RECAST: BEYOND WEIGHT REINITIALIZATION FOR MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning is commonly evaluated through the behavior a model exhibits after data is removed. Yet similar behavioral outcomes can be achieved through different model modifications, raising a basic question: what kind of internal state does an unlearning procedure actually produce? We study this question across classification and generative concept removal using behavioral and internal diagnostics, and additionally examine recovery under subsequent training in our main classification comparison. Even among behaviorally similar checkpoints, internal states can differ substantially, and forgotten behavior can return at markedly different rates. Motivated by these differences, we introduce **RECAST** (**RE**tain-guided **C**ontraction via **A**lignment **S**caled **T**ransformation), a retain-guided weight intervention that uses gradient-weight alignment to continuously contract structural weight groups. When the full retain set is available, as in our classification experiments, RECAST uses retained data only. RECAST achieves strong behavioral agreement with retraining, produces internal-state measurements closer to the retrained reference than most evaluated baselines, and recovers forgotten behavior more slowly than most of them. We further extend RECAST to nudity concept removal in FLUX, adapting the method to the different data-access constraints of large generative models. RECAST strongly suppresses the target concept under both standard and adversarial prompts while preserving benign generation quality. Together, our results show that behavioral outcomes, internal state, and recovery dynamics capture complementary properties of unlearned models, and position retain-guided continuous contraction as an alternative to reset-based weight interventions.

1 INTRODUCTION

Machine unlearning aims to remove the influence of selected training data from a trained model while keeping what the model learned from the remaining data (Ginart et al., 2019; Bourtoule et al., 2021). The most straightforward solution is to retrain the model from scratch without the removed data. This provides a natural reference for unlearning, but repeating full training for every deletion is too expensive in practice. Approximate methods therefore reuse the trained model and alter it through a targeted unlearning procedure instead of rebuilding it from scratch.

Unlearning methods are usually assessed through their outputs: the model should stop predicting the forgotten classes while continuing to perform well on the retained data. However, similar output behavior can result from different changes to the model. A model can reach near-zero accuracy on a forgotten class while its intermediate representations retain substantial structure associated with that class (Figure 1). In particular, encoder features can remain class-separable even when the classifier no longer maps them to the corresponding label (Gao et al., 2026; Kim et al., 2026). Forgotten behavior can also re-emerge during subsequent optimization, and methods that appear similar immediately after unlearning can differ in how quickly it returns (Ha et al., 2026; Siddiqui et al., 2025). Output-based metrics alone therefore do not determine what internal state an unlearning procedure actually leaves the model in.

We therefore characterize unlearned models from three angles. The *behavioral profile* contains measurements derived from the model's outputs, including forget and retain accuracy and membership-inference outcomes. The *internal profile* measures properties of intermediate representations and their relation to the classifier, including class geometry, representation similarity to retraining, and feature–classifier alignment. Finally, the *recovery profile* measures how quickly forgotten behav-

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

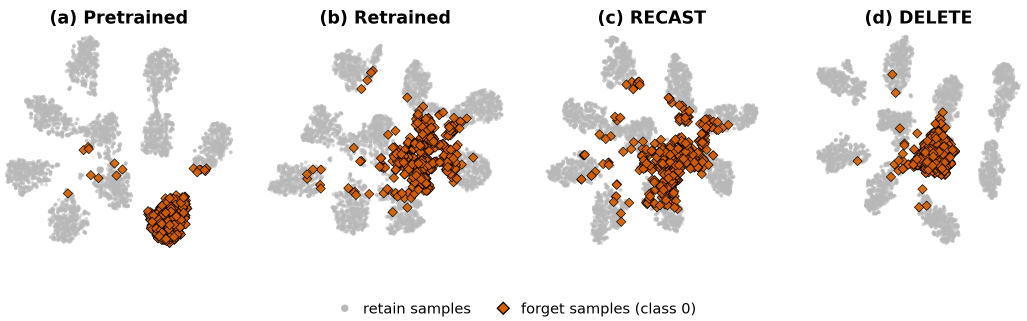

Figure 1: Representation geometry after unlearning on CIFAR-10. t-SNE of post-GAP, pre-classifier representations for 5,000 class-balanced test examples. Retained samples are shown in light gray; orange outlined diamonds denote forgotten-class examples (class 0). Each model is embedded independently and then rigidly aligned to the Retrained panel using class centroids; absolute cross-panel distances are therefore not comparable.

ior returns when training continues. These three perspectives characterize distinct properties of the post-unlearning model rather than interchangeable definitions of successful forgetting.

Importantly, successful unlearning does not require eliminating every feature that is informative about the forgotten classes. Even a model retrained from scratch without them may still separate those classes to some degree, since features learned from retained classes may also be informative for classes excluded from training. We therefore interpret these measurements relative to a retain-only retrained reference whenever such a reference is available; not every experimental setting supports the same diagnostics.

Several recent unlearning methods modify selected parts of the weight space before repairing retained utility. WIG (Lee et al., 2026) selects individual weights using gradient-based criteria and partially reinitializes them, while Scissorhands (Wu & Harandi, 2024) resets individual connections selected by connection sensitivity. DEL (Torkzadehmahani et al., 2025) instead localizes a small subset of parameters associated with memorized examples, resets them, and subsequently fine-tunes the localized subset. These methods differ in how they localize and modify model structure, but they share an intervention–repair pattern: a targeted parameter intervention is followed by training that restores retained utility. This raises a second design question in addition to *where* to intervene: how much of the existing structure should be disrupted before the model has to adapt again during repair?

A useful perspective comes from DASH (Shin et al., 2024), which studies loss of neural-network plasticity—the reduced ability of a trained model to adapt as learning continues. DASH uses direction-aware contraction to weaken less-supported weight directions while preserving structure that remains useful for subsequent learning.

We turn this principle into **RECAST** (**RE**tain-guided **C**ontraction via **A**lignment **S**caled **T**ransformation). Within each selected weight tensor, RECAST considers structural parameter groups—for example, individual filters in a convolutional layer—and measures the alignment between each group and the retain-loss descent direction. Alignment determines the relative contraction strength across groups, while a single hyperparameter limits the maximum contraction applied to any group. Groups better supported by retained data are therefore preserved to a greater degree, whereas less-supported groups are contracted further.

The contracted model is then adapted to the retain objective through a short retain-only repair phase, which we call *consolidation*. RECAST is thus a two-stage intervention–repair procedure rather than a standalone contraction rule: contraction modifies the existing parameter structure, and consolidation restores retained utility from the resulting state. In our classification setting, the full retain set is available, so both stages use retained data only, without requiring examples from the forget set. Compared with WIG, Scissorhands, and DEL, RECAST therefore differs in both the signal that shapes the intervention and the intervention itself: it uses retained data to determine what should be preserved, and contracts weight groups rather than reinitializing selected parameters.

Our experiments show that similar behavioral outcomes can correspond to substantially different post-unlearning states. Even after selecting class-unlearning checkpoints by behavioral agreement with retraining, their internal profiles can differ markedly. A separate controlled recovery comparison likewise shows large differences in how quickly forgotten behavior returns. RECAST remains among the strongest behavioral matches to retraining, while its internal profile is close to the retrained reference and recovery is slower than for most evaluated baselines. These results show that matching the behavioral endpoint does not determine the internal or dynamic properties of the resulting model.

Finally, we test whether the same contraction principle transfers beyond classification by adapting RECAST to nudity concept removal in FLUX, a flow-matching transformer. In this open-ended generative setting, a complete retain set is not practically available, so we adapt how the intervention is localized using source-model generations from target-concept prompts rather than applying the classification procedure unchanged. RECAST strongly suppresses nudity under both standard and adversarial prompts while preserving benign-generation quality.

Our contributions can be summarized as follows:

- **Method.** We introduce RECAST, a retain-guided weight-contraction method followed by retain-only consolidation. When the full retain set is available, as in our classification experiments, the procedure uses retained data only.
- **Analysis.** We characterize the contraction operator through bounds on intervention size and retain-side utility cost, providing local properties of the intervention used by RECAST.
- **Evaluation.** In classification, we compare unlearned models with a retain-only retrained reference using behavioral and internal profiles together with controlled recovery, revealing differences that immediate output-level evaluation does not capture.
- **Generative application.** We adapt RECAST to nudity concept removal in FLUX and evaluate suppression under standard and adversarial prompts together with benign-generation quality and internal concept diagnostics.

2 Related Work

Evaluation beyond outputs. Near-zero forget accuracy can coexist with class-separable intermediate representations (Gao et al., 2026). This motivates comparison with a retain-only retrained reference and direct measurement of residual internal structure (Kim et al., 2026).

Recovery after unlearning. Scissorhands measures relearning time after parameter reset (Wu & Harandi, 2024), Spotter restores forgotten classes from limited class information (Ha et al., 2026), and From Dormant to Deleted relates relearning resistance to parameter distance and linear mode connectivity (Siddiqui et al., 2025). These works distinguish recovery from immediate behavioral forgetting.

Explicit weight interventions. SSD attenuates parameters using forget-side Fisher information (Foster et al., 2024); WIG and Scissorhands reinitialize weights selected by gradient criteria or connection sensitivity (Lee et al., 2026; Wu & Harandi, 2024); and DEL resets and repairs a localized memorized subset (Torkzadehmahani et al., 2025). DASH instead uses direction-aware continuous contraction to preserve adaptability (Shin et al., 2024). RECAST adapts this principle to unlearning using a retain-derived preservation signal.

Objective-based and constrained-update methods. FT uses only retained data, Random Label trains forgotten examples toward incorrect labels (Cadet et al., 2025), and DELETE redesigns distillation for class-centric forgetting (Zhou et al., 2025). SalUn masks updates by forget-gradient saliency (Fan et al., 2024), whereas SEMU confines them to an SVD-derived low-rank subspace (Sendera et al., 2025). Our SalUn and SEMU variants use Random Label, illustrating that a shared objective can still act through different parameter spaces. Appendix C covers additional methods.

Generative concept unlearning and robustness. Text-to-image concept unlearning methods include ESD, UCE, Concept Ablation, MACE, EraseAnything, EraseFlow, and GEM (Gandikota et al., 2023; 2024; Kumari et al., 2023; Lu et al., 2024; Gao et al., 2025; Kusumba et al., 2025;

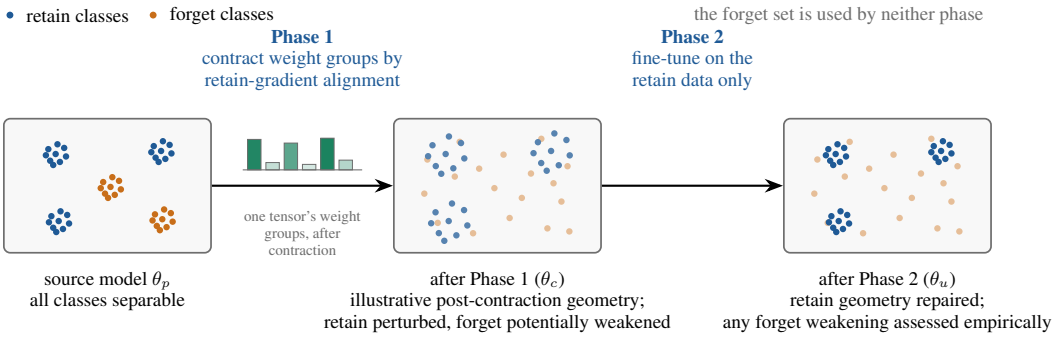

Figure 2: Schematic of RECAST for class-level unlearning. **Phase 1** contracts structural weight groups according to retain-gradient alignment; **Phase 2** repairs retained utility through retain-only consolidation. The feature-space geometry is illustrative rather than a guarantee of information removal. Figure 6 gives the detailed pipeline; the optional low-rank restriction is defined in Appendix B.6.

Grebe et al., 2026). Because standard prompts can miss recoverable concepts, we evaluate FLUX with benign and internal diagnostics, static red-teaming sets, and ReFLUX (Jiang et al., 2026).

3 RECAST: Retain-Guided Weight Contraction

3.1 Preliminaries

We denote the pretrained source, post-contraction, final unlearned, and retain-only retrained models by θ_p, θ_c, θ_u, and θ_r, respectively.

Let $\mathcal{D}_{\text{full}} = \{z_i\}_{i=1}^n$ be the original training set. An unlearning request specifies a forget subset $\mathcal{D}_f \subset \mathcal{D}_{\text{full}}$ and the complementary retain set

$$\mathcal{D}_f \subset \mathcal{D}_{\text{full}}, \qquad \mathcal{D}_r = \mathcal{D}_{\text{full}} \setminus \mathcal{D}_f, \qquad \mathcal{D}_{\text{full}} = \mathcal{D}_r \dot{\cup} \mathcal{D}_f. \tag{1}$$

A class-level request is the special case in which labeled examples $z = (x, y)$ are forgotten whenever y belongs to a designated label subset $\mathcal{Y}_f$; a sample-level request may specify an arbitrary subset of examples. Write θ_p for the parameters obtained by training on $\mathcal{D}_{\text{full}}$ and define the empirical retain risk and retrained reference by

$$\mathcal{L}_r(\theta) = \frac{1}{|\mathcal{D}_r|} \sum_{(x,y)\in\mathcal{D}_r} \ell\big(f_\theta(x), y\big), \qquad \theta_r \in \arg\min_\theta \mathcal{L}_r(\theta), \tag{2}$$

where ℓ is the cross-entropy loss and θ_r is produced by the same architecture, initialization procedure, and training schedule as θ_p, differing only in that $\mathcal{D}_f$ is absent. Given the source model θ_p, unrestricted classification RECAST requires only $\mathcal{D}_r$ during unlearning. Optional restricted variants may additionally use forget-side information to define an admissible update subspace, while the FLUX extension uses target-concept information. The contraction itself is agnostic to whether the forget subset is class- or sample-defined.

3.2 High-level overview of RECAST

RECAST consists of a one-shot retain-guided contraction followed by retain-only consolidation:

$$\theta_c = \mathcal{T}_{\text{RECAST}}(\theta_p; \mathcal{D}_r), \qquad \theta_u = \mathcal{C}_r^{(T)}(\theta_c). \tag{3}$$

$\mathcal{T}_{\text{RECAST}}$ contracts structural weight groups according to retain-gradient alignment, and $\mathcal{C}_r^{(T)}$ performs T steps ($T \geq 1$) on $\mathcal{L}_r$. In the unrestricted classification setting, both stages use $\mathcal{D}_r$ only. Figure 2 summarizes the pipeline.

3.3 Phase 1: Retain-Guided Weight Contraction

Let $\mathcal{M} = \{W_l\}$ denote the tensors transformed by RECAST, each partitioned into non-overlapping structural groups $W_{l,q}$, $q \in \mathcal{G}_l$. Exact group definitions are given in Appendix B.3.

We estimate the retain gradient once at the source checkpoint:

$$G_{r,l} = \frac{1}{|\mathcal{D}_r|} \sum_{(x,y)\in\mathcal{D}_r} \nabla_{W_l} \ell\big(f_{\theta_p}(x), y\big) = \nabla_{W_l} \mathcal{L}_r(\theta_p), \tag{4}$$

The gradient is estimated in one sweep over $\mathcal{D}_r$, with no parameter update.

Writing $w_{l,q} = \mathrm{vec}(W_{l,q})$ and $g_{l,q} = \mathrm{vec}(G_{r,l}[q])$, and following the direction-aware contraction principle of DASH (Shin et al., 2024), RECAST computes each group's alignment with the retain-loss descent direction,

$$c_{l,q} = \cos\big(w_{l,q}, -g_{l,q}\big) \in [-1, 1], \tag{5}$$

and contracts each group by

$$s_{l,q} = \max\big(c_{l,q}, s_{\min}\big) \in [s_{\min}, 1], \qquad W^c_{l,q} = s_{l,q} W_{l,q}, \qquad s_{\min} \in [0, 1]. \tag{6}$$

Higher retain alignment yields weaker or equal contraction, while $s_{\min}$ bounds the maximum contraction applied to any group. Alignment is therefore used as a preservation signal rather than as a measure of forget-specificity. When either group norm is below ε, we set $c_{l,q} = 0$ and hence $s_{l,q} = s_{\min}$.

3.4 Phase 2: Retain-Only Consolidation

Starting from $\theta^{(0)} = \theta_c$, the mandatory second phase runs a finite optimization on the retain risk alone,

$$\theta^{(t+1)} = \mathrm{OPTstep}\big(\theta^{(t)}, \nabla_\theta \mathcal{L}_r(\theta^{(t)}), \eta_t\big), \quad t = 0, \ldots, T-1, \qquad \theta_u = \theta^{(T)}. \tag{7}$$

Consolidation repairs retain behavior disturbed by Phase 1 without computing a forget loss or taking a step on $\mathcal{D}_f$. Because Phase 2 receives only retained examples, any recovery of forgotten behavior can arise indirectly through structure shared with retained data; we therefore measure such recovery empirically rather than assuming it absent.

