} = \boldsymbol{Q}_{\mathcal{F}}^\top \boldsymbol{Q}_{\mathcal{R}}$. Right-multiplication by a matrix with orthonormal rows preserves the Frobenius norm, so $\|(\boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{B}\boldsymbol{Q}_{\mathcal{F}}^\top)\boldsymbol{Q}_{\mathcal{F}}\boldsymbol{Q}_{\mathcal{F}}^\top\|_F = \|\boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}} - \boldsymbol{B}\|_F$, $\|\boldsymbol{B}\boldsymbol{Q}_{\mathcal{F}}^\top \boldsymbol{Q}_{\mathcal{R}}\boldsymbol{Q}_{\mathcal{R}}^\top\|_F = \|\boldsymbol{B}\boldsymbol{C}\|_F$, and $\|\boldsymbol{B}\boldsymbol{Q}_{\mathcal{F}}^\top\|_F = \|\boldsymbol{B}\|_F$. Equation 2 therefore equals $f(\boldsymbol{B}) = \|\boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}} - \boldsymbol{B}\|_F^2 + \lambda\|\boldsymbol{B}\boldsymbol{C}\|_F^2 + \gamma\|\boldsymbol{B}\|_F^2$, with gradient $\nabla f = -2(\boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}} - \boldsymbol{B}) + 2\lambda\boldsymbol{B}\boldsymbol{C}\boldsymbol{C}^\top + 2\gamma\boldsymbol{B}$. Setting it to zero gives $\boldsymbol{B}\big((1+\gamma)\boldsymbol{I} + \lambda\boldsymbol{C}\boldsymbol{C}^\top\big) = \boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}}$. The matrix in parentheses is symmetric positive definite, so f is strictly convex and its unique minimizer is $\boldsymbol{B}^\star = \boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}}\big((1+\gamma)\boldsymbol{I} + \lambda\boldsymbol{C}\boldsymbol{C}^\top\big)^{-1}$. For CAV bases $\boldsymbol{Q}_{\mathcal{F}}$ has $k' \le k$ orthonormal columns and the derivation is unchanged with $\boldsymbol{C} \in \mathbb{R}^{k'\times k}$. The 100-step Adam iterate used for the reported runs does *not* reach $\boldsymbol{B}^\star$ in general; App. D.3 quantifies the distance and the effect of applying $\boldsymbol{B}^\star$ itself.

Proof of Lemma 3.2. Let $\boldsymbol{\Sigma}_{\mathcal{F}} = \frac{1}{N_{\mathcal{F}}}\boldsymbol{X}_{\mathcal{F}}^\top \boldsymbol{X}_{\mathcal{F}}$ be the second-moment matrix of forget inputs with eigenvalues $\nu_1 \ge \cdots \ge \nu_d \ge 0$, and let $\boldsymbol{Q}_{\mathcal{F}} \in \mathbb{R}^{d\times k}$ be the top-k eigenvectors with projector $\boldsymbol{P}_{\mathcal{F}} = \boldsymbol{Q}_{\mathcal{F}}{\boldsymbol{Q}_{\mathcal{F}}}^\top$. Under Assumption 3.1, any task vector $\boldsymbol{\tau}$ whose rows lie in $\mathrm{span}(\boldsymbol{X}_{\mathcal{F}}^\top)$ admits the decomposition $\boldsymbol{\tau} = \boldsymbol{B}{\boldsymbol{Q}_{\mathcal{F}}}^\top + \boldsymbol{E}$ with $\boldsymbol{B} = \boldsymbol{\tau}\boldsymbol{Q}_{\mathcal{F}} \in \mathbb{R}^{m\times k}$ and

$$\|\boldsymbol{E}\|_F^2 \le \|\boldsymbol{C}\|_{\mathrm{op}}^2 \cdot N_{\mathcal{F}} \sum_{i=k+1}^{d} \nu_i. \tag{9}$$

Proof. Write $\boldsymbol{\tau} = \boldsymbol{C}\boldsymbol{X}_{\mathcal{F}}$ (neglecting $\mathcal{O}(\eta^2)$ terms). Decompose $\boldsymbol{\tau} = \underbrace{\boldsymbol{C}\boldsymbol{X}_{\mathcal{F}}\boldsymbol{Q}_{\mathcal{F}}}_{\boldsymbol{B}}{\boldsymbol{Q}_{\mathcal{F}}}^\top + \underbrace{\boldsymbol{C}\boldsymbol{X}_{\mathcal{F}}(I - \boldsymbol{P}_{\mathcal{F}})}_{\boldsymbol{E}}$. By sub-multiplicativity, $\|\boldsymbol{E}\|_F \le \|\boldsymbol{C}\|_{\mathrm{op}}\|\boldsymbol{X}_{\mathcal{F}}(I - \boldsymbol{P}_{\mathcal{F}})\|_F$, and $\|\boldsymbol{X}_{\mathcal{F}}(I - \boldsymbol{P}_{\mathcal{F}})\|_F^2 = \mathrm{tr}\big((I - \boldsymbol{P}_{\mathcal{F}})\boldsymbol{X}_{\mathcal{F}}^\top \boldsymbol{X}_{\mathcal{F}}(I - \boldsymbol{P}_{\mathcal{F}})\big) = N_{\mathcal{F}}\sum_{i>k}\nu_i$. □

Algorithm 1 Unmerging

Input: Finetuned model $\boldsymbol{W}_{\mathrm{FT}}$, initial weights $\boldsymbol{W}_{\mathrm{init}}$, forget set $\mathcal{F}$, retain set $\mathcal{R}$, rank k, regularization λ, γ, scaling α, learning rate η, steps T, basis type $\in \{\mathrm{PCA}, \mathrm{CAV}\}$, CAV mode $\in \{\mathrm{kmeans}, \mathrm{class}\}$ (if basis=CAV)
Output: Unlearned model $\boldsymbol{W}_{\mathrm{UL}}$

for each target layer l **do**
  $\boldsymbol{\tau}_{\mathrm{m}}^{l} \leftarrow \boldsymbol{W}_{\mathrm{FT}}^{l} - \boldsymbol{W}_{\mathrm{init}}^{l}$
  Collect layer inputs $\boldsymbol{X}_{\mathcal{R}}^{l}, \boldsymbol{X}_{\mathcal{F}}^{l}$ via forward pass
  $\boldsymbol{Q}_{\mathcal{R}}^{l} \leftarrow \textsc{PcaBasis}(\boldsymbol{X}_{\mathcal{R}}^{l}, k)$
  if basis type is PCA **then**
    $\boldsymbol{Q}_{\mathcal{F}}^{l} \leftarrow \textsc{PcaBasis}(\boldsymbol{X}_{\mathcal{F}}^{l}, k)$
  else
    $\boldsymbol{Q}_{\mathcal{F}}^{l} \leftarrow \textsc{CavBasis}(\boldsymbol{X}_{\mathcal{F}}^{l}, \boldsymbol{X}_{\mathcal{R}}^{l}, k, \mathrm{mode})$
  end if
  $\boldsymbol{C}^{l} \leftarrow {\boldsymbol{Q}_{\mathcal{F}}^{l}}^{\top} \boldsymbol{Q}_{\mathcal{R}}^{l}$ $\{k \times k$ basis overlap$\}$
end for

$\boldsymbol{B}^{l} \leftarrow \boldsymbol{\tau}_{\mathrm{m}}^{l} \boldsymbol{Q}_{\mathcal{F}}^{l}$ for all l {initialize at the projection, the minimizer of Term 1}
{T Adam steps in projected space: with $\boldsymbol{\tau}_{\mathcal{F}}^{l} = \boldsymbol{B}^{l} {\boldsymbol{Q}_{\mathcal{F}}^{l}}^{\top}$, equation 2 is the strictly convex quadratic below, whose exact minimizer is equation 4; the fixed budget T early-stops (App. D.3)}
for $t = 1$ **to** T **do**
  $\mathcal{L} \leftarrow \sum_{l} \left\| \boldsymbol{\tau}_{\mathrm{m}}^{l} \boldsymbol{Q}_{\mathcal{F}}^{l} - \boldsymbol{B}^{l} \right\|_{F}^{2} + \lambda \left\| \boldsymbol{B}^{l} \boldsymbol{C}^{l} \right\|_{F}^{2} + \gamma \left\| \boldsymbol{B}^{l} \right\|_{F}^{2}$
  $\boldsymbol{B}^{l} \leftarrow \mathrm{Adam}\big(\boldsymbol{B}^{l}, \nabla_{\boldsymbol{B}^{l}} \mathcal{L}, \eta\big)$ for all l
end for

{Recover $\boldsymbol{\tau}_{\mathcal{F}}$ and apply}
for each target layer l **do**
  $\boldsymbol{\tau}_{\mathcal{F}}^{l} \leftarrow \boldsymbol{B}^{l} {\boldsymbol{Q}_{\mathcal{F}}^{l}}^{\top}$
  $\boldsymbol{W}_{\mathrm{UL}}^{l} \leftarrow \boldsymbol{W}_{\mathrm{FT}}^{l} - \alpha\, \boldsymbol{\tau}_{\mathcal{F}}^{l}$
end for
return $\boldsymbol{W}_{\mathrm{UL}}$

Algorithm 2 PCA Basis Construction (PcaBasis)

Input: Layer inputs $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, rank k
Output: Orthonormal basis $Q \in \mathbb{R}^{d \times k}$

$\boldsymbol{\mu} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$
$\boldsymbol{C} \leftarrow \boldsymbol{X}^{\top} \boldsymbol{X} - N\, \boldsymbol{\mu} \boldsymbol{\mu}^{\top}$ {centered covariance}
$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \leftarrow$ top-k eigenvectors of $\boldsymbol{C}$
$Q \leftarrow [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$
return Q

Proof of Proposition 3.3. Let $\boldsymbol{\Sigma}_{\mathcal{F}} \in \mathbb{R}^{d \times d}$ be the second-moment matrix of forget inputs with eigenvalues $\nu_1 \geq \cdots \geq \nu_d$ and top-k projector $\boldsymbol{P}_{\mathcal{F}}$. Similarly, let $\boldsymbol{\Sigma}_{\mathcal{R}}$ have eigenvalues $\mu_1 \geq \cdots \geq \mu_d$ with top-k projector $\boldsymbol{P}_{\mathcal{R}}$. Let $\alpha > 0$ be the scaling used in the update and denote $\boldsymbol{\tau}_{\mathcal{R}} = \boldsymbol{\tau}_{\mathrm{m}} - \alpha \boldsymbol{\tau}_{\mathcal{F}}$. Then:

$$\mathbb{E}_{x \sim \mathcal{F}} \left[\left\| \boldsymbol{\tau}_{\mathcal{R}}\, x \right\|^2 \right] \leq \nu_1 \left\| (\boldsymbol{\tau}_{\mathrm{m}} - \alpha \boldsymbol{\tau}_{\mathcal{F}}) \boldsymbol{P}_{\mathcal{F}} \right\|_{F}^{2} + \nu_{k+1} \left\| \boldsymbol{\tau}_{\mathrm{m}} - \alpha \boldsymbol{\tau}_{\mathcal{F}} \right\|_{F}^{2}, \tag{10}$$

$$\mathbb{E}_{x \sim \mathcal{R}} \left[\left\| \alpha \boldsymbol{\tau}_{\mathcal{F}}\, x \right\|^2 \right] \leq \alpha^2 \Big(\mu_1 \left\| \boldsymbol{\tau}_{\mathcal{F}} \boldsymbol{P}_{\mathcal{R}} \right\|_{F}^{2} + \mu_{k+1} \left\| \boldsymbol{\tau}_{\mathcal{F}} \right\|_{F}^{2} \Big). \tag{11}$$

Algorithm 3 CAV Basis Construction (CAVBASIS)

Input: Forget inputs $\boldsymbol{X}_{\mathcal{F}} \in \mathbb{R}^{N_{\mathcal{F}} \times d}$ with class labels $\boldsymbol{y}_{\mathcal{F}}$, retain inputs $\boldsymbol{X}_{\mathcal{R}} \in \mathbb{R}^{N_{\mathcal{R}} \times d}$, rank k, ridge penalty μ, mode $\in \{\text{kmeans}, \text{class}\}$
Output: Orthonormal basis $Q \in \mathbb{R}^{d \times k'}$ where $k' \leq k$ (kmeans) or $k' = |\mathcal{Y}_{\mathcal{F}}|$ (class)