The complete RECAST procedure is given in Algorithm 1 in Appendix B.3.

The FLUX extension constructs the optional low-rank restriction from target-conditioned gradients.

3.5 Motivation for Direction-Aware Contraction

The contraction rule admits a direct retain-side interpretation. For a group (l, q),

$$g_{l,q}^\top \big(w^c_{l,q} - w_{l,q}\big) = (1 - s_{l,q}) \|g_{l,q}\| \|w_{l,q}\| c_{l,q}. \tag{8}$$

For $c_{l,q} > 0$, contraction can increase the retain objective to first order, whereas for $c_{l,q} \le 0$ its contribution is non-positive. This motivates using alignment as a preservation signal.

Why contract rather than reset the same group? For interventions on the same parameter subvector under the zero-mean reinitialization model, Proposition 2 shows that contraction by a factor s scales the first-order retain effect of hard reinitialization by $1 - s$. Thus, for groups with $c \ge 0$, continuous contraction incurs no larger expected first-order retain cost, and strictly less whenever $c > 0$ and $s > 0$. Corollary 1 further shows that contraction perturbs the source model no more: its squared displacement is no larger than the expected displacement of hard reinitialization.

The alignment rule makes contraction direction-aware, while Corollary 2 bounds the total Phase-1 displacement through the contraction floor. Proposition 3 bounds the retain-side cost under smoothness. These results characterize the cost and geometry of the intervention; they do not establish removal of forgotten information, which is evaluated empirically.

4 Experiments

We first reproduce WIG's CIFAR-10/preactivation ResNet-18 benchmark, then study TinyImageNet-200/ResNet-50 profiles and recovery, and finally test FLUX.1 [dev] concept removal. Detailed protocols are in Appendices D (CIFAR-10), I (TinyImageNet selection), B.7 (recovery), and J (FLUX).

4.1 External Benchmark Reproduction

We reproduce the three-seed CIFAR-10/preactivation ResNet-18 benchmark of Lee et al. (2026) and add internal diagnostics; Appendix D gives the full protocol.

Table 1: CIFAR-10/preactivation ResNet-18 reproduction with class 5 forgotten. Parentheses are absolute gaps to Retrained. Avg. Gap is the reproduced WIG behavioral aggregate, distinct from BP-AG. MIA subscripts denote loss, entropy, and confidence; F/R and tr/te denote forget/retain and train/test. SHs uses the corrected BP and released IP implementations.

	Behavioral Profile (BP)							Internal Profile (IP)		
Method	MIA_L	MIA_E	MIA_C	TFA	TRA	VRA	Avg. Gap↓	$\mathrm{NC3}_F^{\cos}$	NCC_F	NCC_R
Pretrained	70.50	99.75	0.01	100.00	100.00	84.01	68.70	0.91 ± 0.00	70.20 ± 0.71	83.52 ± 0.35
	(20.73)	(54.06)	(99.99)	(100.00)	(0.00)	(1.36)		(1.00)	(15.17)	(2.59)
Retrained	49.77	45.69	100.00	0.00	100.00	85.37	0.00	-0.09 ± 0.00	55.03 ± 0.80	80.93 ± 0.28
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)		(0.00)	(0.00)	(0.00)
FT	72.33	8.33	88.53	70.41	100.00	85.59	35.45	0.82 ± 0.00	69.13 ± 0.62	83.72 ± 0.24
	(22.56)	(37.36)	(11.47)	(70.41)	(0.00)	(0.22)		(0.91)	(14.10)	(2.79)
NegGrad	65.10	17.47	98.51	5.32	99.08	81.99	12.59	0.76 ± 0.00	66.87 ± 0.40	82.60 ± 0.41
	(15.33)	(28.22)	(1.49)	(5.32)	(0.92)	(3.38)		(0.85)	(11.84)	(1.67)
SalUn	55.93	35.94	100.00	0.00	100.00	86.79	3.98	0.19 ± 0.02	71.63 ± 1.53	82.34 ± 0.43
	(6.16)	(9.75)	(0.00)	(0.00)	(0.00)	(1.42)		(0.28)	(16.60)	(1.41)
SSD	54.47	20.65	100.00	0.00	85.41	73.47	7.44	-0.12 ± 0.03	54.87 ± 1.76	78.19 ± 0.62
	(4.70)	(25.04)	(0.00)	(0.00)	(14.59)	(11.90)		(0.03)	(0.16)	(2.74)
SHs	52.57	51.52	100.00	0.00	99.97	84.19	2.16	0.07 ± 0.30	44.77 ± 13.26	61.42 ± 24.69
	(2.80)	(5.83)	(0.00)	(0.00)	(0.03)	(1.18)		(0.16)	(10.26)	(19.51)
WIG	51.50	44.94	100.00	0.00	100.00	81.49	0.62	0.63 ± 0.03	61.33 ± 1.21	77.01 ± 0.64
	(1.73)	(0.75)	(0.00)	(0.00)	(0.00)	(3.88)		(0.72)	(6.30)	(3.92)
RECAST	50.63	43.18	100.00	0.00	99.97	86.24	0.85	-0.20 ± 0.06	55.83 ± 2.83	80.36 ± 0.53
	(0.86)	(2.51)	(0.00)	(0.00)	(0.03)	(0.87)		(0.11)	(0.80)	(0.57)

Behavioral agreement again does not determine the internal endpoint: WIG and RECAST have similar Average Gap, yet RECAST is closer to Retrained on all three internal diagnostics, while SalUn reaches zero test forget accuracy with forget NCC still near Pretrained.

4.2 Main Multi-Profile Comparison

Our main TinyImageNet-200/ResNet-50 comparison uses a fixed random 20-class forget split. Appendix I details checkpoint admissibility, validation-based behavioral selection, and post-hoc Internal Profile reporting. NCC_F probes whether forgotten classes remain separable in representation space, $\mathrm{NC3}_F$ measures their alignment with the classifier, and $\mathrm{CKA}_{\to R}$ measures similarity to Retrained.

Table 2: TinyImageNet-200/ResNet-50 multi-profile comparison. Validation-only behavioral criteria select hyperparameters and stopping epochs; all entries are recomputed on held-out test data, with Internal Profile metrics post hoc. Unlearning rows are mean ± SD over seeds 2/45/100, relative to Retrained. TFA/TRA are test forget/retain accuracy; MIA subscripts denote loss, entropy, and confidence. BP-AG and IP-AG are mean absolute discrepancies to Retrained (lower is better). *DEL excludes BatchNorm parameters from reinitialization; original DEL resets them.

	Behavioral Profile (BP)						Internal Profile (IP)			
Method	MIA_L	MIA_E	MIA_C	TFA	TRA	BP-AG↓	$\mathrm{NC3}_F$	NCC_F	$\mathrm{CKA}_{\to R}$	IP-AG↓
Pretrained	76.00	99.81	1.12	69.20	67.49	.6999	.0225	.675	.710	.2863
Retrained	49.20	14.71	100.00	.00	67.69	.0000	.4791	.563	1.000	.0000
FT	69.70 ± 1.25	$4.10 \pm .30$	100.00	$.27 \pm .06$	$68.31 \pm .17$	$.0784 \pm .0023$	$.1336 \pm .0018$	$.703 \pm .003$	$.814 \pm .001$	$.2240 \pm .0016$
SalUn	$57.77 \pm .55$	12.38 ± 1.57	100.00	.00	$65.75 \pm .31$	$.0272 \pm .0033$	$.2507 \pm .0052$	$.615 \pm .007$	$.761 \pm .009$	$.1730 \pm .0036$
SEMU	$60.73 \pm .45$	$6.45 \pm .67$	100.00	$.13 \pm .06$	$67.04 \pm .10$	$.0498 \pm .0015$	$.3204 \pm .0044$	$.697 \pm .006$	$.761 \pm .003$	$.1771 \pm .0011$
DELETE	60.77 ± 1.50	$16.57 \pm .56$	100.00	$.70 \pm .10$	$67.78 \pm .18$	$.0353 \pm .0030$	$.2882 \pm .0120$	$.706 \pm .003$	$.786 \pm .003$	$.1826 \pm .0023$
WIG	53.90 ± 1.85	$20.64 \pm .22$	100.00	$.00 \pm .00$	$66.17 \pm .37$	$.0266 \pm .0052$	$.1742 \pm .0024$	$.619 \pm .013$	$.807 \pm .006$	$.1845 \pm .0063$
DEL*	50.73 ± 1.21	14.40 ± 1.06	100.00	$.00 \pm .00$	$66.31 \pm .55$	$.0056 \pm .0032$	$.4831 \pm .0014$	$.686 \pm .006$	$.783 \pm .002$	$\mathbf{.1146 \pm .0022}$
RECAST	49.50 ± 1.20	$14.68 \pm .25$	100.00	$.00 \pm .00$	$68.50 \pm .11$	$\mathbf{.0027 \pm .0011}$	$.2362 \pm .0005$	$.617 \pm .003$	$.823 \pm .004$	$.1579 \pm .0016$

On held-out test data, RECAST has the smallest approximate-method BP-AG (0.0027 ± 0.0011), whereas DEL* has the smallest post-hoc IP-AG (0.1146 ± 0.0022). RECAST, DEL*, and WIG all

Table 3: TinyImageNet-200/ResNet-50 recovery diagnostics. T_p is the first validation-forget epoch reaching $p \in \{0.85, 0.90, 0.95\}$ times fixed seed-2 Pretrained accuracy for three evaluations; entries follow seeds 2/45/100. D_θ covers learnable Conv/Linear and BatchNorm-affine parameters. B_R/B_R^{BN} are direct/BN-recalibrated test-retain LMC barriers (percentage points). Higher T_p means slower recovery.

Method	$T_{85}\uparrow$	$T_{90}\uparrow$	$T_{95}\uparrow$	D_θ (%)	B_R	B_R^{BN}
Retrained	27	33	–	124.65	67.07	62.81
FT	1/ 1 /1	1/ 1 /1	3/ 4 /2	13.60 ± 0.00	0.95 ± 0.03	0.00
SalUn	1/ 1 /1	1/ 1 /1	3/ 1 /2	13.49 ± 0.02	0.73 ± 0.13	0.00
SEMU	1/ 1 /1	1/ 1 /2	3/ 6 /3	6.76 ± 0.06	0.00	0.00
DELETE	1/ 1 /1	1/ 1 /1	3/ 2 /2	4.97 ± 0.37	0.01 ± 0.01	0.00
WIG	3/ 4 /3	8/ 6 /8	15/27/17	170.93 ± 1.57	12.75 ± 0.68	0.21 ± 0.21
DEL	3/ 3 /3	4/ 4 /4	12/ 9 /5	49.69 ± 0.60	28.99 ± 14.78	2.76 ± 2.20
RECAST	**12/11/11**	**16/15/16**	**26/25/23**	60.33 ± 0.01	66.54 ± 0.12	0.00

reach zero test forget accuracy but differ internally, confirming that the profiles are complementary. RECAST also reaches $68.50 \pm 0.11\%$ test-retain accuracy after consolidation.

4.3 Recovery Profile

Recovery tests whether forgetting remains fragile under ordinary training. From a behaviorally selected cohort with unlearning learning rate at most 10^{-3}, each checkpoint trains for 50 epochs on $D_f \cup D_r$ with true labels using SGD (10^{-3} learning rate, 0.9 momentum, 5×10^{-4} weight decay) and no oversampling. Appendices B.7 and B.8 describe the recovery protocol and source-checkpoint selection.

RECAST recovers slowest: mean $T_{85}/T_{90}/T_{95}$ is $11.3/15.7/24.7$ epochs, versus $3.3/7.3/19.7$ for WIG and $3.0/4.0/8.7$ for DEL; the remaining methods recover nearly immediately. Neither displacement nor LMC reproduces this ordering: WIG moves farther than RECAST (170.9% versus 60.3%), while RECAST's direct barrier vanishes after BN recalibration but the Pretrained–Retrained control's does not. The mechanism remains open (Siddiqui et al., 2025).

Sequential requests. As an auxiliary one-seed stress test, we apply five consecutive 20-class deletion requests, each starting from the previously unlearned model. After removing half of TinyImageNet, RECAST maintains zero cumulative forget accuracy and nearly matches Retrained-50% in NCC-within (70.68% versus 70.70%), yet individual forgotten blocks remain substantially separable under within-block NCC; full results are in Appendix H.

4.4 Concept Unlearning in FLUX

We test nudity removal in FLUX.1 [dev]. Contraction is retain-based, but concept remapping uses target supervision; without a retrained oracle, we compare suppression, benign utility, adversarial robustness, and internal diagnostics to the base model. Appendix J.1 gives the intervention and standard generation/suppression protocol.

At $s_{\min} \leq 0.1$, RECAST reduces I2P detections from 593 to 16 while keeping COCO CLIP within 0.15 of Base without degrading FID. At $s_{\min}$=0.001, its combined ASR over 1,557 red-team prompts is 0.06%, versus 5.33% for the $s_{\min}$=1 control, 8.67% for GEM, 22.74% for EraseAnything, and 38.15% for Base; ReFLUX ASR remains 0.00% before and after attack (Appendix J.3 details the adversarial protocol).

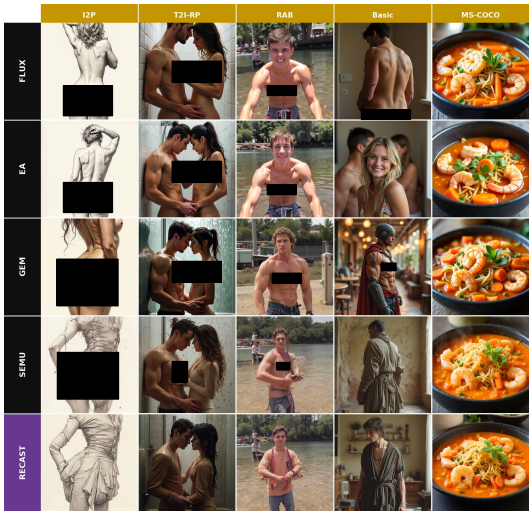

Figure 3: Fixed-seed FLUX.1 [dev] samples. RECAST preserves benign prompts (top) and suppresses nudity across the $s_{\min}$ sweep (bottom); unsafe base panels are censored.

Direct probing shows reduced concept decodability with benign representations largely preserved. A contraction-free control lowers decodability further but drifts more (COCO-retain CKA 0.58 versus ≈ 0.89 for RECAST), so decodability alone is insufficient (Appendix J.2 details the probe and CKA protocol).

5 Conclusion

Similar behavior can mask different internal states and recovery dynamics, so all three profiles are complementary. RECAST is the strongest behavioral match to Retrained, is close on several internal diagnostics, and recovers more slowly than most approximate methods. In FLUX it suppresses the target concept under standard and adversarial prompts while preserving benign quality.

Open directions are to explain the recovery ordering beyond displacement and LMC, combine retain-guided contraction with forget-dependent localization, and develop stronger reference points and adaptive evaluations for generative unlearning without a retrained oracle.

Table 4: FLUX.1 [dev] suppression and benign utility on 10k unfiltered COCO prompts. †Values from Miksa et al. (2026) use a separate evaluation (their Base: 605 detections, FID 21.32).

Method	I2P Tot.↓	CLIP↑	FID↓
Base FLUX	593	30.86	28.34
CA (Model)†	344	29.05	22.66
CA (Noise)†	390	28.73	23.07
ESD†	506	28.44	23.08
UCE†	173	24.56	30.71
MACE†	256	29.52	24.15
EAP†	386	29.86	22.30
Meta-Unlearn†	521	29.91	22.69
EraseAnything†	199	30.24	21.75
SEMU ($s_{\min}$=1)	41	30.85	25.54
RECAST	16	30.76	24.49

Limitations RECAST is an empirical transformation, not a certificate that all influence of $\mathcal{D}_f$ has been removed; our guarantees bound the size and first-order cost of the intervention, not whether it achieves the intended unlearning outcome. Probes, CKA, and relearning attacks cover only selected forms of residual access, and a stronger extraction attack could find structure they miss.

The method depends on the retain sample and transformation scope, and the optional variant additionally requires forget gradients and an SVD. The main TinyImageNet-200 and recovery analyses use one fixed class split with three unlearning seeds, whereas the sequential experiment uses one split and one seed. Recovery uses one subsequent-training protocol with full-model access. SCRUB and Model Sparsity/OMP are not included in the final comparison. FLUX lacks a retrained reference but includes selected linear probes and CKA relative to Base; these diagnostics do not establish the absence of target-concept information throughout the model. Instance-level unlearning and non-visual domains remain open.

Broader Impact and Ethics Statement

Machine unlearning is a compliance technology, and the dominant risk in this area is a false claim of forgetting. A method that reports zero forgotten-class accuracy while retaining a fully decodable representation could be presented to a regulator or a data subject as successful unlearning; our results indicate that several evaluated methods are in this position and recover forgotten behavior within only a few epochs under our protocol. We consider this the paper's most important practical message and a reason to prefer conservative, multi-axis reporting over a single headline number. We also note the dual-use character of class removal: the same operator used to address an unlearning request can suppress a category of content for reasons the affected parties never consented to. RECAST is more resistant to recovery under our tested protocol, which increases both its compliance value and this risk. All experiments use established public research datasets (CIFAR-10, CIFAR-100, Tiny ImageNet, and ImageNet-1K) under their respective terms. The generative experiments use FLUX.1 [dev] and public prompts from I2P and MS-COCO. Potentially unsafe generations are evaluated automatically with NudeNet, censored when shown, and not released. No human subjects, private data, or personally identifying information are involved.

Reproducibility Statement

Section 3 and Algorithm 1 specify the main procedure. Complete algorithms and the optional restriction are in Appendices B.3 and B.6. Classification training, search, and compared-method settings are in Appendices B.4, B.10, B.11, and B.5; validation-based selection is in Appendix I. Appendices B.7 and B.8 document recovery; Appendices D, H, and J document the external reproduction, sequential experiment, and FLUX protocol. Complete proofs are in Appendix E. Configurations and code will accompany the submission as anonymous supplementary material.

AI Use Statement

We used large language models as research and writing assistants. They were used to support literature searches aimed at checking whether specific ideas had prior precedent, to review and correct code, to draft preliminary tests, and to edit technical documentation. We also used LLMs for editorial assistance in writing the manuscript, including improving grammar, spelling, clarity, and concision. All literature claims, code changes, tests, and manuscript edits assisted by LLMs were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

References

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy*, pp. 141–159. IEEE, 2021.

Xavier F. Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep unlearn: Benchmarking machine unlearning for image classification. In *2025 IEEE European Symposium on Security and Privacy*, pp. 939–962. IEEE, 2025.

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. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/40.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: Empowering machine unlearning via gradient-based weight saliency. In *International Conference on Learning Representations*, 2024.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12043–12051, 2024.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2426–2436, October 2023.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5111–5120, January 2024.

Daiheng Gao, Shilin Lu, Wenbo Zhou, Jiaming Chu, Jie Zhang, Mengxi Jia, Bang Zhang, Zhaoxin Fan, and Weiming Zhang. Eraseanything: Enabling concept erasure in rectified flow transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18470–18494. PMLR, 2025.

Yichen Gao, Altay Unal, Akshay Rangamani, and Zhihui Zhu. An illusion of unlearning? assessing machine unlearning through internal representations. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4114–4122. PMLR, 2026.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jonas Henry Grebe, Tobias Braun, Anna Rohrbach, and Marcus Rohrbach. GEM: Geometric erasure by contrastive velocity matching in rectified flows. *arXiv preprint arXiv:2606.00140*, 2026.

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*, 2026. arXiv:2506.01318.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Nanxiang Jiang, Zhaoxin Fan, Enhan Kang, Daiheng Gao, Yun Zhou, Yanxia Chang, Zheng Zhu, Yeying Jin, and Wenjun Wu. Erased, but not forgotten: Erased rectified flow transformers still remain unsafe under concept attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 8080–8089, June 2026.

Yongwoo Kim, Sungmin Cha, and Donghyun Kim. Are we truly forgetting? a critical re-examination of machine unlearning evaluation protocols. *Engineering Applications of Artificial Intelligence*, 167:113785, 2026. doi: 10.1016/j.engappai.2026.113785.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22691–22702, October 2023.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Abhiram Kusumba, Maitreya Patel, Kyle Min, Changhoon Kim, Chitta Baral, and Yezhou Yang. Eraseflow: Learning concept erasure policies via GFlowNet-driven alignment. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Doun Lee, Jongyun Shin, Jinwoo Bae, Hyunjoon Cho, and Jangho Kim. Weight initialization based on gradient similarity for versatile machine unlearning. *Proceedings on Privacy Enhancing Technologies*, 2026(3):575–603, 2026. doi: 10.56553/popets-2026-0097.

Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. MACE: Mass concept erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6430–6440, June 2024. doi: 10.1109/CVPR52733.2024.00615.

Jan Miksa, Patryk Krukowski, Przemyslaw Spurek, Dawid Damian Rymarczyk, and Marcin Sendera. Barrier: Bounded activation regions for robust information erasure. *arXiv preprint arXiv:2605.15737*, 2026.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117.

Marcin Sendera, Lukasz Struski, Kamil Ksiazek, Kryspin Musiol, Jacek Tabor, and Dawid Damian Rymarczyk. SEMU: Singular value decomposition for efficient machine unlearning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53843–53866. PMLR, 2025.

Baekrok Shin, Junsoo Oh, Hanseul Cho, and Chulhee Yun. DASH: Warm-starting neural network training in stationary settings without loss of plasticity. In *Advances in Neural Information Processing Systems*, 2024.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy*, pp. 3–18. IEEE, 2017.

Shoaib Ahmed Siddiqui, Adrian Weller, David Krueger, Gintare Karolina Dziugaite, Michael Curtis Mozer, and Eleni Triantafillou. From dormant to deleted: Tamper-resistant unlearning through weight-space regularization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4308.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium*, pp. 2615–2632, 2021.

Reihaneh Torkzadehmahani, Reza Nasirigerdeh, Georgios Kaissis, Daniel Rueckert, Gintare Karolina Dziugaite, and Eleni Triantafillou. Improved localized machine unlearning through the lens of memorization. *Transactions on Machine Learning Research*, 2025.

Jing Wu and Mehrtash Harandi. Scissorhands: Scrub data influence via connection sensitivity in networks. In *Computer Vision – ECCV 2024*, volume 15105 of *Lecture Notes in Computer Science*, pp. 367–384. Springer, 2024. doi: 10.1007/978-3-031-72970-6_21.

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*, 2025.

A One-Module Diagnostic

The motivating random-sample diagnostic uses ResNet-18 on CIFAR-100 with LGA. In Phase 1 all parameters are frozen except one convolutional or classifier weight tensor, which is optimized for ten epochs. Phase 2 unfreezes all convolutional and classifier weights for a further ten epochs; biases and BatchNorm parameters remain frozen. The all-weights control omits Phase 1.

Table 5: One-module diagnostic. Accuracy in percent; P1 and P2 denote the constrained and all-weights phases.

Protocol	Retain data	P1 F	P1 R	P1 Test	P2 F	P2 R	P2 Test
one-module	yes	13.33	99.98	75.40	0.22	99.98	75.68
one-module	no	8.00	99.93	74.26	0.22	99.71	72.88
all-weights	yes	–	–	–	0.22	99.98	75.67
all-weights	no	–	–	–	0.00	98.69	71.32

Table 5 reports the representative `layer4.0.conv1` intervention. Late-module choices preserve utility: without retain data, `layer4.1.conv2` retains 99.98% training-retain and 75.65% test accuracy, and `fc` retains 99.91% and 74.42%. By contrast, selecting `layer1.0.conv1` reduces these to 19.88% and 17.29%, although all three choices reach nearly zero Phase-2 forget accuracy. For `layer4.0.conv1`, the selected module's Phase-2 displacement is $5.78\times$ its displacement in the all-weights control, whereas every non-selected module has a ratio below 0.4. A forget objective can therefore be satisfied through a localized decision-path change.

Table 18: Results of the random data forgetting scenario for the ResNet-18 model on the CIFAR-100 dataset. "()" denotes the performance gap against the Retrain.

Method	*MIA loss*	*MIA entropy*	*MIA confidence*	D^F *Acc*	*Avg Gap*	DR *Acc*	D^{Te} *Acc*
Retrain	50.14±0.49	53.02±1.06	46.16±1.21	71.93±0.55	0.00	99.95±0.01	72.17±0.26
Finetune	69.98±0.45(19.8)	85.85±0.36(32.8)	13.48±0.66(32.7)	99.87±0.03(27.9)	28.32	99.99±0.00	73.57±0.24
GA	65.43±0.49(15.3)	89.22±1.88(36.2)	8.15±0.22(38.0)	97.62±0.25(25.7)	28.80	99.24±0.14	71.12±0.02
NegGrad	69.80±0.40(19.7)	86.15±0.52(33.1)	13.44±0.95(32.7)	99.83±0.07(27.9)	28.35	**99.98±0.01**	**73.62±0.31**
CFK	69.59±0.38(19.5)	88.53±0.22(35.5)	10.78±0.41(35.4)	99.84±0.08(27.9)	29.56	**99.98±0.00**	73.04±0.25
EUK	57.13±0.60(7.0)	58.87±0.86(5.8)	**41.87±1.07**(4.3)	83.88±0.53(11.9)	7.27	99.97±0.00	68.88±0.45
Bad-T	64.77±1.04(14.6)	65.45±1.64(12.4)	16.54±0.66(29.6)	99.45±0.08(27.5)	21.05	99.87±0.01	69.41±0.35
SCRUB+R	57.65±0.28(7.5)	48.60±0.57(4.4)	34.53±0.18(11.6)	84.13±2.23(12.2)	8.94	96.56±1.96	64.04±1.23
SSD	64.41±0.59(14.3)	86.67±4.60(33.6)	8.57±4.09(37.6)	95.46±2.83(23.5)	27.26	95.65±2.90	69.50±2.09
SalUn	56.31±0.52(6.2)	48.36±3.57(4.7)	39.67±3.53(6.5)	89.77±0.85(17.8)	8.79	99.78±0.13	71.35±0.21
WIG	**55.48±0.38**(5.3)	**55.14±1.52**(2.1)	41.60±0.43(4.6)	**75.23±0.14**(3.3)	**3.83**	99.12±0.16	63.43±0.68

Figure 4: Published random-data forgetting results for CIFAR-100 with ResNet-18 (Table 18 of Lee et al. (2026)), included as context for the random-sample diagnostic. Parentheses report the performance gap relative to the Retrain reference, following the source table.

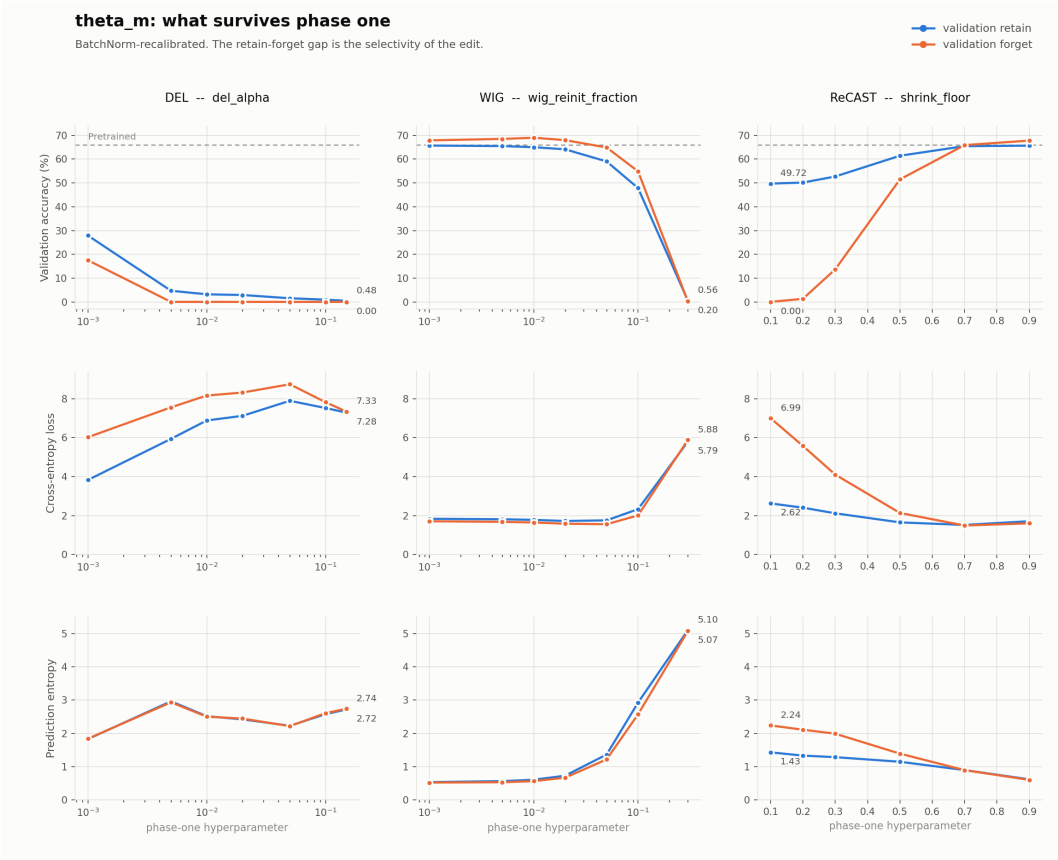

Figure 5: Phase-1 diagnostic sweeps for DEL, WIG, and RECAST after BatchNorm recalibration and before repair. Rows report validation retain and forget accuracy, cross-entropy loss, and prediction entropy. The dashed line marks pretrained validation accuracy; separation between the retain and forget curves visualizes the immediate selectivity of the intervention rather than the full unlearning endpoint.

The RECAST floor sweep provides an empirical sanity check that $s_{\min}$ controls the strength of the Phase-1 intervention, complementing the ordered parameter-space path characterized in Proposi-

tion 1. This does not imply monotonic forgetting or retain accuracy and is not a substitute for the alignment-vs-flat ablation.

B Implementation and Protocol Details

B.1 Data Splits

The CIFAR-100 diagnostic uses a fixed dataset-seed-2 random-sample forget manifest. The TinyImageNet-200 random-20 class-unlearning campaign also uses dataset seed 2. The Tiny ImageNet animal-far split assigns 40 animal classes to C, the remaining 19 animals and one nearest buffer class to B, and 140 non-animal classes to A; it is constructed from WordNet rather than from model features or test performance.

B.2 Classification Settings and Evaluation Definitions

The main classification campaign uses TinyImageNet-200 with ResNet-50 and a fixed random 20-class forget split. Its selected rows come from method-specific configuration and checkpoint sweeps rather than a shared final epoch; Appendix I specifies qualification, behavioral selection, and post-hoc internal reporting. The external CIFAR-10/preactivation ResNet-18 experiment follows the released WIG benchmark instead (Appendix D). CIFAR-100 is used only for the random-sample diagnostic in Appendix A.

The Behavioral Profile contains quantities available from model outputs. In the TinyImageNet comparison these are validation forget and retain accuracy and the loss-, entropy-, and confidence-based membership-inference outcomes (MIA-L/E/C). The membership audit follows the prediction-statistic formulation of prior image-classification unlearning evaluations (Shokri et al., 2017; Song & Mittal, 2021; Jia et al., 2023; Fan et al., 2024). Labels and membership annotations are used where required; "behavioral" refers to the absence of access to hidden representations, not to the absence of auxiliary evaluation data.

The Internal Profile characterizes the same selected checkpoint using forget-side nearest-class-center accuracy (NCC-F), the NC3 classifier–class-mean alignment statistic (Papyan et al., 2020), and linear centered kernel alignment with Retrained (Kornblith et al., 2019). NCC uses labeled prototypes and is therefore a representation diagnostic rather than an operational classifier. All metrics are interpreted relative to Retrained: an extremal raw value is not automatically preferable if it moves farther from the retain-only reference. Neither profile alone certifies complete removal of training influence.

B.3 RECAST Algorithms and Implementation

Algorithms 2 and 3 give the two phases of RECAST in their general form. Unrestricted RECAST sets the restriction mode to NONE, so its inputs exclude $\mathcal{D}_f$ and both phases use retained examples only; this specialization is exactly Algorithm 1 below. The optional FORGET mode uses forget gradients solely to construct an admissible low-rank space and does not change the retain-derived contraction strengths.

The modes differ as follows. With $m = \text{NONE}$, RECAST applies the raw retain-guided contraction and then performs ordinary retain-only consolidation with no SVD construction and no projection. With $m = \text{FORGET}$, the Phase-1 contraction displacement of each restricted tensor is projected into a subspace estimated exclusively from forget gradients, and Phase 2 optimizes coordinates of that subspace through a SEMU-style adapter. Unrestricted RECAST is therefore a complete algorithmic branch, not a restricted method with an additional approximation.