Mean-center $\boldsymbol{X}_{\mathcal{F}}$ and $\boldsymbol{X}_{\mathcal{R}}$
if mode is KMEANS **then**
  $\{C_1, \ldots, C_k\} \leftarrow k$-means clustering on $\boldsymbol{X}_{\mathcal{F}}$
else
  $\{C_c\}_{c \in \mathcal{Y}_{\mathcal{F}}} \leftarrow$ partition $\boldsymbol{X}_{\mathcal{F}}$ by class label $\boldsymbol{y}_{\mathcal{F}}$ {one cluster per forget class}
end if
$Q \leftarrow []$
for each cluster C_j **do**
  $\boldsymbol{X}^+ \leftarrow \boldsymbol{X}_{\mathcal{F}}[C_j], \quad \boldsymbol{X}^- \leftarrow \boldsymbol{X}_{\mathcal{R}}$
  $\boldsymbol{Z} \leftarrow [\boldsymbol{X}^+; \boldsymbol{X}^-]^\top, \quad \boldsymbol{y} \leftarrow [+\mathbf{1}; -\mathbf{1}]$
  $\boldsymbol{w}_j \leftarrow (\boldsymbol{Z}\boldsymbol{Z}^\top + \mu \boldsymbol{I})^{-1} \boldsymbol{Z}\boldsymbol{y}$ {ridge regression}
  Orthogonalize $\boldsymbol{w}_j$ against columns of Q
  if $\|\boldsymbol{w}_j\| > \epsilon$ **then**
    Append $\boldsymbol{w}_j / \|\boldsymbol{w}_j\|$ to Q
  end if
end for
return Q

Proof. For any $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ and random vector x with second-moment $\boldsymbol{\Sigma} = \sum_i \sigma_i v_i v_i^\top$:

$$\mathbb{E}\big[\|\boldsymbol{A}x\|^2\big] = \operatorname{tr}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top) = \sum_{i=1}^{d} \sigma_i \|\boldsymbol{A}v_i\|^2 \tag{12}$$

$$= \underbrace{\sum_{i=1}^{k} \sigma_i \|\boldsymbol{A}v_i\|^2}_{\leq\, \sigma_1 \|\boldsymbol{A}\boldsymbol{Q}\|_F^2} + \underbrace{\sum_{i=k+1}^{d} \sigma_i \|\boldsymbol{A}v_i\|^2}_{\leq\, \sigma_{k+1} \|\boldsymbol{A}\|_F^2}, \tag{13}$$

where $\boldsymbol{Q} = [v_1, \ldots, v_k]$ and the tail bound uses $\sum_{i>k} \|\boldsymbol{A}v_i\|^2 \leq \|\boldsymbol{A}\|_F^2$. Since $\boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_k$, we have $\|\boldsymbol{A}\boldsymbol{Q}\|_F = \|\boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^\top\|_F = \|\boldsymbol{A}\boldsymbol{P}\|_F$. Setting $\boldsymbol{A} = \boldsymbol{\tau}_{\mathcal{R}} = \boldsymbol{\tau}_{\mathrm{m}} - \alpha \boldsymbol{\tau}_{\mathcal{F}}$ with $\boldsymbol{\Sigma}_{\mathcal{F}}$ gives equation 10; setting $\boldsymbol{A} = \alpha \boldsymbol{\tau}_{\mathcal{F}}$ with $\boldsymbol{\Sigma}_{\mathcal{R}}$ and factoring α^2 out of both norms gives equation 11. □

Remark B.1 (Bases that are not eigenbases)**.** *Proposition 3.3 takes $\boldsymbol{P}_{\mathcal{F}}$ and $\boldsymbol{P}_{\mathcal{R}}$ to be top-k eigenprojectors. The retain basis is always built by PCA, so equation 7 applies to every variant. The CAV forget bases are orthonormal but are not eigenvectors of $\boldsymbol{\Sigma}_{\mathcal{F}}$; for any orthonormal $\boldsymbol{Q}_{\mathcal{F}}$ with projector $\boldsymbol{P}_{\mathcal{F}}$, splitting $x = \boldsymbol{P}_{\mathcal{F}}x + (\boldsymbol{I} - \boldsymbol{P}_{\mathcal{F}})x$ and using $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ gives*

$$\mathbb{E}_{x \sim \mathcal{F}}\big[\|\boldsymbol{\tau}_{\mathcal{R}}\, x\|^2\big] \leq 2\nu_1 \left\|\boldsymbol{\tau}_{\mathcal{R}} \boldsymbol{P}_{\mathcal{F}}\right\|_F^2 + 2 \left\|(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{F}})\boldsymbol{\Sigma}_{\mathcal{F}}(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{F}})\right\|_{\mathrm{op}} \left\|\boldsymbol{\tau}_{\mathcal{R}}\right\|_F^2, \tag{14}$$

which reduces to equation 6 without the factor 2 when $\boldsymbol{P}_{\mathcal{F}}$ commutes with $\boldsymbol{\Sigma}_{\mathcal{F}}$. Term 1 therefore still controls the dominant component of forget leakage, and the tail constant ν_{k+1} is replaced by the largest forget variance left outside the span of $\boldsymbol{Q}_{\mathcal{F}}$, which the captured-variance diagnostic of §4.5 reports.

Proof of Corollary 3.4. Write the singular value decomposition $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top$ with $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_i)$; since $\boldsymbol{Q}_{\mathcal{F}}$ and $\boldsymbol{Q}_{\mathcal{R}}$ are orthonormal, $\sigma_i \in [0, 1]$ are the cosines of the principal angles between their spans. Then $\boldsymbol{C}\boldsymbol{C}^\top = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top$ and $\big((1+\gamma)\boldsymbol{I} + \lambda \boldsymbol{C}\boldsymbol{C}^\top\big)^{-1} = \boldsymbol{V} \operatorname{diag}(g_i)\, \boldsymbol{V}^\top$, which gives the form of $\boldsymbol{B}^\star$ from equation 4. For the retain term, $\|\boldsymbol{B}^\star \boldsymbol{C}\|_F^2 = \sum_i g_i^2 \sigma_i^2 \|\boldsymbol{a}_i\|^2$ with $\boldsymbol{a}_i = \boldsymbol{\tau}_{\mathrm{m}} \boldsymbol{Q}_{\mathcal{F}} \boldsymbol{v}_i$, against $\sum_i \sigma_i^2 \|\boldsymbol{a}_i\|^2$ for the plain projection $\boldsymbol{B}_0$. □

Scope of the assumption and of the bounds. *Assumption 3.1.* Each SGD update of a linear (or unfolded convolutional) layer is an outer product between a backpropagated signal and that layer's inputs, so the rows of the accumulated update lie in the span of the inputs seen during training. The

Table 4: Resources consumed at unlearning time (CIFAR-100 Trees, ResNet-50). $\boldsymbol{W}_{\text{init}}$: whether the pre-finetuning checkpoint is used ($\boldsymbol{W}_{\text{init}}$=0 is the convention in the $\boldsymbol{W}_0$ regime, App. D.4). Runtimes: $\boldsymbol{W}_{\text{PT}}$ cells of Tab. 10.

Method	Forget data	Retain data	Forget labels	Gradient updates	$\boldsymbol{W}_{\text{init}}$	Runtime
NegGrad+	yes	yes	yes	10 epochs (ascent on $\mathcal{F}$, descent on $\mathcal{R}$)	no	24s
Random Label	yes	yes	replaced by random	10 epochs	no	110s
SalUn	yes	yes	replaced by random	10 epochs + saliency pass	no	111s
NegTV	yes	no	yes	5 epochs on $\mathcal{F}$	no	4s
NegMerge	yes	no	yes	3× 5 epochs on $\mathcal{F}$	no	11s
Kodge et al.	1 forward pass (5k images)	1 forward pass (5k images)	no	none	no	8s
UNMERGE-PCA / CAV-KMEANS	1 forward pass	1 forward pass	no	none	$\boldsymbol{W}_{\text{PT}}$ regime only	8s / 26s
UNMERGE-CAV-CLASS	1 forward pass	1 forward pass	yes	none	$\boldsymbol{W}_{\text{PT}}$ regime only	22s

error term accounts for the drift of those inputs while earlier layers change, and its $\mathcal{O}(\eta^2)$ rate is a small-learning-rate idealization suited to finetuning from $\boldsymbol{W}_{\text{PT}}$. For from-scratch training with $\boldsymbol{W}_{\text{init}}$=0 we use the property heuristically: weight decay discounts the early updates, whose inputs differ most from the final activations, and we check the resulting low-rank structure empirically (§4.3). *What equation 6 measures.* Its left-hand side is the response of $\boldsymbol{\tau}_{\mathcal{R}}$ on forget inputs, a proxy for forgetting rather than a distance to $\boldsymbol{W}_{\text{RT}}$: a retrained model still responds to the features that forget inputs share with retain inputs, which is the part Term 2 protects. *Gain.* The first term of equation 6 is smallest at unit gain, $\alpha/(1+\gamma+\lambda\sigma_i^2)=1$, whereas the selected α exceeds $1+\gamma$ (e.g. $\alpha/(1+\gamma)\approx 1.7$ for the CAV bases on $\boldsymbol{W}_{\text{PT}}$): the edit over-corrects the k captured directions. We attribute this to forget signal left outside the rank-k span (the tail term of equation 6), which a unit-gain edit cannot out-vote at the classifier; the bounds motivate the three terms and are not minimized by the tuned edit. *Layers.* The bounds are per layer, with bases computed on the un-edited network, and do not track how upstream edits shift downstream activations; `skip-layers` confines this interaction to the last 8–18 layers, and we did not test sequential re-estimation of the bases.

B.3 REQUIREMENTS, RUNTIME AND COST

Requirements. Tab. 4 makes precise what each method consumes at unlearning time. No method in our comparison is data-free in the sense of touching no data; UNMERGE is *optimization-time* data-free: $\mathcal{F}$ and $\mathcal{R}$ are read in forward passes only (one for PCA, two over $\mathcal{R}$ for CAV), to form second-moment statistics, and no gradient is ever taken with respect to a training sample. The retain statistics can be cached and reused across forget requests on the same checkpoint. Every method uses the retrained model for evaluation only.

Runtime and cost model. Tab. 5 times the three phases of Alg. 1 on the official Trees cells. The pipeline is data-bound: more than 95% of the wall-clock is the single forward pass over $\mathcal{F}\cup\mathcal{R}$ plus basis construction, the optimization of $\{\boldsymbol{B}^l\}$ takes under one second. Per layer, the costs are $\mathcal{O}(Nd_l)$ to collect activations, $\mathcal{O}(Nd_l^2)$ to accumulate the second-moment matrix, $\mathcal{O}(d_l^3)$ for its eigendecomposition (only over non-skipped layers; a randomized top-k solver reduces this to $\mathcal{O}(d_l^2 k)$), and $\mathcal{O}(Tmk^2)$ for the T Adam steps ($\mathcal{O}(mk^2+k^3)$ for the exact solve of equation 4), so neither solver affects how the method scales. A gradient baseline that trains for E epochs costs E forward and backward passes over the same data, so the runtime ratio is governed by E rather than by model size as long as the per-layer eigendecomposition stays cheaper than one forward pass over N samples—true by orders of magnitude for d_l up to $\sim 10^4$. The measured ratios against Random Label and SalUn (4–23× on ResNet-50 and ViT-S/16 at E=10) vary with how much of UNMERGE's time goes to basis construction, which is larger for the CAV variants. One detail of the timing should be kept in mind: the baselines' training loops evaluate on the test set after every epoch to log progress, and their timers include it (0.72s per epoch on CIFAR-100, 7.2s over E=10 epochs), whereas UNMERGE's timer stops before evaluation. This is under 7% of Random Label's and SalUn's runtime but about a third of NegGrad+'s, whose epochs are short because they only traverse $\mathcal{F}$ with as many retain samples (20 steps). Net of that logging pass NegGrad+ takes 16.7s instead of 23.9s on Trees ($\boldsymbol{W}_{\text{PT}}$), so UNMERGE-PCA (8.6s) remains about 2× faster than NegGrad+ and the CAV variants (18–26s) are on par with it; the ∼5× advantage over Random Label and SalUn is unaffected. Beyond the preliminary 3B study of App. E.4, language-model scale is scoped as future work.

C EXPERIMENTAL DETAILS

Forget tasks. CIFAR-100 superclasses (five classes, $|\mathcal{F}|$=2500): Trees `[47,52,56,59,96]`, Aquatic Mammals `[4,30,55,72,95]`, Vehicles-1 `[8,13,48,58,90]`. Tiny-

Table 5: Phase-wise runtime of UNMERGE on CIFAR-100 (Trees), ResNet-50, single H100.

Cell	Phase 1 (activations + bases + τ_m)	Phase 2 (100 Adam steps)	Phase 3 (apply)	Total
$\boldsymbol{W}_0$, CAV-Class	17.1s	0.5s	0.0s	17.6s
$\boldsymbol{W}_{\mathrm{PT}}$, PCA ($k$=64)	8.2s	0.35s	0.0s	8.6s

Table 6: UNMERGE configurations on CIFAR-100 (Trees), ResNet-50.

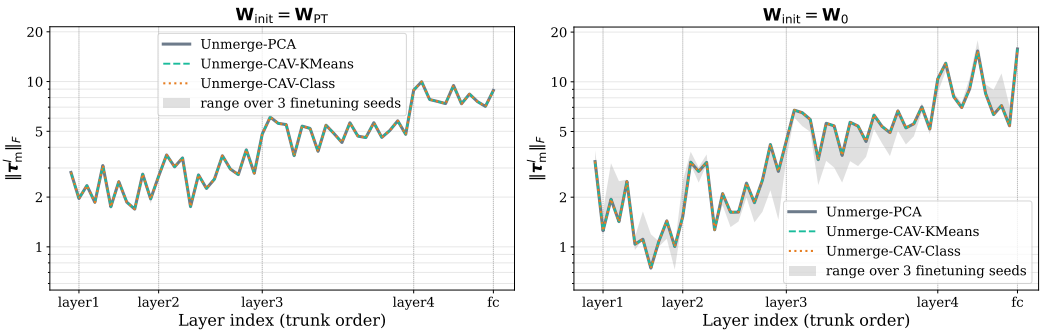
Cell	rank k	`skip-layers`	α	λ	γ
$\boldsymbol{W}_{\mathrm{PT}}$, PCA	64	45	1.6	5	0.5
$\boldsymbol{W}_{\mathrm{PT}}$, CAV-KMeans	16	35	2.5	5	0.5
$\boldsymbol{W}_{\mathrm{PT}}$, CAV-Class	16	35	2.5	5	0.5
$\boldsymbol{W}_0$, PCA	128	45	1.25	50	0.5
$\boldsymbol{W}_0$, CAV-KMeans	16	40	1.0	50	0.5
$\boldsymbol{W}_0$, CAV-Class	16	40	1.3	50	0.5

ImageNet semantic groups (ten classes): Dogs & Cats `[24-33]` and Arthropods `[7,9,10,36,38,39,41,42,43,45]`. Within each task $\mathcal{F}$ is the union of the chosen classes' training images and $\mathcal{R} = \mathcal{S} \setminus \mathcal{F}$; the instance-level protocols are described in App. E.7.

Training. All models are trained with SGD (momentum 0.9, cosine schedule, batch size 128). ResNet-50 on CIFAR-100: 100 epochs at learning rate 0.1 from $\boldsymbol{W}_0$ and 50 epochs at 0.01 from $\boldsymbol{W}_{\mathrm{PT}}$ (torchvision `IMAGENET1K_V1`; hash in App. D.4), weight decay 5×10^{-4}; TinyImageNet uses the same schedules with weight decay 10^{-4}; ViT-S/16 is finetuned for 10 epochs at 0.01. Each retrained reference $\boldsymbol{W}_{\mathrm{RT}}$ uses the schedule of its $\boldsymbol{W}_{\mathrm{FT}}$ on $\mathcal{R}$ only. All experiments run on a single NVIDIA H100.

Baselines. All gradient baselines run for E=10 epochs. CIFAR-100: NegGrad+ (α=0.5, learning rate 10^{-4}), Random Label (learning rate 0.01), SalUn (saliency threshold 0.5, learning rate 0.01–0.02). TinyImageNet: NegGrad+ ($\alpha \in \{0.5, 0.8\}$, learning rate 10^{-4}), Random Label and SalUn at learning rate 10^{-3}. The task-vector baselines subtract a vector obtained by gradient training on $\mathcal{F}$: NegTV (Ilharco et al., 2023) finetunes $\boldsymbol{W}_{\mathrm{FT}}$ on $\mathcal{F}$ for 5 epochs to $\boldsymbol{W}_{\mathcal{F}}$ and applies $\boldsymbol{W}_{\mathrm{UL}} = \boldsymbol{W}_{\mathrm{FT}} - \alpha(\boldsymbol{W}_{\mathcal{F}} - \boldsymbol{W}_{\mathrm{FT}})$; NegMerge (Kim et al., 2025) obtains K=3 such vectors at $\{0.5, 1, 2\}\times$ the base learning rate, keeps the entries whose sign agrees across all K, averages them, and subtracts. Both were given an equal-budget grid on Trees (learning rate $\{10^{-3}, 10^{-2}\}$, $\alpha \in [0.25, 4]$, 28 runs plus an α refinement), and the Trees configuration (learning rate 10^{-3}, 5 epochs) is transferred to the other tasks with only α re-swept over $\{0.25, 0.35, 0.5, 0.75, 1.0\}$; every reported optimum is interior. Their failure mode is retain collapse: the finetuned vector overlaps retain directions, so the best cells already lose 1–13 points of retain accuracy or leave up to 26% forget accuracy and every stronger α pushes retain toward chance, and the two methods are within 0.04 ToW of each other in every cell. The activation-subspace baseline of Kodge et al. (2024) is run as published, on every convolutional and linear layer, from the same forward-pass statistics as UNMERGE (5000 images per set): its retain and forget scalings are tuned on Trees and Dogs & Cats by ToW over $\alpha_r \in \{100, 300, 1000, 3000\}$ and $\alpha_f \in \{3, 10, 30, 100\}$, and α_f is re-swept on the transfer tasks; the grids were extended at the edges until every reported optimum is interior.

UNMERGE configurations. Phase 2 always uses T=100 Adam steps at learning rate 10^{-3}, never tuned. Tab. 6 lists the Trees configurations. They are tuned once per (variant, init) on Trees (CIFAR-100) and Dogs & Cats (TinyImageNet) and transferred to the other tasks changing at most α (by ≤ 0.5; PCA on $\boldsymbol{W}_{\mathrm{PT}}$ transfers with α unchanged) and, in three $\boldsymbol{W}_0$ cells, `skip-layers` by one step of five layers. The three metrics are defined in §4.1; attack details are in App. E.6. The entanglement distances of §4.4 are computed on penultimate-layer features (the classifier input) of $\mathcal{F}$ and a same-size retain subsample ($|\mathcal{R}|$=$|\mathcal{F}|$): the unbiased Gaussian-RBF MMD^2 at the median bandwidth σ_{med} and averaged over $\sigma \in \{0.5, 1, 2\}\cdot\sigma_{\mathrm{med}}$ (multi), the closed-form Bures–Wasserstein W_2^2 between Gaussian fits (mean-shift dominated when the feature dimension exceeds the sample size), and the Sliced-W_2^2 over 1000 random one-dimensional projections.



(a) $\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$ (b) $\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$

Figure 2: Per-layer merged-task-vector magnitude $\|\boldsymbol{\tau}_{\text{m}}^l\|_F$ on CIFAR-100 (Trees), ResNet-50, log-y. The three variant curves coincide exactly because $\boldsymbol{\tau}_{\text{m}}$ is basis-independent; the shaded band is the range over three finetuning seeds. The cross-init gap is depth-dependent: shallow blocks (layer1–2) carry slightly larger magnitudes on $\boldsymbol{W}_{\text{PT}}$ (pretrained features must be shifted away from ImageNet), while at layer4 and the classifier $\|\boldsymbol{\tau}_{\text{m}}\|$ is $\sim$1.5–2$\times$ larger on $\boldsymbol{W}_0$, where the deepest task-specific weights are encoded from scratch.

D Ablations of Unmerge

D.1 Per-Layer Diagnostics and Hyperparameters

This subsection covers the design choices that determine Unmerge performance—basis construction (PCA vs. CAV), rank k, the scaling α, `skip-layers`, and the regularizers λ, γ—and how α is selected without the retrained model; the objective terms and the solver are ablated in App. D.2 and D.3. The per-layer figures of §4.3 (Figs. 1, 2) drive the recipe; the paragraphs below summarize what each knob controls and the ranges that worked across all reported runs.

Recipe in brief. The two diagnostics read off the hyperparameters used throughout: `skip-layers` at the overlap elbow (layer3→layer4; CAV 35/40, PCA 45–50), α inversely to the deepest-layer $\|\boldsymbol{\tau}_{\text{m}}\|$ ($[1.3, 2.5]$ on $\boldsymbol{W}_{\text{PT}}$, $[0.5, 1.3]$ on $\boldsymbol{W}_0$), and CAV where entanglement is the binding constraint ($\boldsymbol{W}_0$). They also explain why Unmerge-PCA wins ToW on $\boldsymbol{W}_{\text{PT}}$ yet loses on $\boldsymbol{W}_0$ at the same entanglement (next paragraph).

PCA vs. CAV. PCA gives a strong, unsupervised baseline but its top-k eigenvectors absorb dataset-generic variance that overlaps with retain—visible as the highest per-direction overlap at every depth in Fig. 1 (0.54–0.58 at layer4 and 0.46–0.56 at the classifier, all bases at rank 16). CAV's discriminative ridge probes target only forget-distinguishing directions (0.40–0.46 at layer4, 0.08–0.20 at the classifier); the two CAV modes are essentially tied per direction, so the gap between them in the main tables comes from the number of directions ($|\mathcal{Y}_{\mathcal{F}}|$ vs. k), not from entanglement. The empirical consequence is the regime-split observed in Tab. 1, which the two diagnostics of §4.3 explain jointly: by equation 7, retain damage scales as $\alpha^2\mu_1\|\boldsymbol{Q}_{\mathcal{F}}^\top\boldsymbol{Q}_{\mathcal{R}}\|_F^2$ times the basis mass $\|\boldsymbol{B}\|^2$, so entanglement and $\|\boldsymbol{\tau}_{\text{m}}\|$ enter the retain budget together, and since the entanglement of each basis is nearly the same in both regimes it is the deep $\|\boldsymbol{\tau}_{\text{m}}\|$ that decides. PCA wins ToW on $\boldsymbol{W}_{\text{PT}}$ (0.9851), where moderate deep $\|\boldsymbol{\tau}_{\text{m}}\|$ tolerates the higher overlap at larger α; on $\boldsymbol{W}_0$, deep $\|\boldsymbol{\tau}_{\text{m}}\|$ peaks at $\sim$15 and forces a smaller α to protect retain, which under-drives PCA (0.9230), while CAV-Class's less entangled basis (0.45 at layer4, 0.17 at the classifier) lets α climb back and forget cleanly (0.9909). All three bases are nearly init-invariant per direction ($\leq$0.13 change in any block between $\boldsymbol{W}_{\text{PT}}$ and $\boldsymbol{W}_0$), and CAV-Class remains the safer default when the init is unknown.

Rank k. Per Lemma 3.2, the residual $\|\boldsymbol{E}\|_F^2$ is bounded by $N_{\mathcal{F}}\sum_{i>k}\nu_i$, so accuracy saturates once k covers the effective dimensionality of the per-layer forget covariance. We sweep $k \in \{8, 16, 32, 64, 128\}$; for CAV bases the elbow lies near $k=16$ on ResNet-50 and $k=16$–32 on ViT-S, beyond which retain accuracy degrades by 0.5–2% at fixed α as the rank-k form picks up more retain-aligned directions. PCA prefers larger ranks ($k=64$–128): its variance-ordered eigenvectors need more directions to cover the forget signal, and on $\boldsymbol{W}_0$—where deep task vectors are large—$k=128$ captures more forget energy per unit α and clearly pays off. CAV-Class is invariant to k by

construction (one direction per forget class, $k' = |\mathcal{Y}_\mathcal{F}|$), which removes this knob in the supervised regime.

Scale α. $\alpha \in [0.5, 2.5]$ tracks the deepest-layer $\|\boldsymbol{\tau}_\mathrm{m}\|$ (Fig. 2) *inversely*: $\boldsymbol{W}_\mathrm{PT}$ keeps $\alpha \in [1.3, 2.5]$ where deep $\|\boldsymbol{\tau}_\mathrm{m}\| \leq 10$, $\boldsymbol{W}_0$ uses $\alpha \in [0.5, 1.3]$ to cap retain damage when deep $\|\boldsymbol{\tau}_\mathrm{m}\| \approx 15$. The product $\alpha\|\boldsymbol{\tau}_\mathrm{m}\|$ is the operative quantity in equation 7.

Skip-layers. Restricting unmerging to deeper layers (`skip-layers` = 35–50 of the 53 convolutions of ResNet-50; the classifier is always edited: 35 edits the last 18 convolutions, from `layer3.3.conv2` on, 40 the last 13, from `layer3.5.conv1` on, 45 the last 8, all in layer4, and 50 the last 3) preserves the forget drop while *improving* retain/test by 2–5 points relative to all-layers updates. The cutoff is read directly off the overlap elbow in Fig. 1 (layer3→layer4 transition) and is pushed deeper the more entangled the basis: CAV uses 35/40, while PCA—whose variance-driven eigenvectors overlap retain most—works best at 45–50. Shallower blocks have larger retain-damage constant μ_1 and more entangled bases, so their unmerging budget costs more retain than it buys in forget.

Regularizers λ, γ. $\lambda \in \{5, 20, 50\}$ scales the retain-leakage penalty (Term 2 of equation 2), chosen per initialization regime: $\lambda = 5$–20 on $\boldsymbol{W}_\mathrm{PT}$ and $\lambda = 50$ on $\boldsymbol{W}_0$, where larger deep task vectors demand stronger retain protection. The penalty is most binding on PCA bases, whose overlap with retain is highest; CAV-Class runs are insensitive to λ across an order of magnitude because the basis itself already separates forget from retain. $\gamma \in \{0.25, 0.5\}$ acts as a uniform magnitude prior on $\boldsymbol{B}^l$ (shrinking the closed-form solution of Term 1 by $1/(1+\gamma)$) and is most helpful on $\boldsymbol{W}_0$ where $\boldsymbol{\tau}^l_\mathcal{F}$ would otherwise over-expand along tail directions of the forget covariance. Both regularizers transfer across superclasses without retuning (§4.1), which we view as part of the unlearning quality of UNMERGE.