Algorithm 1 RECAST ($m = \text{NONE}$)

Require: source parameters θ_p; retain set $\mathcal{D}_r$; contraction tensors $\mathcal{M}$ with fixed group partitions $\{\mathcal{G}_l\}$; floor $s_{\min}$; consolidation steps T; optimizer OPT; schedule $\{\eta_t\}$

// the forget set $\mathcal{D}_f$ is not an input

$G_{r,l} \leftarrow \nabla_{W_l} \mathcal{L}_r(\theta_p)$ for each $W_l \in \mathcal{M}$ ▷ Eq. 4; one sweep, no update

$\theta \leftarrow \theta_p$

for all $W_l \in \mathcal{M}$ **do**

  for all $q \in \mathcal{G}_l$ **do**

    $w \leftarrow \text{vec}(W_{l,q}), \quad g \leftarrow \text{vec}(G_{r,l}[q])$

    $s_{l,q} \leftarrow \max\big(\cos(w, -g),\ s_{\min}\big)$ ▷ $s_{l,q} \leftarrow s_{\min}$ if $\|w\|\|g\| < \varepsilon$

    $W_{l,q} \leftarrow s_{l,q} W_{l,q}$

  end for

end for

$\theta_c \leftarrow \theta$ ▷ Phase-1 diagnostic checkpoint only

initialize OPT on the parameters designated for repair

for $t = 0, \ldots, T-1$ **do**

  sample retain mini-batch $b_r \subset \mathcal{D}_r$

  $\theta \leftarrow \text{OPTstep}\big(\theta, \nabla_\theta \mathcal{L}_r(b_r; \theta), \eta_t\big)$ ▷ retain risk only

end for

return $\theta_u \leftarrow \theta$

For every tensor $W_l \in \mathcal{M}_{\text{res}}$ with shape $d_{l,1} \times \cdots \times d_{l,k}$, we fix an architecture-specific matricization

$$\text{mat}_l : \mathbb{R}^{d_{l,1} \times \cdots \times d_{l,k}} \to \mathbb{R}^{m_l \times n_l}. \tag{9}$$

The map may reshape and permute axes, but it is fixed before unlearning and used consistently in both phases. The construction therefore applies to any selected parameter tensor admitting a prescribed matrix view, including tensors from linear, convolutional, attention, MLP, mixing, or other modules.

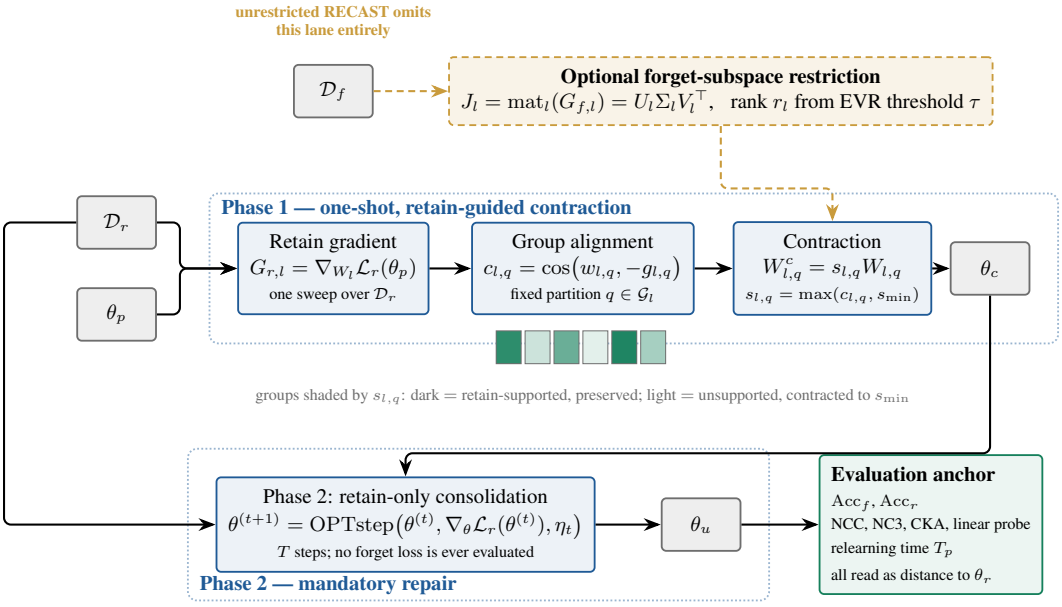

Figure 6: Detailed RECAST pipeline, expanding Figure 2. A single pass over the retain set yields a per-tensor gradient $G_{r,l}$; each structural group is contracted by its own cosine alignment with the retain descent direction, floored at $s_{\min}$. Retain-only consolidation repairs the collateral utility damage. The forget set enters only through the optional restriction lane (dashed), which unrestricted RECAST does not use; in restricted RECAST the same subspace also constrains every Phase-2 update (Proposition 5). The retrained reference θ_r is an evaluation target and never an input.

The retain-gradient estimator is Equation 4. RECAST inherits the grouping and direction-aware contraction rule from DASH (Shin et al., 2024) but replaces DASH's temporal exponential average over sequential data chunks with a static retain-set estimator tailored to the unlearning setting. Its example-weighted normalization prevents a short final mini-batch from receiving a full batch's influence.

Algorithm 2 RECAST Phase 1: retain-guided contraction with optional restriction

Require: source parameters θ_p; retain set $\mathcal{D}_r$; contraction tensors $\mathcal{M}$; structurally grouped indices $\{\mathcal{G}_l\}$; floor $s_{\min}$; restriction mode $m \in \{\text{NONE}, \text{FORGET}\}$; optional forget set $\mathcal{D}_f$, restricted parameter subset $\mathcal{M}_{\text{res}} \subseteq \mathcal{M}$, and EVR threshold τ

$\theta \leftarrow \theta_p$
$G_{r,l} \leftarrow \nabla_{W_l} \mathcal{L}_r(\theta_p)$ for each $W_l \in \mathcal{M}$ using Equation 4
if $m \neq$ NONE **then**
    $G_{f,l} \leftarrow \nabla_{W_l} \mathcal{L}_f(\theta_p)$ for each $W_l \in \mathcal{M}_{\text{res}}$
end if
$\mathcal{S} \leftarrow \emptyset$
for all $W_l \in \mathcal{M}$ **do**
    for all $q \in \mathcal{G}_l$ **do**
        $w \leftarrow \text{vec}(W_{l,q})$, $g \leftarrow \text{vec}(G_{r,l}[q])$
        $c \leftarrow \cos(w, -g)$
        $s \leftarrow \max(c, s_{\min})$
        $\Delta_{l,q} \leftarrow (s-1)W_{l,q}$
    end for
    assemble Δ_l from the non-overlapping group updates $\{\Delta_{l,q} : q \in \mathcal{G}_l\}$
    if $m =$ NONE or $W_l \notin \mathcal{M}_{\text{res}}$ **then**
        keep the raw contraction displacement Δ_l
    else
        $J_l \leftarrow \text{mat}_l(G_{f,l})$
        $U_l, \Sigma_l, V_l^\top \leftarrow \text{SVD}(J_l)$
        $r_l \leftarrow \min\{r : \text{EVR}(\Sigma_l, r) \geq \tau\}$
        $\Delta_l \leftarrow \text{mat}_l^{-1}\big(U_{l,r_l} U_{l,r_l}^\top \text{mat}_l(\Delta_l) V_{l,r_l} V_{l,r_l}^\top\big)$
    end if
    $W_l^c \leftarrow W_l + \Delta_l$; $W_l \leftarrow W_l^c$
    if $W_l \in \mathcal{M}_{\text{res}}$ and $m \neq$ NONE **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(l, W_l^c, U_{l,r_l}, V_{l,r_l})\}$
    end if
end for
$\theta_c \leftarrow \theta$
return contracted checkpoint θ_c and restriction state $\mathcal{S}$

The contraction set $\mathcal{M}$ and the restriction set $\mathcal{M}_{\text{res}}$ are distinct. The former specifies which weight tensors Phase 1 contracts; the latter contains the tensors whose Phase-1 change and Phase-2 displacement are confined to a forget-gradient SVD subspace. The definition is architecture independent; task-specific exclusions such as the classification head are specified below.

Algorithm 3 RECAST Phase 2: retain-only consolidation with SEMU-style adapters

Require: contracted checkpoint θ_c; retain set $\mathcal{D}_r$; restriction mode m; restriction state $\mathcal{S}$; full-rank repair parameters Θ_{free}; state-update policy; consolidation steps $T \geq 1$; optimizer OPT; schedule $\{\eta_t\}$

$\mathcal{A} \leftarrow \emptyset$
if $m = \text{FORGET}$ **then**
  for all $(l, W_l^c, U_{l,r_l}, V_{l,r_l}) \in \mathcal{S}$ **do**
    $A_l^{(0)} \leftarrow U_{l,r_l}^\top \operatorname{mat}_l(W_l^c) V_{l,r_l}$
    $B_l \leftarrow \operatorname{mat}_l(W_l^c) - U_{l,r_l} A_l^{(0)} V_{l,r_l}^\top$
    freeze B_l, U_{l,r_l}, and V_{l,r_l}; initialize trainable $A_l \leftarrow A_l^{(0)}$
    parameterize $W_l(A_l) \leftarrow \operatorname{mat}_l^{-1}\big(B_l + U_{l,r_l} A_l V_{l,r_l}^\top\big)$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{A_l\}$
  end for
  $\Xi \leftarrow \Theta_{\text{free}} \cup \mathcal{A}$
else
  $\Xi \leftarrow \Theta_{\text{free}}$ ▷ all designated Phase 2 tensors are optimized directly
end if
apply the task-specific state-update policy using retain data
initialize OPT on the trainable variables Ξ
for $t = 0, \ldots, T-1$ **do**
  sample retain mini-batch $b_r \subset \mathcal{D}_r$
  materialize restricted tensors $W_l(A_l)$ when $m = \text{FORGET}$
  $\Xi \leftarrow \text{OPTstep}\big(\Xi, \nabla_\Xi \mathcal{L}_r(b_r; \theta(\Xi)), \eta_t\big)$
end for
materialize and return the complete RECAST model $\theta_u = \theta(\Xi)$ and diagnostic checkpoint θ_c

With $m = \text{NONE}$ we have $\mathcal{S} = \emptyset$ and Θ_{free} denotes the complete set of parameters selected for ordinary Phase-2 optimization. With $m = \text{FORGET}$, tensors in $\mathcal{M}_{\text{res}}$ are represented only through their trainable cores $\mathcal{A}$, while Θ_{free} contains all remaining parameters selected for full-rank repair. No full restricted tensor is passed to the optimizer and no projection is performed after an optimizer step. Phase 2 is mandatory in both modes and always minimizes only the retain risk.

The group map $\mathcal{G}_l$ is fixed by the structure of W_l rather than selected from data. It partitions the tensor into non-overlapping blocks whose union is the full tensor; the particular blocks are architecture-specific—for example neuron-wise vectors, spatial kernels, rows or columns of attention projections, MLP projection groups, or blocks of a mixing operator. RECAST requires only that the partition be specified unambiguously and used consistently when slicing both W_l and $G_{r,l}$. The same scalar $s_{l,q}$ is broadcast to every coordinate in group q, and Phase 1 modifies only tensors included in $\mathcal{M}$.

Classification-specific parameter policy. For classification experiments we use the disjoint conceptual partition

$$\theta = \theta_{\text{res}} \,\dot\cup\, \theta_{\text{head}} \,\dot\cup\, \theta_{\text{free}}. \tag{10}$$

Here θ_{res} contains the selected feature-extractor tensors in $\mathcal{M}_{\text{res}}$ and is optimized through adapter cores. The complete classification head, weights and biases alike, belongs to θ_{head}: it participates in Phase-1 contraction, is excluded from $\mathcal{M}_{\text{res}}$, and is optimized freely in Phase 2. The set θ_{free} contains the remaining explicitly unfrozen full-rank parameters, including direct biases and affine normalization parameters. In Algorithm 3 the classification-specific full-rank input is therefore $\Theta_{\text{free}} = \theta_{\text{head}} \cup \theta_{\text{free}}$. Stateful normalization statistics are updated using retain mini-batches; specifically, BatchNorm running means and variances remain adaptive during consolidation. This policy lets the classifier rotate, rescale, and translate its discriminative axes to align with the contracted representation—a freely adapting head cannot by itself establish representation-level unlearning, which is why our conclusions are anchored by head-independent diagnostics.

Numerical details. Gradients are accumulated without changing model parameters, and the loss is normalized per example before aggregation. We set the alignment to zero when either group norm falls numerically below ε, which maps such a group to $s_{\min}$. Taking the maximum with $s_{\min} \geq 0$ prevents sign reversal, while the upper bound of cosine similarity prevents weight expansion. The

displacement introduced for the algorithmic description is

$$\Delta_{l,q} = W^c_{l,q} - W_{l,q} = (s_{l,q} - 1)W_{l,q}. \tag{11}$$

Each run stores $G_{r,l}$ summaries, the distribution of $s_{l,q}$, and the normalized intervention size

$$d_l = \frac{\|W^c_l - W_l\|_F}{\|W_l\|_F + \varepsilon}. \tag{12}$$

The mean-matched flat-contraction ablation replaces all group-wise values in layer l by $\bar{s}_l = |\mathcal{G}_l|^{-1} \sum_q s_{l,q}$, preserving the average contraction while removing direction awareness.

B.4 Reference and Auxiliary Training Schedules

Table 6: Auxiliary fixed-budget optimization schedules. The 200-epoch reference row applies only when an auxiliary experiment explicitly trains a new reference, including Retrained-50%; it does not describe the fixed main TinyImageNet reference pair. Warm-up is linear to η_{base}.

Stage	Epochs	Batch	Warm-up	Initial LR	SGD schedule	AdamW schedule	Final LR
Auxiliary reference training	200	128	5 epochs	$10^{-3}\eta_{\text{base}}$	milestones 90/140/180	cosine	$10^{-3}\eta_{\text{base}}$
Unlearning	20	128	2 epochs	$10^{-2}\eta_{\text{base}}$	milestones 14/18	cosine	$10^{-2}\eta_{\text{base}}$

The active main TinyImageNet experiments use one fixed Pretrained checkpoint and one fixed retain-only Retrained checkpoint, each trained once with seed 2; the reporting seeds 2, 45, and 100 vary only the unlearning runs. The generic 200-epoch schedule above is retained solely for auxiliary experiments that train a new reference, such as Retrained-50%. It uses batch size 128, no early stopping, a five-epoch warm-up from $10^{-3}\eta_{\text{base}}$, and either SGD milestones 90/140/180 (factor 0.1) or AdamW cosine decay to $10^{-3}\eta_{\text{base}}$.

In auxiliary fixed-budget and ablation experiments, iterative unlearning runs for 20 epochs with batch size 128 and no early stopping. Its two-epoch linear warm-up starts at $10^{-2}\eta_{\text{base}}$. SGD milestones occur at epochs 14 and 18 with decay factor 0.1; AdamW follows cosine decay to $10^{-2}\eta_{\text{base}}$. Non-iterative transformations such as SSD or the pruning/reset stage of a weight-centric baseline are applied once; when a method prescribes subsequent repair, consolidation, or fine-tuning, that stage uses the same auxiliary 20-epoch budget. The main TinyImageNet comparison instead selects from method-specific sweeps with different checkpoint grids; its exact candidate records are documented in Appendix I.

B.5 Compared Methods

Table 7 groups the compared methods by primary mechanism and Table 8 places them on the two axes used in Section 2.

Table 7: Compared methods by primary mechanism.

Primary mechanism	Methods and variants
Output/objective	ROFT; LGA; cyclic Random Label (RL)
Features/distillation	DELETE; SCRUB; Spotter
Restricted updates	UNSC; SalUn; SalUn+LGA; SEMU+RL; SEMU+LGA
Weight modification	SSD; Model Sparsity/OMP; **RECAST**

Table 8: Positioning of class-unlearning methods along the signal used to decide what to modify or preserve and the form of the intervention.

	Iterative objective	**One-shot weight map (+ repair)**
Forget-dependent signal	LGA, RL, SCRUB, DELETE, Spotter, SalUn, SEMU, UNSC	SSD, RECAST (optional variant) WIG: partial reinitialization + repair Scissorhands: connection-sensitivity selection; reinitialization + projection-based repair
No signal / generic	—	Model Sparsity (OMP)
Retain signal	ROFT	**RECAST**: continuous contraction + retain-only repair

We describe the implemented variants rather than relying only on method names, because several baselines combine a published parameter restriction with a different unlearning objective. Unless stated otherwise, methods start from the same pretrained checkpoint and retain the original classification head.

Pretrained and retrained references. The pretrained model is the source checkpoint before unlearning. The retrained reference uses the same architecture but is trained from scratch without the forget classes. It defines the target for both output and internal diagnostics and is not counted as an efficient unlearning method, following the standard separation between exact retraining and approximate unlearning (Ginart et al., 2019; Bourtoule et al., 2021).

Retain-only fine-tuning (ROFT). ROFT minimizes cross-entropy only on $\mathcal{D}_r$ for 20 epochs and never uses forgotten examples, relabeled or otherwise. It is our exposure control: it measures the change caused by additional retain training without an explicit forgetting mechanism. Fine-tuning is a common baseline family in image-classification unlearning benchmarks (Cadet et al., 2025), but our acronym denotes this specific retain-only implementation.

Loss gradient ascent (LGA). LGA maximizes the true-label cross-entropy on $\mathcal{D}_f$, pushing the classifier away from correct forget predictions. The main comparison pairs these ascent updates with ordinary retain cross-entropy to limit utility loss; the one-module diagnostic additionally reports a no-retain variant. Gradient ascent is included as the classical objective-level baseline (Cadet et al., 2025).

Cyclic random labeling (RL). RL replaces each forget target with an incorrect randomly selected class and minimizes cross-entropy toward that target. Our cyclic schedule alternates relabeled forget updates with retain updates rather than optimizing one fixed joint dataset (Cadet et al., 2025).