Selecting α without the retrained model. All configurations in this paper, ours and the baselines', are selected by ToW, which uses $\boldsymbol{W}_\mathrm{RT}$ and the test set, as is common practice (Fan et al., 2024; Kurmanji et al., 2023). For a class-level request a practitioner has neither, but knows the target forget accuracy (zero). Because $\boldsymbol{B}^l$ does not depend on α, the update of equation 3 can be line-searched after a single solve, at the cost of one forward pass over $\mathcal{F}$ per candidate. We swept α on a uniform grid for the six Trees configurations. Forget and retain accuracy are monotone in α, so we test the rule *take the smallest α whose forget-set accuracy is at most* 1%. It selects the ToW-optimal grid point in five of the six cells and loses 0.003 ToW in the sixth; the training-only score $\mathrm{Acc}_\mathcal{R}(1 - \mathrm{Acc}_\mathcal{F})$ of Kodge et al. (2024) loses at most 0.006. On the coarser grids of the transfer superclasses the rule selects the ToW-optimal cell or loses at most 0.012 (nine cells). On TinyImageNet/$\boldsymbol{W}_0$ it applies only to CAV-CLASS (loss ≤ 0.007): for CAV-KMEANS no α on our grid reaches 1% forget accuracy, and for PCA on Arthropods the rule over-shoots (-0.15 ToW), so the hardest regime still needs a reference. Rank, `skip-layers`, λ and γ are transferred from the tuning task unchanged (App. C).

D.2 OBJECTIVE TERMS

Tab. 7 removes the terms of equation 2 one at a time with the reported estimator and re-sweeps α for every variant. With Term 1 alone the objective is minimized at the initialization, so the edit is α times the projection $\boldsymbol{\tau}_\mathrm{m}\boldsymbol{Q}_\mathcal{F}$: it collapses retain for the PCA basis (ToW 0.46–0.48, retain 71–94%) and trails the full objective by 0.04–0.35 for the CAV bases, whose directions overlap retain less. Term 2 accounts for essentially all of the gain: with Terms 1 and 2 every cell is within 0.02 of the full objective, and above it in one. Term 3 is a uniform shrinkage that α already provides: on its own it closes at most part of the gap (Terms 1+3: 0.50–0.96, below Terms 1+2 in every cell), and at the reported λ its effect is within 0.02 everywhere. The optimization of Phase 2 is therefore meaningful precisely as the solver of the Term-2-regularized problem—the iterate ends at relative distance 0.7–1.9 from its initialization (App. D.3)—and not because the initialization is already a good edit. The task-vector baselines corroborate this from outside the method: NegTV and NegMerge subtract a vector finetuned on $\mathcal{F}$ with no retain penalty, and they lose retain accuracy in the same way as Term 1 alone (App. C).

D.3 THE BUDGETED SOLVER AND THE EXACT MINIMIZER

Phase 2 of Alg. 1 moves $\boldsymbol{B}^l$ from the projection $\boldsymbol{B}^l_0 = \boldsymbol{\tau}^l_\mathrm{m}\boldsymbol{Q}^l_\mathcal{F}$ toward the exact minimizer $\boldsymbol{B}^{l\star}$ of equation 4 under a fixed budget (T=100 Adam steps, learning rate 10^{-3}). This subsection

Table 7: Term ablation on CIFAR-100 (Trees) with the reported estimator (100 Adam steps): best ToW over an α sweep for each variant, with the selected α in parentheses. Term 1 only: λ=γ=0 (the edit is $\alpha\,\boldsymbol{\tau}_{\mathrm{m}}\boldsymbol{Q}_{\mathcal{F}}$). Terms 1+3: λ=0, γ=0.5. Terms 1+2: λ as reported, γ=0. All three: λ and γ as reported. Every optimum is interior to its grid.

Cell	Term 1 only	Terms 1+3	Terms 1+2	All three
$\boldsymbol{W}_0$, PCA	0.4622 (0.5)	0.4978 (0.75)	0.9608 (1.25)	**0.9609 (1.25)**
$\boldsymbol{W}_0$, CAV-KMeans	0.9419 (0.3)	0.9433 (0.4)	0.9819 (1)	**0.9834 (1)**
$\boldsymbol{W}_0$, CAV-Class	0.9530 (0.5)	0.9553 (0.6)	0.9863 (1.5)	**0.9895 (1.4)**
$\boldsymbol{W}_{\mathrm{PT}}$, PCA	0.4802 (1)	0.8174 (1.5)	0.9843 (1.6)	**0.9868 (1.6)**
$\boldsymbol{W}_{\mathrm{PT}}$, CAV-KMeans	0.6162 (1.5)	0.6897 (1.875)	**0.9775 (1.6)**	0.9699 (2.4)
$\boldsymbol{W}_{\mathrm{PT}}$, CAV-Class	0.8089 (1.5)	0.8600 (2.25)	0.9294 (2)	**0.9459 (2.4)**

Table 8: The early-stopped Adam iterate vs. the closed-form model $\boldsymbol{B}_\delta$ of equation 15 on the six official CIFAR-100 (Trees) cells, at the reported $(\lambda, \gamma, \alpha)$, seed 1. Rel. dist.: distance of the Adam iterate to the projection $\boldsymbol{B}_0$, to the minimizer $\boldsymbol{B}^\star$, and to $\boldsymbol{B}_\delta$ at the best $\delta \in \{0.02, \ldots, 0.12\}$, relative to $\|\boldsymbol{B}_{\mathrm{Adam}}\|_F$ and aggregated over edited layers. Resid.: share of entries with $|\boldsymbol{B}_0 - \boldsymbol{B}^\star| > 0.1$. ToW: the reported cell, and $\boldsymbol{B}_\delta$ applied in its place.

	Rel. dist. of the iterate to			Resid.	ToW	ToW of $\boldsymbol{B}_\delta$		
Cell	$\boldsymbol{B}_0$	$\boldsymbol{B}^\star$	$\boldsymbol{B}_\delta$	(%)	Adam	δ=0.1	0.075	0.05
$\boldsymbol{W}_0$, CAV-Class	1.46	0.35	0.09	0.07	0.9851	0.9804	0.9864	0.9860
$\boldsymbol{W}_0$, CAV-KMeans	1.64	0.49	0.32	0.15	0.9834	0.9604	0.9784	0.9795
$\boldsymbol{W}_0$, PCA (k=128)	1.68	0.96	0.31	0.36	0.9609	0.9610	0.9088	0.5926
$\boldsymbol{W}_{\mathrm{PT}}$, CAV-Class	0.74	0.07	0.01	0.01	0.9418	0.9380	0.9418	0.9429
$\boldsymbol{W}_{\mathrm{PT}}$, CAV-KMeans	1.31	0.05	0.01	0.00	0.9652	0.9651	0.9654	0.9648
$\boldsymbol{W}_{\mathrm{PT}}$, PCA ($k$=64)	1.85	0.45	0.16	0.10	0.9868	0.9867	0.9779	0.9452

characterizes the estimator that results, compares it with the exact minimizer, and states what depends on the budget, on the six official Trees cells. (Implementation note: the code averages each loss term over its entries rather than summing, so its exact minimizer uses $\lambda_{\text{eff}} = \lambda\, k_F/k_R$; this differs from λ only for CAV-Class, where $k_F = |\mathcal{Y}_{\mathcal{F}}| = 5$ and $k_R = 16$.)

What the budgeted iterate computes. Adam's normalized updates move every entry of $\boldsymbol{B}^l$ by at most the step budget $\delta = \mathrm{lr} \times T = 0.1$, several times the typical entry of the projection (rms 0.012–0.025). Entries closer than δ to the minimizer therefore converge, and the others move by about δ, so the iterate should be close to

$$\boldsymbol{B}^l_\delta = \boldsymbol{B}^{l\star} + S_\delta\big(\boldsymbol{B}^l_0 - \boldsymbol{B}^{l\star}\big), \tag{15}$$

with S_δ the entrywise soft-threshold: the exact minimizer plus a sparse residual of the largest projection coefficients, interpolating between the projection (δ=0) and the minimizer ($\delta\to\infty$). Tab. 8 tests this model, which has no free parameter. Entrywise, $\boldsymbol{B}_\delta$ is the closest of the three candidates to the Adam iterate in every cell (relative distance 0.01–0.32, against 0.05–0.96 for $\boldsymbol{B}^\star$ and 0.74–1.85 for $\boldsymbol{B}_0$), and applying $\boldsymbol{B}_{0.1}$ in place of the iterate, at the reported hyperparameters, reproduces the reported ToW within 0.005 in five of the six cells (0.023 in the sixth). The model also explains the split between bases. For the CAV bases the residual is nearly empty (at most 0.15% of the entries survive the threshold) and ToW moves by at most 0.02 over $\delta \in [0.05, 0.1]$: the solver hardly matters, which is why the exact minimizer can replace the iterate there. For PCA the residual carries the edit. On $\boldsymbol{W}_0$ the minimizer is a no-op at the reported λ and 0.36% of the coefficients do the forgetting; this is a selection over (output unit, direction) pairs—the iterate places 37% of its energy in the top 5% of output units, against 22% for $\boldsymbol{B}^\star$—that no ridge solution, which filters all output units identically, can express. The price is a dependence on the step budget that the CAV variants do not have (last columns of Tab. 8). All our experiments use one budget (T=100, learning rate 10^{-3}), never tuned per cell.

The exact minimizer, tuned for itself. Hyperparameters do not transfer between the two estimators. At the $(\lambda, \gamma, \alpha)$ tuned for the budgeted iterate, $\boldsymbol{B}^{l\star}$ reproduces the CAV cells on $\boldsymbol{W}_{\mathrm{PT}}$ (ToW 0.932/0.965 vs. 0.942/0.965; the iterate is within relative distance 0.05–0.07 of it there), under-

Table 9: $\boldsymbol{W}_{\text{init}}$ ablation on CIFAR-100 (Trees), $\boldsymbol{W}_{\text{PT}}$ regime: ToW per UNMERGE variant (retrained reference F/R/T 0.00/99.98/80.67).

$\boldsymbol{W}_{\text{init}}$	CAV-Class	CAV-KMeans	PCA
True init (`V1`, paper cells)	0.9418	**0.9652**	**0.9868**
Zero, paper α	0.7249	0.6036	0.8291
Zero, α retuned	**0.9629**	0.9639	0.9422
Wrong checkpoint (`V2`)	0.4742	0.0745	0.0628

forgets the CAV cells on $\boldsymbol{W}_0$ (0.89/0.68) and is a no-op for both PCA cells, because at that λ the retain penalty shrinks $\boldsymbol{\tau}_{\mathcal{F}}$ to 3–10% of $\boldsymbol{\tau}_{\text{m}}$ along PCA's heavily overlapping directions; the plain projection $\boldsymbol{B}_0^l$ at the same α collapses retain instead (1.1%/14.6%). Tuned for itself over $\lambda \in \{0, \ldots, 200\}$, $\gamma \in \{0, 0.5\}$ and α (34–50 runs per cell, all optima interior), the exact minimizer prefers γ=0 and a larger λ (10–100) with a correspondingly larger α—it scales the component of $\boldsymbol{\tau}_{\text{m}}^l \boldsymbol{Q}_{\mathcal{F}}^l$ along the i-th principal direction between the forget and retain spans by $1/(1+\gamma+\lambda\sigma_i^2)$, so a larger λ shrinks only the shared directions—and is then within 0.01 of the reported iterate or above it in five of six cells (0.990/0.975 for CAV-Class/KMeans on $\boldsymbol{W}_0$; 0.999/0.996/0.991 for CAV-Class/KMeans/PCA on $\boldsymbol{W}_{\text{PT}}$), while it fails for PCA on $\boldsymbol{W}_0$ (0.65). The reported estimator run at those hyperparameters reproduces the closed form for the CAV bases (0.999, 0.995, 0.991 and 0.979), so the gap to the reported CAV cells reflects the λ range of the original Trees grid (up to 5 on $\boldsymbol{W}_{\text{PT}}$) rather than the solver, and those cells are conservative; for PCA the two estimators remain different objects (0.40 and 0.0001 at the closed form's optimum).

Scope. Both solvers are negligible in cost (0.2–0.8s for the iterate and 0.01–0.03s for the solve, against 8–21s for Phase 1), so the choice is about universality, not efficiency. We report the budgeted iterate throughout because, at one fixed budget, it is the only estimator that works in all six cells. For the CAV variants the exact minimizer is an equivalent, budget-free alternative, and equation 15 gives the iterate an explicit form in which the threshold δ replaces the step count; the PCA variants, in contrast, depend on the budget (last columns of Tab. 8), which is a limitation of those variants. Adopting an explicit form as the default changes the scale of (λ, α) and needs a multi-seed, multi-dataset validation that we leave to future work. All cells of this subsection are single-seed ablations on Trees.

D.4 $\boldsymbol{W}_{\text{init}}$ ABLATION

UNMERGE forms $\boldsymbol{\tau}_{\text{m}} = \boldsymbol{W}_{\text{FT}} - \boldsymbol{W}_{\text{init}}$, so we ask how much the pre-finetuning checkpoint matters (Tab. 9; Trees, $\boldsymbol{W}_{\text{PT}}$ regime). Setting $\boldsymbol{W}_{\text{init}}$=0—our convention throughout the $\boldsymbol{W}_0$ regime—makes $\boldsymbol{\tau}_{\text{m}}$ larger (it now contains the pretrained weights), so the paper's α over-shoots; with α retuned on the same grid (0.9 for CAV and 1.2 for PCA, against 2.5 and 1.6 at the true init) zero-init matches the CAV cells and trails only on PCA, whose variance basis otherwise absorbs the shared pretrained content that the true init subtracts. Substituting a *different* pretrained checkpoint (torchvision `IMAGENET1K_V2` in place of `V1`) is catastrophic for every variant: the subtracted vector then contains the difference between two unrelated pretrainings. The exact checkpoint is therefore not required—but a wrong one must not be used, and $\boldsymbol{W}_{\text{init}}$=0 is the safe fallback. The $\boldsymbol{W}_{\text{PT}}$ checkpoint is part of the method's input and we release it: torchvision ResNet-50 `IMAGENET1K_V1` with CIFAR-adapted first convolution and classifier (SHA-256 prefix `2a54223ea63a790f`; full hash in the code release). The adapted layers are seeded, so the file, not the recipe, is the artifact.

E ADDITIONAL RESULTS

E.1 DETAILED PER-TASK RESULTS

The summary in Tab. 1 averages over three CIFAR-100 superclasses (Trees, Aquatic Mammals, Vehicles); we expose the per-superclass breakdown here. Tabs. 10, 11, and 12 report the three CIFAR-100 superclasses individually, and Tab. 13 reports the second TinyImageNet superclass (Arthropods). The hyperparameter set is *tuned once per (method, init)* on Trees and Dogs & Cats and *transferred* to Aquatic Mammals, Vehicles, and Arthropods with at most a small change of α (App. C), so these tables also stress-test transferability. The pattern from the main paper is preserved: in every (superclass, init) cell at least one UNMERGE variant leads every baseline on ToW, including the

Table 10: CIFAR-100 unlearning results (Trees). Rows: evaluation metric (Forget/Retain/Test accuracy, %) and tug-of-war score (equation 8; higher is better). Columns: unlearning method. $\boldsymbol{W}_{\mathrm{FT}}$ and $\boldsymbol{W}_{\mathrm{RT}}$ are the finetuned upper bound and retrained gold standard. Best non-retrained in **bold**.

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{\mathrm{PT}}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	0.36	12.24	4.32	1.72	2.60	0.16	0.08	0.00	0.12
Retain Acc ↑	99.98	97.21	99.94	99.92	96.87	97.50	99.67	99.80	99.40	98.82
Test Acc ↑	80.67	74.48	80.62	80.64	74.59	75.24	78.25	79.60	77.75	76.07
ToW ↑	1	0.9089	0.8768	0.9560	0.8943	0.8983	**0.9712**	**0.9868**	0.9652	0.9418
Runtime ↓		24s	110s	111s	**4s**	11s	8s	8s	26s	22s
$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	4.40	15.44	3.08	1.40	1.96	0.08	0.12	0.68	1.12
Retain Acc ↑	99.97	96.04	99.96	99.94	95.60	96.50	99.01	98.75	99.83	99.87
Test Acc ↑	72.92	66.05	74.00	74.14	66.19	66.91	70.05	70.31	72.07	72.65
ToW ↑	1	0.8553	0.8364	0.9571	0.8795	0.8895	0.9613	0.9609	**0.9834**	**0.9851**
Runtime ↓		24s	111s	115s	**4s**	12s	9s	9s	21s	18s

Table 11: CIFAR-100 unlearning results (Aquatic Mammals). Best non-retrained in **bold**.

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{\mathrm{PT}}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	0.56	3.36	5.52	4.68	7.24	0.44	0.20	0.20	1.40
Retain Acc ↑	99.99	95.34	99.94	99.98	98.00	98.68	99.67	99.72	99.71	99.52
Test Acc ↑	80.74	73.16	80.26	81.05	75.64	76.46	78.54	79.20	79.16	77.95
ToW ↑	1	0.8763	0.9613	0.9417	0.8866	0.8763	0.9705	**0.9799**	**0.9795**	0.9540
Runtime ↓		23s	110s	112s	**4s**	11s	8s	9s	27s	22s
$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	5.72	3.40	4.32	5.72	6.32	0.72	0.72	0.56	0.16
Retain Acc ↑	99.97	97.18	99.97	99.97	96.93	97.18	99.00	95.80	99.66	99.81
Test Acc ↑	72.8	69.02	74.55	74.69	68.82	68.85	70.68	66.66	72.57	73.48
ToW ↑	1	0.8819	0.9491	0.9387	0.8778	0.8747	0.9623	0.8929	**0.9891**	**0.9900**
Runtime ↓		23s	109s	112s	**4s**	11s	8s	6s	21s	18s

task-vector baselines and the projection of Kodge et al. (2024); on the CIFAR-100 superclasses at least two of three lead every gradient-based baseline, whereas on TinyImageNet/$\boldsymbol{W}_0$ (Dogs & Cats and Arthropods) only UNMERGE-CAV-CLASS leads and PCA/KMeans trail SalUn. UNMERGE-CAV-CLASS is consistently strongest on $\boldsymbol{W}_0$; on $\boldsymbol{W}_{\mathrm{PT}}$ the lead goes to UNMERGE-PCA on the CIFAR-100 superclasses, to UNMERGE-CAV-KMEANS on Arthropods and to UNMERGE-CAV-CLASS on Dogs & Cats.

E.2 SEED VARIANCE

Tab. 14 repeats the Trees experiments for three seeds with frozen configurations. The seed controls model initialization only: the data order and augmentation streams are fixed across seeds. In the $\boldsymbol{W}_0$ regime it therefore changes the initialization of the finetuned and the retrained model; in the $\boldsymbol{W}_{\mathrm{PT}}$ regime retraining from the fixed checkpoint is deterministic given the data order, so all seeds share one retrained reference and the variance reflects finetuning and method stochasticity ($\boldsymbol{W}_{\mathrm{init}}$ is held fixed across seeds, including its seeded stem and classifier entries). Every ordering between the UNMERGE variants and SalUn/Random Label in Tab. 10 is preserved in the means; NegGrad+ and Random Label swap on $\boldsymbol{W}_0$ within noise. On $\boldsymbol{W}_0$ the UNMERGE standard deviations (0.008–0.021) are 1.7–4.5× smaller than those of Random Label and SalUn (0.036–0.037), and UNMERGE-PCA's mean (0.937) sits above SalUn's (0.926) although the single-seed cells are close. On $\boldsymbol{W}_{\mathrm{PT}}$ SalUn is the most stable method (0.003) but trails UNMERGE-PCA and CAV-KMEANS by 1.0–1.3 points. The projection of Kodge et al. (2024) is on par with UNMERGE-PCA on $\boldsymbol{W}_{\mathrm{PT}}$ (0.974 ± 0.005 vs. 0.972 ± 0.015) and below the CAV variants on $\boldsymbol{W}_0$ (0.944 ± 0.016 vs. 0.969–0.976); the task-vector baselines stay at 0.86–0.90.

E.3 UNLEARNING WITH VIT

We evaluate UNMERGE on ViT-S/16 Dosovitskiy et al. (2021) (ImageNet-1k weights, 224×224 inputs) finetuned on CIFAR-100 (test 90.2%) with the Trees superclass as forget set, in the $\boldsymbol{W}_{\mathrm{PT}}$ regime. The retrained reference uses the same schedule and reaches 86.1% test accuracy, so equation 8 is well calibrated. The per-layer pipeline (Alg. 1) is applied to the linear projections of the attention and MLP blocks; the `skip-layers` recipe of §4.3 carries over to the transformer's stack of 49

Table 12: CIFAR-100 unlearning results (Vehicles). Best non-retrained in **bold**.

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	0.00	12.12	8.52	0.40	0.44	0.36	0.04	0.04	0.56
Retain Acc ↑	99.97	97.38	99.94	99.91	90.43	91.31	98.82	99.89	99.78	99.32
Test Acc ↑	80.15	73.82	80.05	79.87	68.21	68.85	76.07	79.14	78.09	76.68
ToW ↑	1	0.9125	0.8777	0.9117	0.7934	0.8066	0.9448	**0.9887**	**0.9771**	0.9537
Runtime ↓		23s	110s	112s	**4s**	11s	8s	8s	26s	22s
$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	3.48	9.36	4.36	14.84	15.24	1.80	0.68	0.00	0.00
Retain Acc ↑	99.96	97.46	99.96	99.93	94.66	94.94	97.04	96.59	99.62	99.86
Test Acc ↑	71.7	67.56	73.29	73.70	65.15	65.09	66.83	67.07	70.74	71.85
ToW ↑	1	0.9021	0.8920	0.9370	0.7536	0.7519	0.9069	0.9153	**0.9870**	**0.9975**
Runtime ↓		23s	110s	111s	**4s**	11s	8s	9s	21s	18s

Table 13: TinyImageNet unlearning results (Arthropods). Best non-retrained in **bold**.

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	4.42	15.12	14.70	1.30	2.04	23.68	1.84	0.00	0.00
Retain Acc ↑	99.98	94.33	99.98	99.98	87.49	90.45	98.83	99.04	99.96	99.95
Test Acc ↑	73.62	65.38	74.17	73.91	60.69	62.65	69.68	71.44	73.20	72.72
ToW ↑	1	0.8275	0.8441	0.8505	0.7520	0.7890	0.7247	0.9512	**0.9956**	**0.9907**
Runtime ↓		94s	671s	674s	**18s**	53s	22s	71s	104s	100s
$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Forget Acc ↓	0.0	0.70	10.02	9.56	23.22	25.86	12.42	4.12	9.30	0.12
Retain Acc ↑	99.98	80.14	99.98	99.97	96.34	96.42	85.35	94.91	98.85	98.18
Test Acc ↑	61.86	44.78	60.97	60.78	53.28	53.75	47.46	52.26	56.51	55.76
ToW ↑	1	0.6600	0.8918	**0.8946**	0.6764	0.6570	0.6400	0.8228	0.8488	**0.9210**
Runtime ↓		93s	666s	673s	**18s**	53s	22s	38s	85s	61s

linear layers and, as in ResNet-50, deep-block editing wins: the best cells skip the first 32 (PCA) and 40 (CAV-KMeans) linear layers and edit only the last blocks and the head. Tab. 15 reports all three axes of §4.

Result. UNMERGE-CAV-KMEANS leads every baseline on every axis simultaneously: ToW 0.9975 vs. 0.9949 (SalUn) and 0.9925 (Random Label), at 34s vs. $\sim$474s ($14\times$), with the MIA gap to retraining 1.7–$1.8\times$ smaller (0.061 vs. 0.105–0.109) and the feature-entanglement gap 6.5–$7.4\times$ smaller (0.101 vs. 0.653–0.743). UNMERGE-PCA has the smallest MIA gap (0.056) and runs in 21s; NegGrad+ under-forgets (20% forget accuracy) and damages retain. The retrained reference is strong (test 86.1%), and the deep-layer locality finding of §4.3 transfers to attention-based backbones.

Bias rule. ViT-S/16 has a bias in every edited layer, so the bias treatment of App. B.1 (implementation note iv) could matter here. It does not: with biases left at their finetuned values the two cells move from 0.9975 to 0.9977 (CAV-KMEANS) and from 0.9810 to 0.9792 (PCA), and the α profiles are unchanged to within 0.002.

E.4 PRELIMINARY STUDY ON A LANGUAGE MODEL

We test whether unmerging carries over to language models on TOFU Maini et al. (2024) (`forget10`: the 400 question–answer pairs of 20 fictitious authors; 3600 retain pairs) with Llama-3.2-3B-Instruct Grattafiori et al. (2024), using the checkpoints and the evaluation harness of OpenUnlearning Dorna et al. (2025): $\boldsymbol{W}_{\text{init}}$ is the public instruction-tuned model, $\boldsymbol{W}_{\text{FT}}$ the released model finetuned on all of TOFU, and $\boldsymbol{W}_{\text{RT}}$ the released model finetuned on the retain split, so nothing is trained. We apply the per-layer objective to the linear projections of the last 12–16 of 28 decoder layers, with activation statistics taken at question and answer tokens, and use the closed-form solver of equation 4 (the entries of $\boldsymbol{\tau}_{\text{m}}\boldsymbol{Q}_{\mathcal{F}}$ are about two orders of magnitude smaller here than in the vision models, so the fixed-budget iterate would simply converge to the minimizer). This is a preliminary exploration: each variant receives only light tuning of α, the rank and the edited layers on a proxy that needs no generation (TOFU's normalized answer probability on $\mathcal{F}$ and on held-out retain pairs, relative to $\boldsymbol{W}_{\text{RT}}$), and the baselines run at OpenUnlearning's default settings.