DELETE teacher. DELETE uses decoupled distillation to construct a modified distillation target while preserving non-target knowledge (Zhou et al., 2025). Our teacher variant freezes the source model, masks the true-class component of its output on a forget example, and distills the student toward the remaining distribution. It is optimized on forget examples only; the self-distillation alternative is not included.

UNSC projected. UNSC calibrates an unlearning update in a null space derived from retained representations (Chen et al., 2024). We use a frozen teacher to assign pseudo-labels to forget examples, estimate the retain representation basis, and project the forget gradients using the published null-space construction. Direct retain-set training is disabled in this variant and PCGrad alternatives are not included.

SalUn and SalUn+LGA. SalUn ranks parameters by forget-gradient saliency and updates only a selected weight mask (Fan et al., 2024). Our SalUn baseline fixes a global 50% mask and applies the paper's random-label objective together with retain cross-entropy. SalUn+LGA keeps exactly the same mask but replaces the random-label loss with true-label LGA; it is an experimental objective substitution, not the canonical SalUn algorithm.

SEMU+RL and SEMU+LGA. SEMU uses SVD of forget-gradient matrices to obtain a low-dimensional trainable parameterization (Sendera et al., 2025). SEMU+RL uses this parameterization with fixed random-label cross-entropy; SEMU+LGA retains the same parameterization and schedule but substitutes true-label gradient ascent. We report the latter as an explicit experimental variant so that the effect of the restriction can be compared under a stronger forget objective.

Selective Synaptic Dampening (SSD). SSD compares parameter-wise Fisher information on the forget data with the full training data, identifies parameters disproportionately important for the forget set, and dampens them without iterative retraining (Foster et al., 2024). Our run uses $\alpha = 0.4$ and $\lambda = 3.0$. Only one completed seed is currently available, so SSD is reported without a standard deviation.

Model Sparsity / one-shot magnitude pruning (OMP). Model Sparsity introduces a "prune first, then unlearn" paradigm in which a sparse model prior improves a downstream approximate unlearner (Jia et al., 2023). We obtain the sparse checkpoint by one-shot magnitude pruning—the globally smallest-magnitude weights are zeroed once to reach a prescribed sparsity and the binary mask is fixed thereafter. OMP is a weight-space preconditioner rather than a complete forget objective and is paired with the same downstream objective and 20-epoch budget as its dense control.

SCRUB. SCRUB treats the original model as a frozen teacher and initializes a student from the same parameters (Kurmanji et al., 2023). Its alternating max–min procedure maximizes teacher–student divergence on forget examples while minimizing distillation and supervised losses on retained examples, so it can change both the decision rule and the feature extractor without explicitly resetting weights. We use the canonical variant; checkpoint rewinding is reported separately if enabled rather than folded into the same method name.

Spotter. Spotter is a plug-in objective targeting two failure modes of class unlearning: collateral damage near the forget-class boundary and rapid prototype-based relearning (Ha et al., 2026). It combines masked knowledge distillation on boundary-proximal adversarial examples with an intra-class dispersion loss that scatters forget-class embeddings. It is thus most directly related to our recovery and representation analyses; unlike RECAST it continuously optimizes a forget-dependent objective.

RECAST. RECAST is the proposed method of Section 3. Unrestricted RECAST estimates alignment only from $\mathcal{D}_r$, transforms the selected weights, and performs retain-only consolidation; it never evaluates a forget loss. Forget-gradient restriction, and RECAST followed by LGA or PL, are reserved for ablation and are not included in the RECAST row.

WIG and Scissorhands are RECAST's closest experimental comparators in the external benchmark (Section 4.1). SalUn and SEMU instead restrict where an objective may update the model, while Spotter addresses forget-class feature geometry and relearning through optimized losses.

B.6 Optional SVD Restriction and Free Repair Parameters

The optional variant confines the Phase-1 displacement and every Phase-2 update of a selected tensor subset $\mathcal{M}_{\mathrm{res}} \subseteq \mathcal{M}$ to a low-rank space spanned by the leading singular directions of the forget gradient. During Phase 2, the restriction is enforced by parameterization rather than by post-step projection.

For each $W_l \in \mathcal{M}_{\mathrm{res}}$ the restriction uses the fixed map mat_l from Equation 9. The definition may depend on the architecture and parameter type, but the same map is used when constructing the bases, restricting the Phase-1 contraction, and materializing the Phase-2 adapter. Forget gradients use the same example weighting as Equation 4. The restriction source is

$$J_l = \mathrm{mat}_l(G_{f,l}), \qquad m = \textsc{Forget}, \tag{13}$$

and with $J_l = U_l \Sigma_l V_l^\top$ we select the smallest rank satisfying the explained-variance-ratio threshold

$$r_l = \min \left\{ r : \frac{\sum_{i=1}^{r} \sigma_{l,i}^2}{\sum_j \sigma_{l,j}^2} \geq \tau \right\}. \tag{14}$$

The retained bases $U_{l,r_l} \in \mathbb{R}^{m_l \times r_l}$ and $V_{l,r_l}^\top \in \mathbb{R}^{r_l \times n_l}$ define the admissible bilinear subspace, with projectors $P_l^L = U_{l,r_l} U_{l,r_l}^\top$ and $P_l^R = V_{l,r_l} V_{l,r_l}^\top$.

Restricted Phase-1 contraction. Let W_l^0 be the source tensor and S_l collect the broadcast retain-derived contraction factors. The unrestricted proposal and its displacement are

$$\widetilde{W}_l^c = S_l \odot W_l^0, \qquad \Delta_l^{\text{contract}} = \widetilde{W}_l^c - W_l^0, \tag{15}$$

and for $W_l \in \mathcal{M}_{\text{res}}$ Phase 1 admits only the component inside the forget-gradient subspace:

$$\text{mat}_l(\Delta_l^{\text{proj}}) = P_l^L \,\text{mat}_l(\Delta_l^{\text{contract}}) P_l^R, \qquad W_l^c = W_l^0 + \Delta_l^{\text{proj}}. \tag{16}$$

Restriction therefore changes where contraction may occur, not how its strength is computed. A tensor that is contracted but not restricted receives the full proposal $W_j^c = \widetilde{W}_j^c$.

SEMU-style Phase-2 parameterization. Restricted consolidation does not optimize the full W_l and does not project a full-weight optimizer step. For every $W_l \in \mathcal{M}_{\text{res}}$ we initialize

$$A_l^{(0)} = U_{l,r_l}^\top \,\text{mat}_l(W_l^c) V_{l,r_l}, \tag{17}$$

$$B_l = \text{mat}_l(W_l^c) - U_{l,r_l} A_l^{(0)} V_{l,r_l}^\top. \tag{18}$$

The bases and the anchor are frozen; only the core $A_l^{(t)} \in \mathbb{R}^{r_l \times r_l}$ is trainable, and the tensor used by the network is

$$W_l^{(t)} = \text{mat}_l^{-1}\left(B_l + U_{l,r_l} A_l^{(t)} V_{l,r_l}^\top\right). \tag{19}$$

Proposition 5 states the two properties this construction guarantees; its statement and proof are in Appendix E.6. The adapter cores and the full-rank repair parameters are optimized only on the retain objective:

$$A_l^{(t+1)} = \text{OPTstep}\left(A_l^{(t)}, \nabla_{A_l} \mathcal{L}_r(\theta^{(t)}), \eta_t\right), \tag{20}$$

$$\theta_{\text{free}}^{(t+1)} = \text{OPTstep}\left(\theta_{\text{free}}^{(t)}, \nabla_{\theta_{\text{free}}} \mathcal{L}_r(\theta^{(t)}), \eta_t\right). \tag{21}$$

Forget examples are used only once, to estimate $G_{f,l}$ and the fixed bases. "SEMU-style" (Sendera et al., 2025) refers exclusively to the low-rank parameterization of admissible updates; restricted RECAST does not use a SEMU forget objective and remains retain-only during Phase 2. Only tensors in $\mathcal{M}_{\text{res}}$ are adapterized, and the reduced number of trainable coordinates does not by itself imply a proportional reduction in forward-pass cost.

For classification, selected feature-extractor tensors belong to $\mathcal{M}_{\text{res}}$ while the complete head is contracted in Phase 1, excluded from restriction, and optimized freely in Phase 2. The restriction therefore controls representation rebuilding rather than imposing an orientation on the decision rule—and since a freely adapting head could not by itself establish representation-level unlearning, our conclusions are anchored by head-independent diagnostics.

B.7 Recovery Protocol

Source checkpoints are selected within a learning-rate-controlled cohort restricted to unlearning runs with learning rate at most 10^{-3}. Within this cohort, the same behavioral selection criterion as in the main comparison is used independently for each method. Recovery trains on $\mathcal{D}_f \cup \mathcal{D}_r$ with true labels and no forget-set oversampling, using SGD with learning rate 10^{-3}, momentum 0.9, weight decay 5×10^{-4}, constant learning rate, and a 50-epoch budget.

For $p \in \{0.85, 0.90, 0.95\}$,

$$T_p = \min\left\{t : a_f^{\text{val}}(t+i) \geq p\, a_f^{\text{pre,val}} \quad \forall i \in \{0,1,2\}\right\}.$$

The criterion uses only validation forget accuracy and requires the threshold to hold for three consecutive evaluations. Here $a_f^{\text{pre,val}}$ is the validation-forget accuracy of the single fixed seed-2 Pretrained checkpoint; no train-forget measurement enters the criterion.

Relative parameter displacement is

$$D_\theta = 100 \frac{\|\theta_u - \theta_p\|_2}{\|\theta_p\|_2}.$$

Table 9: Behavioral profiles of the $\mathrm{lr} \leq 10^{-3}$ representatives used as sources for the recovery experiment. Checkpoint gives the selected epoch for seeds 2/45/100. BP-AG is the discrepancy from the retain-only Retrained reference over forget accuracy and the three MIA measurements (lower is better). ‡ marks configurations whose admissible checkpoints arise only from the fallback rule because the forget qualification is not satisfied.

Method	LR	Epochs	BP-AG↓	MIA-L	MIA-E	MIA-C	Test F↓	Test R↑
RECAST (floor 0.10)	10^{-3}	19/19/19	$\mathbf{0.0027 \pm 0.0011}$	49.50 ± 1.20	14.68 ± 0.25	100.00	0.00	68.50 ± 0.11
DEL (no BN reset, $\alpha = 10^{-2}$)	10^{-3}	9/4/4	0.0090 ± 0.0055	49.50 ± 2.17	13.98 ± 2.59	100.00	0.00	66.93 ± 0.20
WIG (frac. 0.25)	10^{-3}	49/44/44	0.0263 ± 0.0026	57.87 ± 0.93	16.17 ± 0.40	100.00	0.40 ± 0.00	65.97 ± 0.12
DELETE	10^{-3}	34/34/49	0.0353 ± 0.0030	60.77 ± 1.50	16.57 ± 0.56	100.00	0.70 ± 0.10	67.78 ± 0.18
SEMU (EVR 0.9)	10^{-3}	49/49/49	0.0662 ± 0.0039	63.50 ± 1.41	2.75 ± 0.28	100.00	0.20	67.31 ± 0.14
SalUn (frac. 0.5)‡	10^{-3}	49/49/49	0.1193 ± 0.0027	75.40 ± 0.72	0.14 ± 0.04	100.00	6.97 ± 0.31	68.56 ± 0.07
FT‡	10^{-3}	49/49/49	0.1762 ± 0.0017	79.67 ± 0.67	2.11 ± 0.06	99.30 ± 0.04	26.70 ± 0.10	68.83 ± 0.06
Retrained	–	–	0.0000	49.20	14.71	100.00	0.00	67.69
Pretrained	–	–	–	76.00	99.81	1.12	69.20	67.49

The norm covers Conv/Linear weights and biases together with BatchNorm affine parameters (γ, β). It excludes `running_mean`, `running_var`, `num_batches_tracked`, and all other non-learnable buffers.

For direct linear mode connectivity, we evaluate

$$\theta(\alpha) = (1-\alpha)\theta_p + \alpha\theta_u, \qquad \alpha \in \{0, 0.05, \ldots, 1\},$$

and linearly interpolate the BatchNorm running buffers along with the learnable parameters. On the held-out test-retain split (`test_retain`), the endpoint chord is

$$A_R^{\mathrm{chord}}(\alpha) = (1-\alpha)A_R(0) + \alpha A_R(1),$$

and the direct test-retain barrier is

$$B_R = \max_\alpha \left[A_R^{\mathrm{chord}}(\alpha) - A_R(\alpha)\right]_+ .$$

For BN-reset/recomputed LMC, the learnable parameters follow the same interpolation path. At every α, BatchNorm running statistics are reset and then re-estimated on retain-train with the standard training augmentation, without updating learnable parameters. BatchNorm affine parameters remain part of the interpolated learnable parameters. Let $\widetilde{A}_R(\alpha)$ denote the resulting test-retain accuracy and let $\widetilde{A}_R^{\mathrm{chord}}(\alpha)$ use the recalibrated endpoint accuracies. The recalibrated barrier is

$$B_R^{\mathrm{BN}} = \max_\alpha \left[\widetilde{A}_R^{\mathrm{chord}}(\alpha) - \widetilde{A}_R(\alpha)\right]_+ .$$

Each test-retain LMC barrier is computed and clipped at zero separately for each seed; the reported value is the mean and sample standard deviation of the seedwise barriers. Curves are not averaged before computing the barrier. We analogously compute a forget-side chord deviation from the direct and BN-recalibrated `test_forget` trajectories, but do not interpret it as a retain-style barrier. Full retain and forget interpolation curves are reported in the appendix.

For the separate auxiliary validation-retain stability summary in Appendix F, let $R_m(t)$ have source value $S_m = R_m(0)$. Its stability deficit is

$$\delta_m(t) = \max\{0, S_m - R_m(t)\}, \tag{22}$$

and that auxiliary analysis reports both $S_m - \min_t R_m(t)$ and the trapezoidal AUC of $\delta_m(t)$. This statistic is not part of the final recovery-threshold or LMC definitions above.

B.8 Learning-Rate-Matched Sources for Recovery

The main recovery experiment uses the representatives selected under the additional constraint $\mathrm{lr} \leq 10^{-3}$. Selection otherwise follows the same validation-based procedure as the main comparison: checkpoints are selected by BP-AG, and representation measurements are computed only after selection. Tables 9 and 10 report the immediate post-unlearning endpoints from which the recovery trajectories start. Values are means and standard deviations over seeds 2, 45, and 100.

The recovery cohort preserves the same broad ordering observed in the main comparison while fixing the unlearning learning-rate regime to $\mathrm{lr} \leq 10^{-3}$. RECAST remains the closest method to Retrained under BP-AG, while DEL obtains the lowest IP-AG. The ‡ rows are retained only for completeness: their selected checkpoints do not satisfy the main forget qualification.

Table 10: Representation profiles and parameter displacement for the same $\mathrm{lr} \leq 10^{-3}$ recovery-source checkpoints. IP-AG averages retraining-relative absolute discrepancies in NC3-F, calibrated NCC-F, and CKA-to-Retrained (lower is better). Displacement is reported relative to the Pretrained model under two parameter scopes: "Disp. conv/lin" excludes BatchNorm affine parameters, whereas "Disp. +BN" matches the D_θ scope used in the main recovery table.

Method	IP-AG↓	NC3-F	NCC-F	CKA→R	Train F↓	Train R↑	Disp. conv/lin	Disp. +BN
RECAST (floor 0.10)	0.1579 ± 0.0016	0.2362 ± 0.0005	0.617 ± 0.003	0.823 ± 0.004	0.00	99.98 ± 0.01	69.58 ± 0.01	60.33 ± 0.01
DEL (no BN reset, $\alpha = 10^{-2}$)	$\mathbf{0.1264 \pm 0.0039}$	0.5017 ± 0.0022	0.674 ± 0.008	0.755 ± 0.005	0.00	99.97 ± 0.01	57.24 ± 0.67	49.69 ± 0.60
WIG (frac. 0.25)	0.1880 ± 0.0026	0.1629 ± 0.0003	0.610 ± 0.010	0.800 ± 0.002	0.54 ± 0.11	99.99 ± 0.00	197.17 ± 1.82	170.93 ± 1.57
DELETE	0.1826 ± 0.0023	0.2882 ± 0.0120	0.706 ± 0.003	0.786 ± 0.003	0.53 ± 0.20	99.99 ± 0.00	5.43 ± 0.42	4.97 ± 0.37
SEMU (EVR 0.9)	0.1944 ± 0.0014	0.2984 ± 0.0029	0.707 ± 0.001	0.742 ± 0.004	0.36 ± 0.19	99.99	7.81 ± 0.06	6.76 ± 0.06
SalUn (frac. 0.5)‡	0.2481 ± 0.0029	0.1084 ± 0.0023	0.706 ± 0.008	0.769 ± 0.001	27.63 ± 1.57	99.99	14.08 ± 0.02	13.49 ± 0.02
FT‡	0.2632 ± 0.0005	0.0599 ± 0.0003	0.743 ± 0.003	0.810 ± 0.001	76.53 ± 0.13	99.99	14.32 ± 0.00	13.60 ± 0.00
Retrained	0.0000	0.4791	0.563	1.000	0.00	99.99	–	–
Pretrained	–	–	–	–	99.99	99.99	0.00	0.00

B.9 Ablation Grid and Additional Protocols

Table 11: Ablation axes. Each row isolates one design decision and names the claim it tests. Reported metrics are Acc. F, Acc. R, mean NCC F, NC3 F, and $T_{85}/T_{90}/T_{95}$ under the designated recovery protocol. Accuracy and internal diagnostics are interpreted relative to the Retrained reference; larger T_p indicates slower recovery.