Results. With a retain-deflated PCA forget basis—the forget covariance after projecting out the top 1024 retain directions, i.e. the hard orthogonal projection that our vision runs leave switched off—and λ=0, k=64, α=64, the edit is selective (Tab. 16). Under the official evaluation its forget quality

Table 14: ToW on CIFAR-100 (Trees) over three seeds (seeds 1–3 control the finetuning run, the from-scratch retraining on $\boldsymbol{W}_0$, and the unlearning method; $\boldsymbol{W}_{\text{init}}$ and the data order are fixed), mean $\pm$ sample std. Seed 1 is the cell reported in Tab. 10. Best per column in **bold**.

Method	$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$	$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$
NegGrad+	0.9142 ± 0.0099	0.8280 ± 0.0237
Rand. Label	0.8803 ± 0.0179	0.8391 ± 0.0374
SalUn	0.9587 ± 0.0025	0.9263 ± 0.0357
NegTV	0.8947 ± 0.0054	0.8571 ± 0.0220
NegMerge	0.9010 ± 0.0097	0.8633 ± 0.0237
Kodge et al.	$\mathbf{0.9744 \pm 0.0046}$	0.9439 ± 0.0156
Unmerge-PCA	$\mathbf{0.9716 \pm 0.0146}$	0.9370 ± 0.0208
Unmerge-CAV-KMeans	0.9688 ± 0.0103	$\mathbf{0.9688 \pm 0.0127}$
Unmerge-CAV-Class	0.9408 ± 0.0102	$\mathbf{0.9756 \pm 0.0083}$

Table 15: ViT-S/16 on CIFAR-100 (Trees), $\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$: accuracies (%), ToW (equation 8), runtime, mean MIA gap to retraining over 9 attacks (§4.5), and mean relative feature-entanglement gap (§4.4). Best non-retrained in **bold**. NegGrad+ was not evaluated on the MIA/entanglement axes.

	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	Unmerge-PCA	Unmerge-CAV KMeans
Forget Acc ↓	0.00	20.00	0.12	0.24	0.92	0.16
Retain Acc ↑	99.97	86.03	99.98	99.98	99.64	99.94
Test Acc ↑	86.11	73.43	85.49	85.85	85.44	86.05
ToW ↑	1	0.6012	0.9925	**0.9949**	0.9810	**0.9975**
Runtime ↓		88s	472s	474s	**21s**	34s
MIA gap ↓	0	–	0.105	0.109	**0.056**	**0.061**
Entanglement gap ↓	0	–	0.653	0.743	**0.171**	**0.101**

is 0.131, i.e. the Kolmogorov–Smirnov test no longer separates the edited model from $\boldsymbol{W}_{\text{RT}}$ at the 5% level, at a model utility of 0.630 with the MLP projections of the last 16 layers edited ($\boldsymbol{W}_{\text{RT}}$: 0.650). RMU Li et al. (2024), the strongest of three baselines, reaches 0.094 and 0.633 (GradDiff: 1.6×10^{-27} and 0.618; NPO: 1.8×10^{-14} and 0.601): UNMERGE attains a higher forget quality at a comparable utility (0.003 lower) and lands closer to $\boldsymbol{W}_{\text{RT}}$ on the forget set, where RMU over-forgets (answer probability 0.04 against 0.12 for $\boldsymbol{W}_{\text{RT}}$ and 0.20 for UNMERGE). The edit needs one pass for activation statistics (43s) and no training, whereas the baselines train for 70–290s. A training-free, closed-form edit is thus already competitive with a dedicated language-model unlearning method, which makes scaling UNMERGE a promising extension.

What changes from vision. Two observations guide that extension. *The basis matters more.* With the plain PCA forget basis the forget and retain bases overlap almost completely (mean per-direction overlap 0.87–0.91) and the edit lowers the answer probability on $\mathcal{F}$ and on $\mathcal{R}$ in lockstep, the signature of the random subsets of §4.5; deflating the retain directions restores selectivity. The calibrated separability ratio reads 2.8–3.4 even for the plain basis, so on this model the small absolute score (S=0.03–0.05) is the more informative signal. *The CAV bases trail.* Token-level ridge probes, one per forget author (CAV-CLASS) or per k-means cluster of forget tokens (CAV-KMEANS), are clean (overlap with the retain basis below 0.13) but capture only 1–8% of the forget-token variance, so retain degrades before forgetting completes (best proxy score 0.55–0.59 against 0.85 for the deflated PCA basis and 0.18 for the un-edited model). Concept probes better matched to language, e.g. on subject tokens, are a natural next step.

Scope. The study uses one split, one model and one seed. UNMERGE is tuned lightly on a proxy that uses $\boldsymbol{W}_{\text{RT}}$, whereas the baselines are untuned. The scale α=64 is far above the vision range: the deflated directions carry little of $\boldsymbol{\tau}_{\text{m}}$, so the edit extrapolates along forget-private input directions rather than removing a finetuning delta.

Table 16: TOFU `forget10` with Llama-3.2-3B-Instruct under the official OpenUnlearning evaluation. Forget quality is the p-value of a two-sample test against $\boldsymbol{W}_{\mathrm{RT}}$ (higher is better; 1 for $\boldsymbol{W}_{\mathrm{RT}}$ by definition). UNMERGE: closed form, retain-deflated PCA forget basis, MLP projections of the last 16 layers, lightly tuned. RMU: OpenUnlearning default settings.

Model	Forget quality ↑	Model utility ↑	Forget answer prob.	Forget ROUGE
$\boldsymbol{W}_{\mathrm{FT}}$ (finetuned)	3.6×10^{-27}	0.666	0.951	0.926
$\boldsymbol{W}_{\mathrm{RT}}$ (retain-only)	1	0.650	0.124	0.386
RMU	0.094	**0.633**	0.037	0.225
UNMERGE	**0.131**	0.630	0.200	0.364

Table 17: Feature-level entanglement on TinyImageNet (Dogs & Cats), same protocol as Tab. 3.

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{\mathrm{PT}}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
MMD^2	0.000	0.514	4.531	4.368	0.508	0.542	0.103	0.178	0.090	**0.058**
MMD^2-multi	0.000	0.545	5.373	5.179	0.562	0.579	0.188	0.076	**0.070**	0.074
W_2^2-Bures	0.000	0.419	2.174	2.084	0.477	0.453	**0.031**	0.756	0.176	0.064
Sliced-W_2^2	0.000	0.545	4.668	4.458	0.562	0.564	**0.074**	0.221	0.083	0.132
Mean rel. gap ↓	0.000	0.506	4.187	4.022	0.527	0.534	0.099	0.308	0.105	**0.082**

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
MMD^2	0.000	0.439	4.187	4.186	0.352	0.342	0.012	0.098	0.268	**0.000**
MMD^2-multi	0.000	0.469	4.968	4.978	0.371	0.359	**0.003**	0.209	0.246	0.024
W_2^2-Bures	0.000	0.331	2.302	2.317	0.136	0.123	0.125	0.490	**0.095**	0.129
Sliced-W_2^2	0.000	0.474	4.441	4.469	0.315	0.301	**0.003**	0.433	0.194	0.093
Mean rel. gap ↓	0.000	0.428	3.974	3.988	0.293	0.281	**0.036**	0.308	0.201	0.062

E.5 FEATURE-LEVEL ENTANGLEMENT ON TINYIMAGENET

The CIFAR-100 picture in Tab. 3 is sharpened on TinyImageNet (Tab. 17), measured on the Dogs & Cats split with the same penultimate-feature protocol and same-size retain subsample. UNMERGE-CAV-CLASS/$\boldsymbol{W}_0$ matches the retrained reference to within $\sim$*6%* relative gap (mean 0.062, with MMD^2 at 0.000 indistinguishable from retrained), while Random Label and SalUn deviate by 3.97–$3.99\times$, roughly *64*$\times$ the unmerge gap. The projection of Kodge et al. (2024) posts a smaller mean gap on $\boldsymbol{W}_0$ (0.036), but the two cells are close on this axis: both match the retrained distance to within 2.4% on the two MMD^2 variants and deviate by the same 13% on W_2^2-Bures (in opposite directions), and the difference in the mean comes from Sliced-W_2^2, where CAV-CLASS lands 9% below the retrained distance (forget and retain features slightly more mixed than under retraining) and the projection 0.3% above. What distinguishes them is how they get there: the projection reaches the retrained feature distance while still classifying 18% of the forget set correctly and losing 7 test points (Tab. 2), i.e. by degrading features overall, which ToW and the MIA gap (0.157 vs. 0.058) expose; the entanglement axis alone cannot. On $\boldsymbol{W}_{\mathrm{PT}}$ the spread compresses but Unmerge still leads: CAV-CLASS (0.082) is $\sim6\times$ tighter than NegGrad+, $\sim50\times$ tighter than RL/SalUn, and slightly tighter than the projection (0.099). The W_2^2-Bures column is mean-shift dominated in the $D \gg N$ regime; the MMD^2 and Sliced-W_2^2 metrics, sensitive to higher moments and full distribution shape, rank the best UNMERGE variant ahead of every gradient and task-vector baseline on both initializations.

E.6 MEMBERSHIP INFERENCE: DETAILED RESULTS

We provide per-attack MIA tables for both datasets, expanding the summary in §4.5. Both protocols are identical (RUM/SalUn-style shadow/target split, nine attack signals, gap-to-retrained metric). Three details matter when reading the numbers. The attack accuracy on $\mathcal{F}$ averages the attacker's true-positive rate on retain samples and its true-negative rate on forget samples, so damage to retain accuracy also registers as a privacy gap. The SVC attackers are trained on an unbalanced shadow set ($\sim83\%$ members) without class weighting, and the correctness signal appears twice (threshold and SVC give the same attacker). Finally, the class-level tables report the mean over signals of the *absolute* gap, whereas the instance-level table (Tab. 20) reports the *signed* gap of the signal-averaged attack accuracy, in which per-signal errors of opposite sign can cancel; App. E.7 gives both aggregates for the coherent protocol.

Table 18: MIA gap to the retrained reference on CIFAR-100 (Trees): $|\mathrm{att}_{\mathcal{F}}^{\mathrm{unl}} - \mathrm{att}_{\mathcal{F}}^{\mathrm{RT}}|$ per attack signal (lower is better; $\boldsymbol{W}_{\mathrm{RT}}$ is 0 by construction).

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_{\mathrm{PT}}$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Threshold-Conf.	0.000	0.034	0.017	**0.011**	0.036	0.035	0.015	0.024	0.028	0.033
Threshold-Ent.	0.000	0.033	0.171	0.173	**0.027**	0.048	0.104	0.160	0.154	0.144
Threshold-M-Ent.	0.000	0.035	0.021	**0.007**	0.036	0.035	0.018	0.024	0.027	0.040
Threshold-Corr.	0.000	0.016	0.061	0.022	0.025	0.025	0.002	**0.001**	0.003	0.007
SVC-Conf.	0.000	0.009	**0.002**	0.003	0.020	0.021	0.004	0.003	**0.002**	0.006
SVC-Ent.	0.000	0.211	0.209	0.207	0.261	0.242	**0.065**	0.197	0.194	0.154
SVC-M-Ent.	0.000	0.049	**0.001**	0.002	0.024	0.027	0.003	0.002	0.003	0.006
SVC-Corr.	0.000	0.015	0.061	0.022	0.023	0.025	0.002	**0.001**	0.003	0.007
SVC-Probs.	0.000	0.203	0.291	0.290	0.201	0.190	**0.038**	0.288	0.258	0.233
Mean|gap| ↓	0.000	0.067	0.093	0.082	0.073	0.072	**0.028**	0.078	0.075	0.070

$\boldsymbol{W}_{\mathrm{init}} = \boldsymbol{W}_0$	$\boldsymbol{W}_{\mathrm{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Threshold-Conf.	0.000	0.037	0.010	0.003	0.022	0.026	0.007	0.008	**0.001**	**0.001**
Threshold-Ent.	0.000	0.103	0.063	0.065	0.072	0.057	**0.017**	0.040	0.046	0.054
Threshold-M-Ent.	0.000	0.040	0.007	0.002	0.025	0.032	0.004	0.005	0.006	**0.001**
Threshold-Corr.	0.000	0.042	0.077	0.016	0.029	0.027	0.005	0.006	**0.004**	0.006
SVC-Conf.	0.000	0.041	**0.000**	**0.000**	0.019	0.021	0.003	0.003	0.003	0.002
SVC-Ent.	0.000	0.281	0.143	0.139	0.295	0.231	**0.017**	0.354	0.084	0.093
SVC-M-Ent.	0.000	0.079	**0.000**	0.001	0.023	0.025	0.004	0.005	0.002	0.001
SVC-Corr.	0.000	0.041	0.077	0.016	0.030	0.027	0.005	0.006	**0.004**	0.006
SVC-Probs.	0.000	0.314	0.184	0.183	0.309	0.294	0.094	0.316	**0.080**	0.094
Mean|gap| ↓	0.000	0.109	0.062	0.047	0.092	0.082	**0.017**	0.083	0.026	0.029

CIFAR-100 (Trees). Tab. 18 reports per-attack gaps. On $\boldsymbol{W}_0$ the projection of Kodge et al. (2024) has the smallest mean gap (0.017), followed by UNMERGE-CAV-KMEANS (0.026) and UNMERGE-CAV-CLASS (0.029), 1.6–1.8× tighter than SalUn (0.047), 2.1–4.2× tighter than Random Label (0.062) and NegGrad+ (0.109), and 2.8–3.5× tighter than NegTV and NegMerge (0.082–0.092); the SVC-probs and SVC-entropy attacks (full softmax-vector attackers, the strongest in our set) are where the gap closes most, with the CAV variants cutting both by 1.5–2.3× over the saliency baselines. On $\boldsymbol{W}_{\mathrm{PT}}$ the projection leads again (0.028) and the spread among the other methods is tight: NegGrad+ 0.067, UNMERGE-CAV-CLASS 0.070, NegMerge 0.072, NegTV 0.073, UNMERGE-CAV-KMEANS 0.075, UNMERGE-PCA 0.078, SalUn 0.082, Random Label 0.093. NegGrad+'s small edge over UNMERGE comes at a 2.8-point retain and 6.2-point test drop (Tab. 10), so the joint utility + privacy ranking still favors UNMERGE.