Axis	Values	Claim under test
Contraction floor $s_{\min}$	$\{0,\ 10^{-8},\ 10^{-4},\ 10^{-2},\ 10^{-1},\ 0.5,\ 0.9\}$	$s_{\min}$ is a monotone intervention-strength knob (Proposition 1)
Direction awareness	alignment-aware vs. mean-matched flat contraction $\bar{s}_l = \lvert\mathcal{G}_l\rvert^{-1} \sum_q s_{l,q}$	**key ablation:** equal average contraction, no direction information—should be worse by Proposition 4.
Transformation scope	x–fc for $x \in \{\text{conv}, \text{layer1}, \ldots, \text{layer4}\}$	depth at which contraction must act
Phase composition	Phase 1 only / + retain consolidation / + LGA / + PL	consolidation is restorative, and adding a forget loss is unnecessary
Restriction	unrestricted vs. SVD-restricted, EVR τ	the optional lane's cost and benefit

Table 12: Efficiency on TinyImageNet-200/ResNet-50. Wall-clock covers the complete procedure from the source checkpoint to the final model on one GPU at batch size 128. Speedup is relative to the 200-epoch retraining reference.

Method	Epochs	Wall-clock (min)	Peak mem. (GB)	Speedup
Retrained *(reference)*	200	—	—	1.0×
ROFT	20	—	—	—
LGA	20	—	—	—
SalUn	20	—	—	—
SEMU+LGA	20	—	—	—
SSD	0 (one shot)	—	—	—
Unrestricted RECAST	1 sweep + 20	—	—	—
Restricted RECAST	1 sweep, SVD, +20	—	—	—

A deterministic seed-2 permutation partitions Tiny ImageNet into the five blocks of Equation 55; each manifest has a stored SHA-256 identifier embedded in its checkpoints. Every unlearning step runs for 20 epochs and initializes from the previous step. Forget and retain batch sizes are 128. The optimizer uses a two-epoch warm-up from $10^{-2}\eta_{\text{base}}$ followed by the common unlearning schedule: SGD milestones at epochs 14 and 18 with factor 0.1, or cosine decay for AdamW. The global 200-class head is not reset. Retrained-50% follows the common 200-epoch reference schedule on the final retain set. Final centroids are fitted on deterministic, non-augmented training features and evaluated on the public test split. We also compare the sequential trajectory with retain-only exposure, retraining on the final retain set, and a single direct-50% request from the pretrained checkpoint.

The ablation grid of Table 11 varies the transformation scope x–fc for $x \in \{\text{conv}, \text{layer1}, \text{layer2}, \text{layer3}, \text{layer4}\}$; $s_{\min} \in \{0, 10^{-8}, 10^{-4}, 10^{-2}, 10^{-1}, 0.5, 0.9\}$; alignment-

aware versus mean-matched flat contraction; a Phase-1-only diagnostic, standard retain-only consolidation, or consolidation augmented with LGA or PL; and unrestricted versus SVD-restricted updates with energy threshold τ.

B.10 Main Classification Protocol and Search Space

The main TinyImageNet-200/ResNet-50 comparison uses a fixed random 20-class forget split (`dataset_seed=2`) and reporting seeds 2, 45, and 100. The Pretrained and retain-only Retrained references are fixed checkpoints trained once with seed 2; reporting seeds vary only the unlearning runs. All iterative methods start from that Pretrained checkpoint and run for 50 epochs with SGD, momentum 0.9, weight decay 5×10^{-4}, batch size 128, and constant learning rate. The common learning-rate sweep is $\{10^{-3}, 3{\times}10^{-3}, 10^{-2}, 3{\times}10^{-2}\}$. Method-specific search ranges are:

Table 13: Method-specific search ranges for the main TinyImageNet-200 comparison.

Method	Additional search parameter
RECAST	`s_min` $\in \{.03, .08, .10, .15, .20, .25\}$
WIG	`wig_reinit_fraction` $\in \{.20, .25, .30, .35, .40, .50\}$
DEL*	`del_alpha` $\in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$
DELETE	no additional search parameter
FT	no additional search parameter
SalUn	`salun_saliency_fraction` $\in \{.1, .3, .5\}$
SEMU	`explained_variance_ratio` $\in \{.8, .9, .95, .99\}$

*Our DEL variant excludes BatchNorm parameters from reinitialization; the original DEL procedure also resets them. SalUn and SEMU use Random Label as their forgetting objective in this comparison. Detailed checkpoint selection and metric definitions are given in Appendix I.

Recovery starts from the behaviorally selected source checkpoints and trains the full model for 50 epochs on the natural union $\mathcal{D}_f \cup \mathcal{D}_r$ with true labels, using fresh SGD with learning rate 10^{-3}, momentum 0.9, weight decay 5×10^{-4}, and batch size 128. Each recovery run follows the seed of its corresponding source run (2, 45, or 100).

B.11 Hyperparameters and Reproducibility Summary

Table 14 collects the settings needed to reproduce the reported runs. Values marked `cfg` are configuration-specific and are released with the code rather than fixed globally.

Table 14: Reproducibility summary. The fixed-budget unlearning schedule applies to auxiliary controlled and ablation experiments; the main TinyImageNet table uses method-specific sweeps and selected checkpoints documented in Appendix I.

Stage	Setting	Value
Reference training	epochs / batch	200 / 128
	optimizer	SGD or AdamW (`cfg`)
	base LR η_{base}	`cfg`
	warm-up	5 epochs, linear from $10^{-3}\eta_{\text{base}}$
	schedule	MultiStep 90/140/180 ($\times 0.1$) or cosine
	early stopping	none; final-epoch checkpoint
Auxiliary unlearning	epochs / batch	20 / 128
	warm-up	2 epochs, linear from $10^{-2}\eta_{\text{base}}$
	schedule	MultiStep 14/18 ($\times 0.1$) or cosine
RECAST	contraction scope $\mathcal{M}$	`layer2`, `layer3`, `layer4`, `fc` (seq.; deferred)
	floor $s_{\min}$	`cfg`; ablated over $\{0, 10^{-8}, 10^{-4}, 10^{-2}, 10^{-1}, 0.5, 0.9\}$
	`dash_lambda`	0.03 (seq.; deferred)
	consolidation LR	0.003 (seq.; deferred)
	gradient safety	disabled (seq.; deferred)
	restriction mode	NONE for unrestricted RECAST
	BatchNorm statistics	updated on retain mini-batches
Main recovery	source cohort	unlearning LR $\leq 10^{-3}$; selected by the same BP criterion
	optimizer	SGD, LR 10^{-3}, momentum 0.9, WD 5×10^{-4}
	schedule	50 epochs; constant learning rate
	thresholds	sustained T_{85}, T_{90}, T_{95}
	data	natural union $\mathcal{D}_f \cup \mathcal{D}_r$, true labels, no forget-set oversampling
	seeds	2, 45, 100
	LMC grid	21 points, $\alpha \in \{0, 0.05, \ldots, 1\}$
Seeds	TinyImageNet main comparison	as specified by the main comparison table
	recovery cohort	2, 45, 100
	CIFAR-100 sample diagnostic	fixed manifest; data seed 2
Hardware	GPU / CUDA	

C Extended Related Work

Objective-based and distillation methods. Many unlearning methods change the model only through the training objective. Standard baselines include gradient ascent on the forget set, training the forget set toward random labels, and fine-tuning on the retained data alone (Cadet et al., 2025). SCRUB alternates between pushing a student model away from a frozen teacher on the forget data and keeping it close to the teacher on the retained data (Kurmanji et al., 2023). DELETE removes the true class from the teacher's output and distills the rest into the student (Zhou et al., 2025). Spotter adds a distillation loss on examples near the class boundary and a loss that spreads the forgotten class out in feature space (Ha et al., 2026).

Restricted-update methods. Another group limits where updates can happen. SalUn updates only the weights with the largest forget-gradient saliency (Fan et al., 2024). SEMU restricts updates to a low-rank subspace built from the SVD of forget gradients (Sendera et al., 2025), and RECAST borrows this parameterization for its optional restricted variant. UNSC projects updates into a null space estimated from retained representations (Chen et al., 2024). These methods can reach the same

forget accuracy as weight interventions while changing the model in very different ways, which is why we compare their internal states as well as their outputs.

Sparsity. Model Sparsity prunes the model once by weight magnitude before running a separate unlearning method, and shows that sparse models are easier to unlearn (Jia et al., 2023). Unlike the weight interventions in Section 2, pruning uses neither the forget nor the retained data to decide which weights to change.

DASH and plasticity. DASH (Shin et al., 2024) studies warm-started networks, which learn new data worse than networks trained from scratch. It contracts each weight group according to how well it aligns with the negative loss gradient, estimated with a running average over incoming data. RECAST adapts this contraction rule by computing the gradient once on the retained data and applying it to unlearning instead of plasticity.

D Contemporary-Benchmark Reproduction

D.1 Protocol and Hyperparameters

We rerun the class-forgetting benchmark from Table 2 of Lee et al. (2026) to obtain checkpoints for extending the published behavioral evaluation with internal profiles. The experiment uses the authors' released repository, preactivation ResNet-18, CIFAR-10, and complete removal of class 5. The training set is split into 80% training and 20% validation examples; all reported entries aggregate seeds 1–3. We use matched pretrained and retain-only retrained checkpoints at index 25 (26 completed passes), deterministic evaluation transforms, final unlearning checkpoints, and no training augmentation. This protocol is separate from the 200/20-epoch schedule of the main controlled comparisons.

Hyperparameter selection follows the released WIG procedure on seed 1; seeds 2 and 3 are reporting seeds. The procedure includes a test-accuracy gate, so this is a reproduction of the released selection procedure, not leakage-free evidence for selecting configurations in later benchmarks.

RECAST is our addition and has no published counterpart in the source table. Its retain-contraction configuration uses a single retain-gradient pass, $s_{\min} = 10^{-2}$, learning rate 3×10^{-3}, 10 consolidation epochs, SGD momentum 0.9, and weight decay 5×10^{-4}.

D.2 Implementation Adjustments and Scissorhands Audit

We distinguish additions needed for the representation evaluation from a correction to the transferred Scissorhands implementation (Wu & Harandi, 2024). The released SHs result is retained rather than replaced by the corrected control. In the full behavioral tables, ordinary entries use x/y (our reproduction/published result), whereas SHs entries use $x/z/y$ (our reproduction/corrected implementation/published result). The full internal-profile tables use x/z: the released SHs profile followed by the final corrected results obtained under the authors' Table 2 protocol. No across-seed dispersion or paired distances D_3, D_4 are available for the corrected profile; these are not inferred from the released profile or from differences between aggregate means.

The experiment adds the following functionality and adaptations to the released benchmark:

- **RECAST integration.** Add retain-guided weight contraction and retain-only consolidation, using the configuration above and the same evaluation protocol as the reproduced methods.
- **Representation evaluation.** Extract the encoder output after global average pooling and before the linear classifier from the reproduced checkpoints, construct class centroids, and evaluate the internal profile specified below.
- **SHs classifier mapping.** Adapt the official Scissorhands implementation to the WIG model by mapping its classifier name to the WIG ResNet head. Retain the authors' SNIP training mode; this mode is preserved, not introduced as another experimental change.
- **SHs projection margin.** In the corrected control, replace the imposed projection lower bound 0.50 by the exact half-space margin 0. The unmodified released path remains the x entry.

One seed of the released SHs path collapses retain-test accuracy to 33.64%, reducing its three-seed mean to 67.07%. The source audit attributes this behavior to the transferred implementation rather than a failed job. The corrected control restores retain-train accuracy to $99.97 \pm 0.01\%$ and retain-test accuracy to $84.19 \pm 0.16\%$. Its MIA-loss mean remains 3.97 percentage points above the published value, so it is an experimental variant, not an exact reproduction of published SHs. Both versions are reported in the same row to preserve this distinction without introducing another method label.

D.3 Complete Behavioral Profiles

Table 15: Full source comparisons for the selected behavioral profiles in Table 1(a). Within each metric, Repr./Corr./Publ. denote reproduction/corrected SHs implementation/published result, separated into subcolumns by vertical rules under grouped metric headers. Reproduction values are three-seed means. The corrected column applies only to SHs; – denotes an unavailable or inapplicable counterpart. Values are in percent; Avg Gap is the reported mean absolute gap from the local retrained reference over MIA-loss, MIA-entropy, MIA-confidence, and forget-train accuracy. MIA-L/E/C use loss/entropy/confidence; F/R denote forget/retain and tr/te train/test. The SHs correction is described in Appendix D.2.

Method	MIA-L			MIA-E			MIA-C			Avg Gap		
	Repr.	Corr.	Publ.	Repr.	Corr.	Publ.	Repr.	Corr.	Publ.	Repr.	Corr.	Publ.
Pretrained	70.50	–	–	99.75	–	–	0.01	–	–	68.70	–	–
Retrained	49.77	–	50.03	45.69	–	44.62	100.00	–	100.00	0.00	–	0.00
FT	72.33	–	70.17	8.33	–	8.37	88.53	–	87.82	35.45	–	34.93
NegGrad	65.10	–	64.50	17.47	–	16.35	98.51	–	98.59	12.59	–	12.34
SalUn	55.93	–	56.33	35.94	–	35.23	100.00	–	100.00	3.98	–	3.92
SSD	54.47	–	53.80	20.65	–	36.36	100.00	–	100.00	7.44	–	3.01
SHs	50.43	52.57	48.60	59.29	51.52	50.51	100.00	100.00	100.00	3.57	2.16	1.83
WIG	51.50	–	51.83	44.94	–	42.33	100.00	–	100.00	0.62	–	1.02
RECAST	50.63	–	–	43.18	–	–	100.00	–	–	0.85	–	–

Method	Acc. F-tr			Acc. R-tr			Acc. R-te		
	Repr.	Corr.	Publ.	Repr.	Corr.	Publ.	Repr.	Corr.	Publ.
Pretrained	100.00	–	–	100.00	–	–	84.01	–	–
Retrained	0.00	–	0.00	100.00	–	100.00	85.37	–	84.89
FT	70.41	–	71.16	100.00	–	100.00	85.59	–	84.75
NegGrad	5.32	–	5.21	99.08	–	99.19	81.99	–	81.84
SalUn	0.00	–	0.00	100.00	–	100.00	86.79	–	86.20
SSD	0.00	–	0.00	85.41	–	77.90	73.47	–	67.39
SHs	0.00	0.00	0.00	77.96	99.97	100.00	67.07	84.19	83.54
WIG	0.00	–	0.00	100.00	–	100.00	81.49	–	80.83
RECAST	0.00	–	–	99.97	–	–	86.24	–	–

Tables 16 and 17 contain every completed method. Values are three-seed means. "Original" denotes the pretrained model. The MIA columns use loss, entropy, and confidence, respectively, under the released attack protocol. Avg Gap follows the released WIG aggregation: the mean absolute gap from the local retrained reference over these three MIA measurements and forget-train accuracy. It summarizes selected behavioral measurements, not the internal profile; lower values indicate closer agreement with retraining on these measurements.

Table 16: Complete accuracy reproduction on CIFAR-10/preactivation ResNet-18 with class 5 forgotten. Entries are reproduction/published results in percent; SHs uses reproduction/corrected implementation/published results. The correction is described in Appendix D.2. The symbol – denotes no published counterpart.

Method	Forget train	Retain train	Retain test
Original	100.00/–	100.00/–	84.01/–
Retrained	0.00/0.00	100.00/100.00	85.37/84.89
FT	70.41/71.16	100.00/100.00	85.59/84.75
GA	5.93/6.08	85.91/86.65	73.41/73.74
NegGrad	5.32/5.21	99.08/99.19	81.99/81.84
CFK	71.84/74.29	100.00/100.00	85.41/84.62
EUK	0.00/0.00	88.48/87.90	76.87/77.04
Bad-T	27.42/29.52	99.96/99.97	84.36/83.73
SCRUB+R	0.00/0.00	99.99/99.99	85.39/84.66
SSD	0.00/0.00	85.41/77.90	73.47/67.39
SalUn	0.00/0.00	100.00/100.00	86.79/86.20
SHs	0.00/0.00/0.00	77.96/99.97/100.00	67.07/84.19/83.54
WIG	0.00/0.00	100.00/100.00	81.49/80.83
RECAST	0.00/–	99.97/–	86.24/–

Table 17: Complete membership-inference reproduction. Entries are reproduction/published results in percent; SHs uses reproduction/corrected implementation/published results. Avg Gap follows the released WIG aggregation over behavioral measurements and is not a representation distance. The symbol – denotes no published counterpart.

Method	MIA loss	MIA entropy	MIA confidence	Avg Gap
Original	70.50/–	99.75/–	0.01/–	68.70/–
Retrained	49.77/50.03	45.69/44.62	100.00/100.00	0.00/0.00
FT	72.33/70.17	8.33/8.37	88.53/87.82	35.45/34.93
GA	59.67/59.67	21.89/21.40	96.31/96.46	10.83/10.62
NegGrad	65.10/64.50	17.47/16.35	98.51/98.59	12.59/12.34
CFK	72.40/70.57	9.20/9.58	87.89/86.81	35.77/35.77
EUK	51.33/50.57	60.04/31.91	100.00/100.00	3.98/3.31
Bad-T	68.43/69.13	4.70/3.73	97.07/97.70	22.50/22.95
SCRUB+R	56.43/54.80	44.45/46.40	100.00/100.00	1.98/1.64
SSD	54.47/53.80	20.65/36.36	100.00/100.00	7.44/3.01
SalUn	55.93/56.33	35.94/35.23	100.00/100.00	3.98/3.92
SHs	50.43/52.57/48.60	59.29/51.52/50.51	100.00/100.00/100.00	3.57/2.16/1.83
WIG	51.50/51.83	44.94/42.33	100.00/100.00	0.62/1.02
RECAST	50.63/–	43.18/–	100.00/–	0.85/–

The matched references and many reproduced baselines are numerically close to the published results, although this is not uniform: the released SHs path is unstable, and SSD and EUK show notable discrepancies in some behavioral measurements. WIG obtains the smallest reproduced Avg Gap (0.62), compared with 0.85 for RECAST. RECAST retains higher test accuracy than WIG (86.24% versus 81.49%), while SalUn has the highest retain-test accuracy in the complete table (86.79%).

D.4 INTERNAL PROFILES

Table 18: Full source comparisons for the selected internal profiles in Table 1(b). Each metric has released/corrected subcolumns; vertical rules separate the source values under grouped metric headers. Corrected values are available only for SHs and have no available dispersion or paired distances; – denotes an unreported or inapplicable counterpart. Values with ± retain the reported three-seed dispersion. F/R denote forget/retain validation subsets; NCC is nearest-class-centroid accuracy and NC4 classifier/NCC disagreement, both in percent. $\mathrm{NC3}_F^{\cos}$ is the forgotten-class classifier–centered- centroid cosine; its dispersion is the standard deviation over three seeds. D_3 and D_4 are reported paired same-seed distances to Retrained (lower is closer). These are internal diagnostics, not published comparisons.

Method	$\mathrm{NC3}_F^{\cos}$		NCC-F		NCC-R	
	Released	Corrected	Released	Corrected	Released	Corrected
Pretrained	0.91 ± 0.00	–	70.20 ± 0.71	–	83.52 ± 0.35	–
Retrained	-0.09 ± 0.00	–	55.03 ± 0.80	–	80.93 ± 0.28	–
FT	0.82 ± 0.00	–	69.13 ± 0.62	–	83.72 ± 0.24	–
NegGrad	0.76 ± 0.00	–	66.87 ± 0.40	–	82.60 ± 0.41	–
SalUn	0.19 ± 0.02	–	71.63 ± 1.53	–	82.34 ± 0.43	–
SSD	-0.12 ± 0.03	–	54.87 ± 1.76	–	78.19 ± 0.62	–
SHs	0.07 ± 0.30	-0.16	44.77 ± 13.26	52.20	61.42 ± 24.69	79.99
WIG	0.63 ± 0.03	–	61.33 ± 1.21	–	77.01 ± 0.64	–
RECAST	-0.20 ± 0.06	–	55.83 ± 2.83	–	80.36 ± 0.53	–

Method	NC4-F		NC4-R		D_3		D_4	
	Released	Corrected	Released	Corrected	Released	Corrected	Released	Corrected
Pretrained	6.00 ± 0.62	–	5.21 ± 0.15	–	15.17 ± 1.51	–	50.10 ± 0.54	–
Retrained	56.10 ± 1.00	–	10.80 ± 0.27	–	0.00 ± 0.00	–	0.00 ± 0.00	–
FT	36.37 ± 0.78	–	5.85 ± 0.29	–	14.10 ± 0.80	–	19.73 ± 1.07	–
NegGrad	68.53 ± 0.58	–	12.77 ± 0.21	–	11.83 ± 0.40	–	12.43 ± 0.98	–
SalUn	71.90 ± 1.49	–	9.89 ± 0.44	–	16.60 ± 1.31	–	15.80 ± 1.50	–
SSD	68.83 ± 1.95	–	24.32 ± 1.30	–	1.43 ± 0.92	–	12.73 ± 1.03	–
SHs	62.07 ± 9.89	53.20	27.25 ± 21.85	10.43	10.33 ± 13.56	–	6.83 ± 8.54	–
WIG	62.93 ± 0.95	–	11.70 ± 0.44	–	6.30 ± 0.51	–	6.83 ± 0.52	–
RECAST	55.93 ± 2.88	–	12.35 ± 0.19	–	1.93 ± 1.20	–	1.90 ± 0.92	–

We evaluate the 512-dimensional encoder output before the linear classifier, after global average pooling. Class centroids are constructed from all 40,000 training examples, including the forgotten class, using deterministic transforms. This access to forgotten examples is for post-unlearning diagnostics only; it is not an input to the RECAST transformation or consolidation. NCC classifies validation features using the centroid bank. NC4 is the percentage of validation examples for which the native classifier and NCC predict different labels. NCC probes feature separability, while NC4 measures agreement between the native classifier and a feature-based read-out. We report $\mathrm{NC3}_F^{\cos}$, the cosine between the forgotten-class classifier weight and its centered feature centroid:

$$\mathrm{NC3}_F^{\cos} = \cos(w_F, \tilde{\mu}_F) = \frac{\langle w_F, \tilde{\mu}_F \rangle}{\|w_F\|_2 \|\tilde{\mu}_F\|_2}. \tag{23}$$

Here w_F is the classifier weight vector for the forgotten class (class 5) and $\tilde{\mu}_F$ is its centered feature centroid. This statistic measures classifier–centroid alignment for the forgotten class and is interpreted relative to the retrained reference. The profile also includes the paired distances D_3 and D_4 from the retrained reference for the same seed; lower distances indicate closer agreement with retraining. These paired distances are reported directly, not recomputed as absolute differences between the displayed across-seed means. Cosine means and standard deviations over three seeds come from the cosine evaluation; NCC, NC4, and distance entries retain the across-seed dispersion from the full internal-profile summary.

Table 19 collects the full internal profiles; Table 1(b) presents selected diagnostics in the main text. "SCRUB+R final" denotes the final SCRUB+R checkpoint. SHs combines the released

WIG implementation and the final corrected result in a single x/z row. The corrected profile has $\mathrm{NC3}_F^{\cos} = -0.16$.

Table 19: Complete internal profiles for the CIFAR-10 reproduction. Values with $\pm$ are three-seed means with the reported across-seed dispersion. F/R denote forget/retain validation subsets; NCC and NC4 are percentages. D_3 and D_4 are paired distances to the same-seed retrained reference on the percentage scale (lower is closer). $\mathrm{NC3}_F^{\cos}$ is the forgotten-class classifier–centered-centroid cosine; its $\pm$ values are standard deviations over three seeds from the cosine evaluation. SHs uses x/z: released implementation/final corrected result under the authors' protocol, stacked on two lines. No dispersion is available for z; – denotes an unreported measurement. These are newly computed diagnostics, not reproduction/published comparisons.

Method	$\mathrm{NC3}_F^{\cos}$	NCC-F	NCC-R	NC4-F	NC4-R	D_3	D_4
Pretrained	0.91 ± 0.00	70.20 ± 0.71	83.52 ± 0.35	6.00 ± 0.62	5.21 ± 0.15	15.17 ± 1.51	50.10 ± 0.54
Retrained	-0.09 ± 0.00	55.03 ± 0.80	80.93 ± 0.28	56.10 ± 1.00	10.80 ± 0.27	0.00 ± 0.00	0.00 ± 0.00
FT	0.82 ± 0.00	69.13 ± 0.62	83.72 ± 0.24	36.37 ± 0.78	5.85 ± 0.29	14.10 ± 0.80	19.73 ± 1.07
GA	0.76 ± 0.01	65.30 ± 0.28	80.87 ± 0.61	78.57 ± 1.60	22.30 ± 0.45	10.27 ± 0.70	22.47 ± 2.41
NegGrad	0.76 ± 0.00	66.87 ± 0.40	82.60 ± 0.41	68.53 ± 0.58	12.77 ± 0.21	11.83 ± 0.40	12.43 ± 0.98
Bad-T	0.77 ± 0.01	69.13 ± 0.24	80.68 ± 0.46	59.00 ± 2.09	11.07 ± 0.36	14.10 ± 0.94	2.90 ± 2.98
CFK	0.83 ± 0.00	69.63 ± 0.33	83.43 ± 0.26	36.33 ± 0.39	6.21 ± 0.15	14.60 ± 1.08	19.77 ± 1.26
EUK	-0.02 ± 0.01	65.40 ± 0.50	74.05 ± 0.81	69.37 ± 0.93	13.18 ± 0.97	10.37 ± 1.30	13.27 ± 1.84
SalUn	0.19 ± 0.02	71.63 ± 1.53	82.34 ± 0.43	71.90 ± 1.49	9.89 ± 0.44	16.60 ± 1.31	15.80 ± 1.50
SCRUB+R final	0.75 ± 0.02	59.60 ± 0.86	74.30 ± 0.59	62.13 ± 1.10	19.82 ± 0.60	4.57 ± 0.37	6.03 ± 0.34
SSD	-0.12 ± 0.03	54.87 ± 1.76	78.19 ± 0.62	68.83 ± 1.95	24.32 ± 1.30	1.43 ± 0.92	12.73 ± 1.03
SHs	0.07 ± 0.30 / -0.16	44.77 ± 13.26 / 52.20	61.42 ± 24.69 / 79.99	62.07 ± 9.89 / 53.20	27.25 ± 21.85 / 10.43	10.33 ± 13.56 / –	6.83 ± 8.54 / –
WIG	0.63 ± 0.03	61.33 ± 1.21	77.01 ± 0.64	62.93 ± 0.95	11.70 ± 0.44	6.30 ± 0.51	6.83 ± 0.52
RECAST	-0.20 ± 0.06	55.83 ± 2.83	80.36 ± 0.53	55.93 ± 2.88	12.35 ± 0.19	1.93 ± 1.20	1.90 ± 0.92

RECAST is close to Retrained simultaneously in forget NCC (55.83% versus 55.03%) and forget NC4 (55.93% versus 56.10%). Its paired distances are $D_3 = 1.93$ and $D_4 = 1.90$, compared with 6.30 and 6.83 for WIG despite WIG's smaller behavioral Avg Gap. SSD has the smallest D_3 among unlearning methods (1.43), but its D_4 remains 12.73 and retain NCC falls to 78.19%, against 80.93% for Retrained. SalUn retains high forget NCC (71.63%) despite zero native forget accuracy. The internal profile therefore distinguishes residual feature separability from agreement with the native classifier and should be read jointly with the behavioral profile, rather than reduced to one favorable metric. The released SHs path has large across-seed dispersion in both retain and forget representation measurements, consistent with the implementation instability discussed above. None of these diagnostics establishes that the influence of the forget set is absent.

D.5 Training-Set Feature Visualization

Figure 7 complements the quantitative internal profiles with a training-set visualization of post-encoder features. The plot uses seed 1 and 250 examples per class; class 5 is the forget class and is highlighted with outlined diamonds. It is a single-seed, training-subset illustration, not the three-seed validation evaluation in Table 19. We use it qualitatively, rather than ranking methods by distances or cluster overlap in the two-dimensional projection.

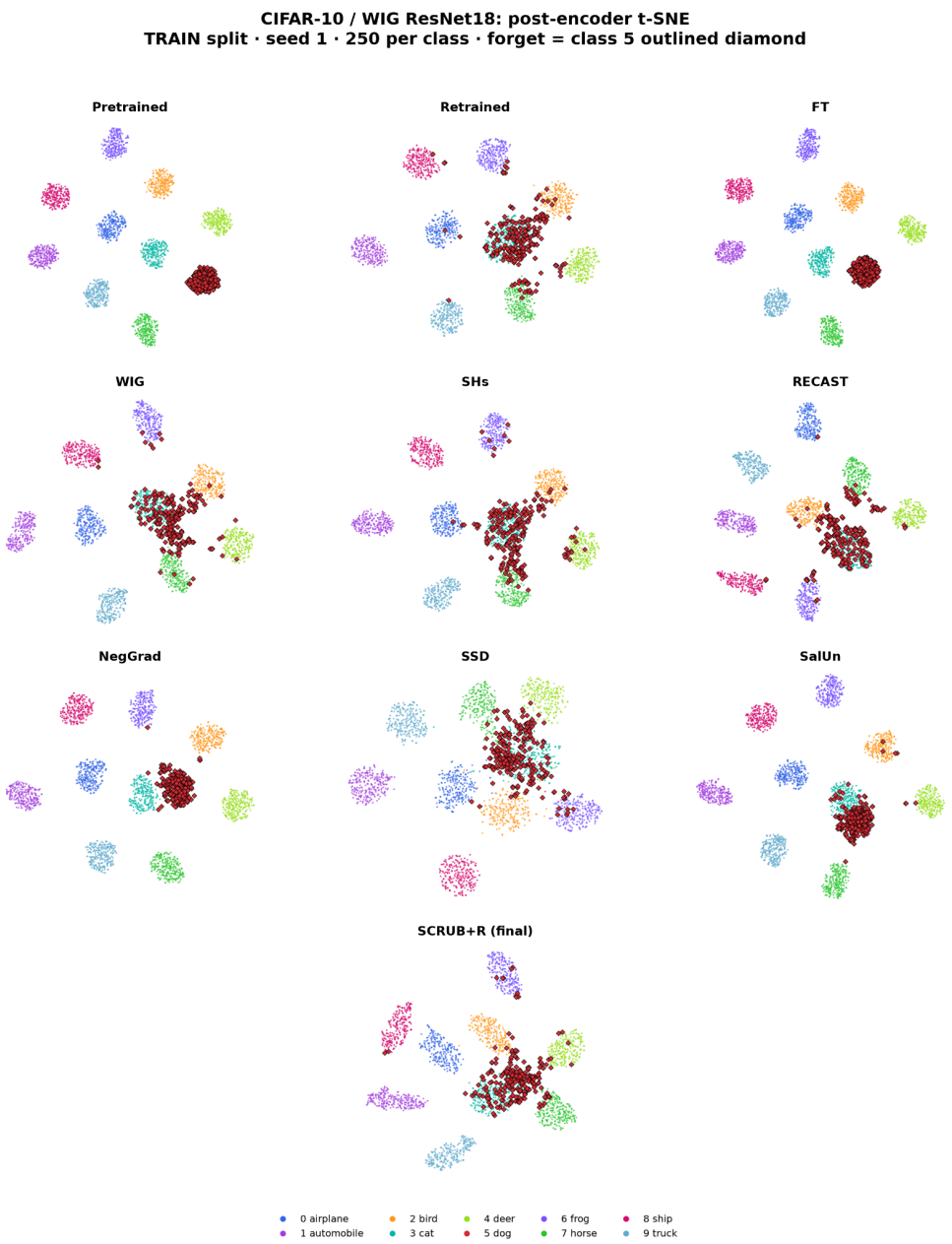

Figure 7: Post-encoder t-SNE visualization for CIFAR-10 with the WIG ResNet-18 model, using the training split, seed 1, and 250 examples per class. Colors identify classes; outlined diamonds highlight class 5, the forget class. Panels show the pretrained and retain-only retrained references alongside FT, WIG, SHs, RECAST, NegGrad, SSD, SalUn, and the final SCRUB+R checkpoint. The panel labels reproduce the original visualization. This figure complements, but does not replace, the quantitative internal diagnostics.

E PROOFS

Throughout this appendix, $\mathcal{M} = \{W_l\}$ is the set of contracted tensors, $\mathcal{G}_l$ is the fixed non-overlapping group partition of W_l, $w_{l,q} = \mathrm{vec}(W_{l,q})$, $g_{l,q} = \mathrm{vec}(G_{r,l}[q])$, $c_{l,q} = \cos(w_{l,q}, -g_{l,q})$, and $s_{l,q} = \max(c_{l,q}, s_{\min})$ with $s_{\min} \in [0, 1]$. We write $\theta|_{\mathcal{M}}$ for the restriction of θ to the coordinates of $\mathcal{M}$, and $\|\cdot\|$ is the Euclidean (Frobenius) norm. Since the groups within a tensor are non-overlapping and their union is the tensor, and since distinct tensors occupy disjoint coordinate blocks, the index set $\{(l, q)\}$ partitions the coordinates of $\theta|_{\mathcal{M}}$. This fact is used repeatedly.