TinyImageNet (Dogs & Cats). On $\boldsymbol{W}_{\mathrm{PT}}$, all three UNMERGE variants sweep the table (Tab. 19): UNMERGE-PCA (0.009) and UNMERGE-CAV-CLASS/UNMERGE-CAV-KMEANS (0.018/0.020) are 2–5× tighter than the saliency baselines (Random Label/SalUn ∼0.043), 9–20× tighter than NegGrad+ (0.177), and 13–29× tighter than the task-vector baselines and the projection of Kodge et al. (2024) (0.254–0.261). On $\boldsymbol{W}_0$, Random Label (0.034) and SalUn (0.040) appear to lead the CAV variants (0.058). This is an artifact of *under-unlearning*: the baseline configurations used on this split (Random Label lr=10^{-3}, SalUn τ=0.5/lr=10^{-3}) leave about 9% forget accuracy (Tab. 2), so their predictions on $\mathcal{F}$ have barely moved from the finetuned model and the MIA gap closes by inaction rather than by erasure. The entanglement axis exposes it: the same checkpoints deviate from the retrained reference by ∼4× its value, while UNMERGE-CAV-CLASS/$\boldsymbol{W}_0$ matches retraining to within 6% (Tab. 17). Neither axis suffices alone: the MIA gap can be closed by leaving the model unchanged, and the feature distance can be matched by degrading the features overall, as the projection does on this split (App. E.5). Read together with accuracy, the two axes single out the methods that forget without either failure.

E.7 INSTANCE-LEVEL FORGETTING: DETAILS

Protocols and references. Coherent: a fixed random half of the 2500 Trees training images forms $\mathcal{F}$ ($|\mathcal{F}|$=1250; the draw is not stratified, giving 237–262 images per class) and the other half stays in $\mathcal{R}$. Random: 2500 training images drawn uniformly from all 100 classes. Each protocol and initialization has its own retrained reference (coherent: $\boldsymbol{W}_0$ F/R/T 70.64/99.96/76.27, $\boldsymbol{W}_{\mathrm{PT}}$ 74.24/99.98/84.93; random: $\boldsymbol{W}_0$ 74.92/99.97/74.73, $\boldsymbol{W}_{\mathrm{PT}}$ 84.56/99.98/84.02). Baselines use the class-level configurations of §4.1, except NegGrad+ on the coherent protocol: its class-level setting over-forgets there (gradient ascent on half a class erases the class; forget accuracy 26%/11%, ToW 0.4970/0.3251), so we tune it on a 40-run grid over its forget/retain weight $\{0.3, \ldots, 0.99\}$, learning rate $\{10^{-5}, \ldots, 10^{-4}\}$ and epochs $\{5, \ldots, 12\}$. UNMERGE uses, on the random protocol,

Table 19: MIA gap to the retrained reference on TinyImageNet (Dogs & Cats), same protocol as Tab. 18.

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Threshold-Conf.	0.000	0.044	**0.007**	0.009	0.051	0.044	0.046	0.016	0.008	0.012
Threshold-Ent.	0.000	0.108	0.012	0.011	0.113	0.095	0.076	**0.005**	0.012	0.006
Threshold-M-Ent.	0.000	0.048	**0.006**	0.011	0.051	0.048	0.048	0.018	0.008	0.009
Threshold-Corr.	0.000	0.025	0.081	0.077	0.052	0.046	0.042	0.007	**0.000**	**0.000**
SVC-Conf.	0.000	0.028	**0.001**	**0.001**	0.500	0.500	0.500	0.002	**0.001**	**0.001**
SVC-Ent.	0.000	0.431	0.066	0.066	0.431	0.431	0.413	**0.021**	0.055	0.043
SVC-M-Ent.	0.000	0.048	**0.000**	**0.000**	0.287	0.278	0.301	0.002	**0.000**	0.001
SVC-Corr.	0.000	0.500	0.081	0.077	0.500	0.500	0.500	0.007	**0.000**	**0.000**
SVC-Probs.	0.000	0.360	0.137	0.137	0.363	0.362	0.362	**0.007**	0.099	0.092
Mean|gap| ↓	0.000	0.177	0.044	0.043	0.261	0.256	0.254	**0.009**	0.020	0.018

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{0}$	$\boldsymbol{W}_{\text{RT}}$	NegGrad+	Rand. Label	SalUn	NegTV	NegMerge	Kodge et al.	Unmerge-PCA	Unmerge-CAV KMeans	Unmerge-CAV Class
Threshold-Conf.	0.000	0.064	**0.005**	0.009	0.064	0.063	0.063	0.047	0.028	0.025
Threshold-Ent.	0.000	0.171	0.018	**0.017**	0.092	0.089	0.061	0.107	0.039	0.040
Threshold-M-Ent.	0.000	0.067	**0.003**	0.008	0.065	0.064	0.065	0.052	0.029	0.025
Threshold-Corr.	0.000	0.072	0.047	0.045	0.076	0.076	0.107	0.104	0.025	**0.013**
SVC-Conf.	0.000	0.500	**0.001**	**0.001**	0.111	0.117	0.170	0.500	0.028	0.006
SVC-Ent.	0.000	0.435	**0.083**	0.137	0.435	0.435	0.304	0.435	0.128	0.171
SVC-M-Ent.	0.000	0.137	**0.001**	0.002	0.214	0.200	0.191	0.311	0.052	0.010
SVC-Corr.	0.000	0.500	0.047	0.045	0.500	0.500	0.106	0.500	0.024	**0.011**
SVC-Probs.	0.000	0.397	**0.101**	**0.101**	0.383	0.380	0.349	0.397	0.172	0.220
Mean|gap| ↓	0.000	0.260	**0.034**	0.040	0.215	0.214	0.157	0.272	0.058	0.058

Table 20: Instance-level forgetting on CIFAR-100 (Trees), ResNet-50. *Coherent*: half of the training images of each Trees class ($|\mathcal{F}|$=1250); *Random*: 2500 images drawn uniformly from all 100 classes. Accuracies in %. $\text{att}_{\mathcal{F}}$: mean attack accuracy on $\mathcal{F}$ over the 9 MIA signals of §4.5, with the signed gap to $\boldsymbol{W}_{\text{RT}}$ in parentheses (positive = forget samples more identifiable than under retraining). Ent.: mean relative entanglement gap of Tab. 3. Best non-retrained ToW and entanglement per protocol in **bold**. †On the coherent protocol NegGrad+ is tuned on a 40-run grid (forget/retain weight, learning rate, epochs), because its class-level setting over-forgets there (ToW 0.4970/0.3251); Random Label and SalUn keep their class-level settings, and the UNMERGE cells sweep only α (five values per basis). The identical forget/retain accuracies of UNMERGE-PCA on $\boldsymbol{W}_0$ and UNMERGE-CAV-KMEANS on $\boldsymbol{W}_{\text{PT}}$ are genuine (901 of 1250 forget images correct in both; test accuracies and losses differ).

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{0}$	Coherent ($\|\mathcal{F}\|$=1250)					Random ($\|\mathcal{F}\|$=2500)				
	Forget	Retain	ToW ↑	$\text{att}_{\mathcal{F}}$ (gap)	Ent. ↓	Forget	Retain	ToW ↑	$\text{att}_{\mathcal{F}}$ (gap)	Ent. ↓
$\boldsymbol{W}_{\text{RT}}$	70.64	99.96	1	0.700	0	74.92	99.97	1	0.681	0
NegGrad+†	71.60	98.62	0.9368	0.624 (−0.076)	0.46	99.92	99.97	0.7302	0.502 (−0.180)	0.62
Rand. Label	88.72	99.96	0.8137	0.839 (+0.140)	0.41	69.80	99.96	0.9405	0.872 (+0.191)	21.9
SalUn	87.44	99.93	0.8243	0.825 (+0.125)	0.29	72.20	99.89	**0.9595**	0.839 (+0.158)	15.5
Unmerge-PCA	72.08	99.15	**0.9665**	0.712 (+0.013)	**0.15**	99.64	99.77	0.7509	0.502 (−0.180)	0.67
Unmerge-CAV-KMeans	73.60	99.25	0.9534	0.691 (−0.009)	0.27	97.32	98.19	0.7515	0.506 (−0.176)	**0.60**

$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$	Coherent ($\|\mathcal{F}\|$=1250)					Random ($\|\mathcal{F}\|$=2500)				
	Forget	Retain	ToW ↑	$\text{att}_{\mathcal{F}}$ (gap)	Ent. ↓	Forget	Retain	ToW ↑	$\text{att}_{\mathcal{F}}$ (gap)	Ent. ↓
$\boldsymbol{W}_{\text{RT}}$	74.24	99.98	1	0.686	0	84.56	99.98	1	0.630	0
NegGrad+†	74.96	99.08	**0.9427**	0.620 (−0.066)	0.48	100.0	99.97	0.8406	0.500 (−0.130)	0.55
Rand. Label	85.84	99.92	0.8639	0.852 (+0.165)	0.55	69.04	99.93	0.8328	0.879 (+0.249)	81.7
SalUn	83.12	99.92	0.8970	0.841 (+0.155)	0.41	73.68	99.88	**0.8788**	0.850 (+0.221)	50.1
Unmerge-PCA	67.12	99.10	0.9002	0.642 (−0.044)	0.16	87.40	90.89	0.7804	0.519 (−0.111)	**0.50**
Unmerge-CAV-KMeans	72.08	99.15	0.9417	0.677 (−0.010)	**0.11**	98.64	99.09	0.8423	0.505 (−0.125)	0.54

the protocol grid of App. D.1 (`skip-layers` $\{35, \ldots, 50\}$, $\lambda \in \{5, 20, 50\}$, $\gamma \in \{0.25, 0.5\}$, PCA $k \in \{64, 128\}$, CAV k=16, fine α), and on the coherent protocol its class-level configuration with only α swept (five values per basis). All cells report the best final ToW of their grid, and every reported optimum is interior in the swept quantity.

Coherent protocol, per group: the aggregate hides a concept-level edit. The forget half and the retained half of a Trees class are i.i.d. draws, so no activation subspace separates them. Tab. 21 splits the retain and test accuracies of Tab. 20 accordingly. UNMERGE matches the retrained accuracy on the forget half by lowering the *whole class*: the retained half falls to the same level (67–73%, against $\geq$99.9% under retraining) and the Trees test accuracy falls 18–26 points below the retrained model's, while the other 95 classes stay within 2.2 test points of retraining. The tuned NegGrad+ does the same. The aggregate retain accuracy moves by at most 1.3 points because the retained half is 2.6% of $\mathcal{R}$, which is why ToW ranks these cells first. Random Label and SalUn are the mirror image: they keep the retained half ($\geq$99%) and most of the Trees test accuracy, and leave the forget half 9–18

Table 21: Coherent protocol, per-group accuracy (%) of the cells of Tab. 20. Forget half: the 1250 forget images; retained half: the 1250 Trees training images that stay in $\mathcal{R}$; other: the 47,500 retain images of the other 95 classes; Trees test / other test: the 500 / 9500 test images. Retraining keeps the retained half at $\geq 99.9\%$ while the forget half drops to test level. UNMERGE and the tuned NegGrad+ (†) lower both halves alike and the Trees test accuracy with them; Random Label and SalUn keep the retained half and under-forget the forget half.

	$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$					$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$				
	Forget half	Retained half	Other retain	Trees test	Other test	Forget half	Retained half	Other retain	Trees test	Other test
$\boldsymbol{W}_{\text{RT}}$	70.64	99.92	99.96	66.40	76.79	74.24	100.00	99.97	74.80	85.46
$\boldsymbol{W}_{\text{FT}}$ (un-edited)	99.92	100.00	99.96	73.80	77.69	99.92	100.00	99.97	74.00	85.27
NegGrad+†	71.60	71.12	99.35	44.20	73.61	74.96	75.44	99.70	47.20	82.51
Rand. Label	88.72	99.92	99.96	67.40	77.44	85.84	99.76	99.93	67.80	83.49
SalUn	87.44	99.36	99.95	68.40	77.63	83.12	99.04	99.94	69.20	84.19
Unmerge-PCA	72.08	72.40	99.85	45.80	76.65	67.12	67.12	99.94	48.40	84.53
Unmerge-CAV-KMeans	73.60	73.36	99.93	44.20	76.85	72.08	73.20	99.83	56.80	83.32

points above the retrained level while marking it (attack accuracy above retraining, Tab. 20). No method reproduces retraining on this protocol; UNMERGE treats the request as a partial removal of the concept, which is the behavior its construction implies and the reason we scope it to concept-level requests (§4.5).

Random protocol: the lockstep ceiling. The retrained model itself barely forgets a random subset (forget accuracy 75/85%, close to its test accuracy), so the untouched $\boldsymbol{W}_{\text{FT}}$ already scores ToW 0.73/0.84, and a 60-configuration sweep confirms that the ceiling is not under-tuning: forget and retain accuracy move together at every α, and matching the retrained forget accuracy on $\boldsymbol{W}_{\text{PT}}$ costs 13 points of retain. Random Label and SalUn reach ToW up to 0.96 only by *marking* the forget samples: attack accuracy on $\mathcal{F}$ rises to 0.84–0.88, above the retrained model's 0.63–0.68, and their forget features become separable from retain features (relative entanglement gap 16–82 against ≤ 0.7 for UNMERGE). The cells in Tab. 20 are the ToW maxima over genuine edits (no-op edits, whose forget accuracy stays at the un-edited level, excluded; ties within 0.005 broken by entanglement, then signed MIA gap).

Instance-level and concept-level forgetting are different problems. Triantafillou et al. (2026) distinguish *untraining*—reversing the effect of having trained on the given forget samples, whose reference is the model retrained on $\mathcal{S} \setminus \mathcal{F}$—from *unlearning* the distribution or concept that the samples represent, and note that the two coincide when $\mathcal{F}$ contains every training instance of the concept, which is our class-level protocol. Otherwise they pull in opposite directions: while the concept stays represented in $\mathcal{R}$, the untraining target still predicts the forget samples correctly and the right edit is close to none, whereas removing the concept calls for a large one; Cooper et al. (2025) describe the same mismatch between removing observed information and suppressing what a model has generalized. Our random protocol is the extreme case: its forget samples are i.i.d. draws from the training distribution, so the retrained forget accuracy matches test accuracy and the un-edited $\boldsymbol{W}_{\text{FT}}$ already scores ToW 0.73/0.84, in line with Zhao et al. (2024), who find that for forget sets that are not memorized the original model is already close to retraining. What remains is per-sample memorization, on which an edit built to generalize over what the forget samples share has nothing to act—which the separability score reports beforehand.

Separability diagnostics. The separability score is $S = \text{fvar} \cdot (1 - \text{overlap})$, the fraction of forget variance captured by $\boldsymbol{Q}_{\mathcal{F}}$ times one minus the mean per-direction overlap of $\boldsymbol{Q}_{\mathcal{F}}$ with $\boldsymbol{Q}_{\mathcal{R}}$; both factors are available after Phase 1, before any edit. Tab. 22 reports them. Within a dataset S separates the regimes cleanly (2.4–2.5$\times$ between class-level and random CIFAR-100 in both initializations, through different factors: $\boldsymbol{W}_0$ loses the private subspace through overlap, $0.61 \to 0.83$, and $\boldsymbol{W}_{\text{PT}}$ through captured variance, $0.63 \to 0.28$). Across datasets it does not: TinyImageNet class-level runs measure $S \approx 0.07$–0.10, the CIFAR-100 "hard" range, yet unlearn well, because from-scratch TinyImageNet features are less structured and S is basis-dependent. The ratio $S_{\text{task}}/S_{\text{rand}}$ against a same-size random subset of the same dataset removes both dependencies at the cost of one extra forward pass.

Table 22: Phase-1 separability diagnostics over edited layers: fraction of forget variance captured by $\boldsymbol{Q}_{\mathcal{F}}$ (fvar), mean per-direction overlap between $\boldsymbol{Q}_{\mathcal{F}}$ and $\boldsymbol{Q}_{\mathcal{R}}$, and $S = \text{fvar} \cdot (1 - \text{overlap})$. CIFAR-100: CAV-KMeans basis (k=16); the coherent row uses the PCA basis of its winning cell. TinyImageNet: bases of the official $\boldsymbol{W}_0$ cells (the diagnostic was not logged for the $\boldsymbol{W}_{\text{PT}}$ TinyImageNet runs). Last column: S relative to the same-dataset random-subset value.

Dataset	Forget request	fvar	overlap	S	S/S_{rand}
CIFAR-100 ($\boldsymbol{W}_0$)	Trees, class-level	0.438	0.611	0.170	2.4
	Trees, coherent half-classes	–	–	0.17	2.4
	Random subset (2500)	0.421	0.832	0.071	1
CIFAR-100 ($\boldsymbol{W}_{\text{PT}}$)	Trees, class-level	0.629	0.729	0.170	2.5
	Random subset (2500)	0.281	0.755	0.069	1
TinyImageNet ($\boldsymbol{W}_0$)	Dogs & Cats, CAV-KMeans	0.252	0.628	0.094	–
	Arthropods, CAV-Class	0.135–0.143	0.347–0.460	0.073–0.093	–
	Arthropods, CAV-KMeans	0.259	0.611	0.101	–
	Arthropods, PCA	0.526	0.864	0.072	–

Table 23: Decomposition validation on CIFAR-100 (Trees), ResNet-50. Weight space: layer-averaged relative Frobenius error and size-weighted cosine of $\boldsymbol{\tau}_{\mathcal{R}}$ vs. $\boldsymbol{\tau}^{\star}_{\mathcal{R}}$ over all weight matrices. Output space: $D_{\text{KL}}(p_{\text{UL}} \| p_{\text{RT}})$ per split (lower is closer to retraining). Floor: two independent retrains from the same $\boldsymbol{W}_{\text{init}}$. Best non-reference per KL column in **bold**; the $\boldsymbol{W}_0$ weight-space columns are uninformative (see text).

	$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_{\text{PT}}$					$\boldsymbol{W}_{\text{init}} = \boldsymbol{W}_0$				
Method	rel. err	cos	$\text{KL}_{\mathcal{F}}$	$\text{KL}_{\mathcal{R}}$	KL_{test}	rel. err	cos	$\text{KL}_{\mathcal{F}}$	$\text{KL}_{\mathcal{R}}$	KL_{test}
Retrain floor (2 runs)	0.718–0.731	0.684–0.688	–	–	–	1.539–1.669	0.005	–	–	–
$\boldsymbol{W}_{\text{FT}}$ (no unlearning)	0.751	0.691	–	–	–	1.579	0.006	–	–	–
NegGrad+	0.751	0.691	5.89	0.397	1.151	1.579	0.006	4.89	0.500	1.293
Rand. Label	0.823	0.682	6.62	**0.014**	0.704	1.515	0.006	5.20	**0.011**	0.755
SalUn	0.908	0.684	6.62	0.040	**0.698**	1.547	0.006	5.21	0.024	0.736
Unmerge-PCA	0.750	0.688	6.03	0.172	0.864	1.572	0.006	4.99	3.873	3.471
Unmerge-CAV-KMeans	0.754	0.683	**5.29**	0.107	0.825	1.577	0.006	**2.89**	0.043	0.717
Unmerge-CAV-Class	0.754	0.685	7.15	0.192	1.049	1.577	0.006	3.31	0.029	**0.708**

E.8 Decomposition Validation: Details

Accuracy-based scores are proxies; the quantity the method acts on, $\boldsymbol{\tau}_{\mathcal{R}} = \boldsymbol{\tau}_{\text{m}} - \alpha\boldsymbol{\tau}_{\mathcal{F}}$, can be compared with $\boldsymbol{\tau}^{\star}_{\mathcal{R}} = \boldsymbol{W}_{\text{RT}} - \boldsymbol{W}_{\text{init}}$ directly. For every checkpoint on Trees we compute, over all weight tensors with at least two dimensions (53–54 layers, not only the edited ones), the relative Frobenius error $\|\boldsymbol{\tau}_{\mathcal{R}} - \boldsymbol{\tau}^{\star}_{\mathcal{R}}\|_F / \|\boldsymbol{\tau}^{\star}_{\mathcal{R}}\|_F$ averaged over layers and the size-weighted cosine between $\boldsymbol{\tau}_{\mathcal{R}}$ and $\boldsymbol{\tau}^{\star}_{\mathcal{R}}$ (for baselines $\boldsymbol{\tau}_{\mathcal{R}} = \boldsymbol{W}_{\text{UL}} - \boldsymbol{W}_{\text{init}}$), and the predictive divergence $D_{\text{KL}}(p_{\text{UL}} \| p_{\text{RT}})$ on the forget, retain, and test splits (Tab. 23). Two references anchor the weight-space columns: the un-edited $\boldsymbol{W}_{\text{FT}}$, and the retrain-to-retrain noise floor obtained by retraining twice more from the same $\boldsymbol{W}_{\text{init}}$ with different data orders. On $\boldsymbol{W}_0$ every checkpoint, including independent retrains, has cosine ≈ 0.005 and relative error 1.5–1.7 to the reference: from-scratch runs converge to unrelated basins, so only the output-space columns are meaningful there. On $\boldsymbol{W}_{\text{PT}}$ the floor is 0.718–0.731 and $\boldsymbol{W}_{\text{FT}}$ sits at 0.751; all UNMERGE variants stay at the un-edited level (0.750–0.754: the minimal edit does no weight-space harm), whereas Random Label (0.823) and SalUn (0.908) move *away* from the retrained solution. Output space discriminates more sharply in the same direction: UNMERGE-CAV-KMEANS has the smallest forget-set divergence in both regimes (2.89 on $\boldsymbol{W}_0$ vs. 4.9–5.2 for the baselines; 5.29 vs. 5.9–6.6 on $\boldsymbol{W}_{\text{PT}}$), because relabeling pushes forget inputs confidently to wrong classes—far from a model that never saw them—while subspace subtraction lands closer (cf. §4.4). *Caveats.* (a) CAV-CLASS at α=2.5 on $\boldsymbol{W}_{\text{PT}}$ matches the retrained forget accuracy to within 0.12 points but has the largest forget-set divergence of any cell (7.15): an aggressive α over-shoots distributionally while matching accuracy, so α should be selected on a distributional criterion when one is available. (b) UNMERGE-PCA on $\boldsymbol{W}_0$ has large divergence on every split (3.87 on retain despite 98.75% retain accuracy): the argmax survives but calibration does not, consistent with its entanglement outlier in Tab. 3 and with the regime split of §4.2.

F LIMITATIONS AND FUTURE WORK

Scope. UNMERGE is a training-free edit for concept-level forget requests, i.e. requests that occupy a separable activation subspace, such as whole classes and semantic groups (§4.5). On a coherent instance subset it lowers the forget and retained halves of a class alike (Tab. 21). Random instance subsets have none—they pose a per-sample de-memorization problem rather than a concept-removal one (Triantafillou et al., 2026) (App. E.7)—and no subspace edit can move their forget accuracy without moving retain; gradient-on-retain methods can, at 4–$23\times$ the cost and at the price of a membership signature on the forget set. The Phase-1 separability ratio flags the regime beforehand but needs a same-size random-subset control for calibration (App. E.7). *Requirements.* The method is optimization-time data-free, not data-free: it needs forward passes over $\mathcal{F} \cup \mathcal{R}$, CAV-CLASS needs forget labels (App. B.3), and the $\boldsymbol{W}_{\mathrm{PT}}$ regime needs the *same* pretraining checkpoint—substituting a different one is catastrophic, while $\boldsymbol{W}_{\mathrm{init}}{=}0$ is a safe fallback (App. D.4). *Fidelity and solver.* Unmerging does no weight-space harm and has the smallest forget-set divergence to $\boldsymbol{W}_{\mathrm{RT}}$ among the methods of Tab. 23, but does not reconstruct $\boldsymbol{\tau}^{\star}_{\mathcal{R}}$ in weight space beyond the un-edited model (§4.5). At class level a logit mask on the un-edited model already matches retraining on ToW and the MIA gap, and a linear probe on frozen features still recovers the forget classes from every approximate method we tested, ours included; the feature-entanglement axis separates UNMERGE from such a mask on $\boldsymbol{W}_0$ and on TinyImageNet, but not on CIFAR-100/$\boldsymbol{W}_{\mathrm{PT}}$, where the un-edited features already sit near retraining. We therefore claim fidelity to retraining in outputs and feature distribution, not that class information becomes undecodable. The reported cells use an early-stopped Adam iterate rather than the exact minimizer, so their λ is nominal and the PCA variants depend on the step budget, which we fix throughout (App. D.3). *Breadth and future work.* The instance-level, task-vector-validation and seed-variance studies use one CIFAR-100 superclass, and the cost model is validated on ResNet-50 and ViT-S/16; at language-model scale we only have the preliminary study of App. E.4. The framework itself is general: the CAV construction accepts arbitrary positive/negative pools, so concept labels (toxicity, bias, copyrighted style) lift UNMERGE to language and vision-language models without changing the objective, and the layerwise structure fits LoRA-merged and mixture-of-experts checkpoints; explicit solvers (App. D.3), continual forget requests, and adaptive per-layer rank and skip selection are further directions.