It is convenient to introduce the coordinate embedding operator

$$\mathcal{E}_{l,q} : \mathbb{R}^{d_{l,q}} \to \mathbb{R}^P, \tag{24}$$

the linear map that places its argument into the coordinates occupied by group (l, q) and sets every other coordinate to zero; here $d_{l,q}$ is the number of coordinates in the group and P the total parameter count. $\mathcal{E}_{l,q}$ is pure bookkeeping, not a learned embedding. Because the groups are disjoint, $\mathcal{E}_{l,q}$ has orthonormal columns and $\mathcal{E}_{l,q}^\top \mathcal{E}_{l,q'} = 0$ for $(l, q) \neq (l', q')$.

E.1 STATEMENT AND PROOF OF PROPOSITION 1

Proposition 1 (The contraction floor defines an ordered intervention path)**.** *Let $\tau = s_{\min}$ and $s_{l,q}(\tau) = \max(c_{l,q}, \tau)$, with the alignments $c_{l,q}$ fixed at their values computed at the source model θ_p. Let $\theta_c(\tau)$ denote the model after Phase 1 with floor τ. Define*

$$D^2(\tau) = \|\theta_c(\tau) - \theta_p\|^2 = \sum_{l,q} \big(1 - \max(c_{l,q}, \tau)\big)^2 \|w_{l,q}\|^2.$$

Then $D(\tau)$ is non-increasing on $[0, 1]$. In particular,

$$\tau_2 \geq \tau_1 \quad \Longrightarrow \quad D(\tau_2) \leq D(\tau_1), \qquad D(1) = 0.$$

Away from values $\tau = c_{l,q}$,

$$\frac{dD^2(\tau)}{d\tau} = -2(1 - \tau) \sum_{l,q:\, c_{l,q} < \tau} \|w_{l,q}\|^2 \leq 0.$$

Proof. For each group, the factor $1 - \max(c_{l,q}, \tau)$ is nonnegative and non-increasing in $\tau \in [0, 1]$. Therefore every summand in $D^2(\tau)$ is non-increasing, and so are $D^2(\tau)$ and its nonnegative square root $D(\tau)$. At $\tau = 1$, $\max(c_{l,q}, 1) = 1$ for every group, hence $D(1) = 0$.

On any interval containing no breakpoint $\tau = c_{l,q}$, groups with $c_{l,q} \geq \tau$ contribute the constant $(1 - c_{l,q})^2 \|w_{l,q}\|^2$, whereas groups with $c_{l,q} < \tau$ contribute $(1 - \tau)^2 \|w_{l,q}\|^2$. Differentiating this finite sum gives

$$\frac{dD^2(\tau)}{d\tau} = -2(1 - \tau) \sum_{l,q:\, c_{l,q} < \tau} \|w_{l,q}\|^2 \leq 0.$$

■

Thus $s_{\min}$ parameterizes a continuous, ordered path from stronger contraction toward the pretrained source model. It acts as an intervention-strength parameter rather than only as a numerical safeguard. This is a parameter-space statement and does not imply monotonic retain or forget accuracy.

E.2 CONTINUOUS CONTRACTION VERSUS ZERO-MEAN REINITIALIZATION

We compare the Phase-1 contraction operator with hard reinitialization on the *same fixed set of coordinates*. This isolates the intervention operator from the parameter-selection rule used to choose those coordinates.

Let $u \in \mathbb{R}^d$ denote the selected parameter subvector and let

$$h = \nabla_u \mathcal{L}_r(\theta_p)$$

be the corresponding retain-gradient subvector at the source model. For nonzero u and h, define

$$c = \cos(u, -h), \qquad \kappa = \|u\| \, \|h\|. \tag{25}$$

Hence

$$h^\top u = -\kappa c. \tag{26}$$

Proposition 2 (Continuous contraction versus zero-mean reinitialization)**.** *Consider two interventions on the same parameter subvector u.*

Continuous contraction. *For $s \in [0,1]$, let*

$$u_C = su. \tag{27}$$

Hard reinitialization. *Let*

$$u_H = \xi, \qquad \mathbb{E}[\xi] = 0, \qquad \mathbb{E}\|\xi\|^2 < \infty. \tag{28}$$

Define the first-order retain effects

$$\Delta L^{(1)}_{r,C} = h^\top (u_C - u), \qquad \Delta L^{(1)}_{r,H} = h^\top (u_H - u). \tag{29}$$

Then

$$\Delta L^{(1)}_{r,C} = (1-s)\kappa c, \tag{30}$$

whereas

$$\mathbb{E}\Big[\Delta L^{(1)}_{r,H}\Big] = \kappa c. \tag{31}$$

Consequently,

$$\Delta L^{(1)}_{r,C} = (1-s)\,\mathbb{E}\Big[\Delta L^{(1)}_{r,H}\Big]. \tag{32}$$

In particular, for retain-supported coordinates, $c \geq 0$,

$$\Delta L^{(1)}_{r,C} \leq \mathbb{E}\Big[\Delta L^{(1)}_{r,H}\Big]. \tag{33}$$

For $c > 0$ and $s > 0$, the inequality is strict.

Proof. Using Equation 26,

$$\begin{aligned}\Delta L^{(1)}_{r,C} &= h^\top (su - u) \\ &= (s-1)h^\top u \\ &= (1-s)\kappa c.\end{aligned}$$

For hard reinitialization,

$$\begin{aligned}\mathbb{E}\Big[\Delta L^{(1)}_{r,H}\Big] &= \mathbb{E}\big[h^\top(\xi - u)\big] \\ &= h^\top \mathbb{E}[\xi] - h^\top u \\ &= \kappa c,\end{aligned}$$

because $\mathbb{E}[\xi] = 0$. Equation 32 follows immediately. If $c \geq 0$ and $s \in [0,1]$, then $(1-s)\kappa c \leq \kappa c$. ■

Corollary 1 (Expected squared displacement)**.** *Under the assumptions of Proposition 2,*

$$\|u_C - u\|^2 = (1-s)^2 \|u\|^2, \tag{34}$$

whereas

$$\mathbb{E}\|u_H - u\|^2 = \|u\|^2 + \mathbb{E}\|\xi\|^2. \tag{35}$$

Hence,

$$\|u_C - u\|^2 \leq \|u\|^2 \leq \mathbb{E}\|u_H - u\|^2. \tag{36}$$

Proof. For contraction,

$$\|u_C - u\|^2 = \|(s-1)u\|^2 = (1-s)^2\|u\|^2.$$

For reinitialization,

$$\begin{aligned}\mathbb{E}\|u_H - u\|^2 &= \mathbb{E}\|\xi - u\|^2 \\ &= \mathbb{E}\|\xi\|^2 - 2u^\top \mathbb{E}[\xi] + \|u\|^2 \\ &= \|u\|^2 + \mathbb{E}\|\xi\|^2.\end{aligned}$$

The ordering follows from $s \in [0,1]$ and $\mathbb{E}\|\xi\|^2 \geq 0$. ■

Proposition 2 is conditional on modifying the same coordinates. It therefore compares the intervention operators rather than the complete RECAST, WIG, or DEL algorithms, whose parameter-selection rules differ. The result is strongest for retain-supported coordinates ($c > 0$): contraction scales the first-order retain effect by the controllable factor $1 - s$, whereas zero-mean hard reinitialization removes the full pre-existing component in expectation. Corollary 1 additionally shows that hard reinitialization contributes both the squared norm of the removed subvector and the expected energy of the newly sampled parameters to the expected squared displacement.

E.3 Auxiliary Bounded-Intervention Result

Corollary 2 (Bounded intervention)**.** *The Phase-1 displacement satisfies*

$$\big\|\theta_c - \theta_p\big\| = \Big(\textstyle\sum_{l,q}(1-s_{l,q})^2\|w_{l,q}\|^2\Big)^{1/2} \;\le\; (1-s_{\min})\,\big\|\theta_p|_{\mathcal{M}}\big\|, \tag{37}$$

with equality when every group with nonzero weight norm is contracted to the floor. Coordinates outside $\mathcal{M}$ are unchanged, and no group's norm increases: $\|w^c_{l,q}\| \le \|w_{l,q}\|$ for all (l,q).

Proof. By Equation 6, $W^c_{l,q} = s_{l,q}W_{l,q}$, so the displacement of group (l,q) is $\mathcal{E}_{l,q}\big((s_{l,q}-1)w_{l,q}\big)$ and coordinates outside $\mathcal{M}$ are unchanged. Because the group embeddings have mutually orthogonal ranges,

$$\big\|\theta_c - \theta_p\big\|^2 = \sum_{l,q}\big\|(s_{l,q}-1)w_{l,q}\big\|^2 = \sum_{l,q}(1-s_{l,q})^2\|w_{l,q}\|^2. \tag{38}$$

Since $c_{l,q} \le 1$ always and $s_{l,q} = \max(c_{l,q}, s_{\min})$, we have $s_{l,q} \in [s_{\min}, 1]$ and hence $1 - s_{l,q} \in [0, 1-s_{\min}]$. Substituting the upper bound into Equation 38 gives

$$\big\|\theta_c - \theta_p\big\|^2 \le (1-s_{\min})^2 \sum_{l,q}\|w_{l,q}\|^2 = (1-s_{\min})^2\big\|\theta_p|_{\mathcal{M}}\big\|^2, \tag{39}$$

where the last equality is again the disjointness of the groups. Taking square roots yields Equation 37. Equality requires $1 - s_{l,q} = 1 - s_{\min}$, i.e. $s_{l,q} = s_{\min}$, for every group with $w_{l,q} \neq 0$. Finally $\|w^c_{l,q}\| = s_{l,q}\|w_{l,q}\| \le \|w_{l,q}\|$ because $s_{l,q} \le 1$. ■

E.4 Statement and Proof of Proposition 3

Proposition 3 (Utility-damage bound)**.** *Assume $\mathcal{L}_r$ is L-smooth and write $\kappa_{l,q} = \|g_{l,q}\|\|w_{l,q}\| \ge 0$. Then*

$$\mathcal{L}_r(\theta_c) - \mathcal{L}_r(\theta_p) \;\le\; \underbrace{\sum_{l,q}(1-s_{l,q})\,\kappa_{l,q}\,c_{l,q}}_{=:\;\Phi(s)} \;+\; \frac{L}{2}(1-s_{\min})^2\big\|\theta_p|_{\mathcal{M}}\big\|^2. \tag{40}$$

Moreover, under the RECAST rule $s_{l,q} = \max(c_{l,q}, s_{\min})$ every group's first-order contribution satisfies

$$(1-s_{l,q})\,\kappa_{l,q}\,c_{l,q} \;\le\; \tfrac{1}{4}\kappa_{l,q}, \tag{41}$$

with equality only when $c_{l,q} = \frac{1}{2} \ge s_{\min}$, and the contribution is nonpositive whenever $c_{l,q} \le 0$. Hence $\Phi(s) \le \frac{1}{4}\sum_{l,q}\kappa_{l,q}$, a cap independent of $s_{\min}$.

Proof. Write $\delta\theta = \theta_c - \theta_p$ and recall from Equation 4 that $g_{l,q} = \mathrm{vec}\big(\nabla_{W_l}\mathcal{L}_r(\theta_p)[q]\big)$, so that the group-$(l,q)$ block of $\nabla_\theta\mathcal{L}_r(\theta_p)$ is exactly $g_{l,q}$. L-smoothness gives the descent lemma

$$\mathcal{L}_r(\theta_p + \delta\theta) - \mathcal{L}_r(\theta_p) \le \big\langle \nabla_\theta\mathcal{L}_r(\theta_p), \delta\theta\big\rangle + \frac{L}{2}\|\delta\theta\|^2. \tag{42}$$

The inner product decomposes over the disjoint groups:

$$\big\langle \nabla_\theta\mathcal{L}_r(\theta_p), \delta\theta\big\rangle = \sum_{l,q}(s_{l,q}-1)\langle g_{l,q}, w_{l,q}\rangle. \tag{43}$$

By the definition of the cosine, $\langle g_{l,q}, w_{l,q}\rangle = -\|g_{l,q}\|\|w_{l,q}\|c_{l,q}$, since $c_{l,q} = \cos(w_{l,q}, -g_{l,q})$. Substituting,

$$\big\langle \nabla_\theta\mathcal{L}_r(\theta_p), \delta\theta\big\rangle = \sum_{l,q}(1-s_{l,q})\|g_{l,q}\|\|w_{l,q}\|c_{l,q} = \Phi(s). \tag{44}$$

Bounding the quadratic term with Corollary 2 gives Equation 40.

For the per-group cap, fix a group and abbreviate $\kappa = \kappa_{l,q} = \|g_{l,q}\|\|w_{l,q}\| \ge 0$ and $c = c_{l,q} \in [-1,1]$. The group's contribution to Φ is $\varphi = (1-s)\kappa c$ with $s = \max(c, s_{\min})$ and $s_{\min} \in [0,1]$. Three cases exhaust the possibilities:

- $c \le 0$. Then $s_{\min} \ge 0 \ge c$, so $s = s_{\min}$ and $\varphi = (1-s_{\min})\kappa c \le 0$, because $1 - s_{\min} \ge 0$ and $\kappa \ge 0$. Contracting an anti-aligned group therefore *lowers* the retain risk to first order.

- $0 < c \le s_{\min}$. Again $s = s_{\min}$, so $\varphi = (1 - s_{\min})\kappa c \le (1 - s_{\min})s_{\min}\kappa \le \frac{1}{4}\kappa$, using $t(1-t) \le \frac{1}{4}$ for $t \in [0,1]$.
- $c > s_{\min}$. Then $s = c$ and $\varphi = c(1-c)\kappa \le \frac{1}{4}\kappa$ by the same inequality, with equality exactly at $c = \frac{1}{2}$.

In every case $\varphi \le \frac{1}{4}\kappa$, which is Equation 41; summing over groups gives $\Phi(s) \le \frac{1}{4}\sum_{l,q}\kappa_{l,q}$, a quantity that depends only on the source checkpoint and the retain gradient, not on $s_{\min}$. Equality in the per-group bound forces $c = \frac{1}{2}$ and $s_{\min} \le \frac{1}{2}$. ■

The cap is the point. A group strongly aligned with the retain descent direction ($c \to 1$) is barely contracted and costs almost nothing; a group that is anti-aligned ($c \le 0$) is contracted to the floor and its first-order contribution is a *decrease* in retain risk. Only groups near $c = \frac{1}{2}$ are expensive, and even those cost at most $\kappa_{l,q}/4$. A rule that ignores alignment enjoys no such guarantee:

E.5 Statement and Proof of Proposition 4

Proposition 4 (Direction-aware allocation versus flat contraction)**.** *Fix a layer l, let $\bar{s}_l = \big(\sum_q \kappa_{l,q}\big)^{-1}\sum_q \kappa_{l,q}s_{l,q}$ be the κ-weighted mean of the RECAST factors, and let Φ_{flat} be the first-order term obtained by replacing every $s_{l,q}$ in that layer with $\bar{s}_l$. Then $\Phi_{\mathrm{flat}} \ge \Phi(s)$, with equality if and only if all $s_{l,q}$ within the layer coincide.*

Proof. Fix a layer l and write κ_q, s_q, c_q for $\kappa_{l,q}, s_{l,q}, c_{l,q}$ and $K_l = \sum_q \kappa_q$. If $K_l = 0$ both sides vanish, so assume $K_l > 0$ and set $\bar{s} = K_l^{-1}\sum_q \kappa_q s_q$ and $\bar{c} = K_l^{-1}\sum_q \kappa_q c_q$. Substituting $\bar{s}$ for every s_q,

$$\Phi_{\mathrm{flat}} - \Phi(s) = \sum_q \kappa_q\big(s_q - \bar{s}\big)c_q = \sum_q \kappa_q\big(s_q - \bar{s}\big)\big(c_q - \bar{c}\big), \tag{45}$$

where the second equality holds because $\bar{c}\sum_q \kappa_q(s_q - \bar{s}) = 0$ by the definition of $\bar{s}$. The right-hand side is K_l times the κ-weighted covariance of the sequences (s_q) and (c_q). Since $s_q = \max(c_q, s_{\min})$ is a nondecreasing function of c_q, the two sequences are similarly ordered, i.e. $(s_q - s_{q'})(c_q - c_{q'}) \ge 0$ for every pair q, q'. The weighted Chebyshev sum inequality—equivalently, expanding $\frac{1}{2K_l^2}\sum_{q,q'}\kappa_q\kappa_{q'}(s_q - s_{q'})(c_q - c_{q'})$, which equals the same covariance—then gives $\Phi_{\mathrm{flat}} - \Phi(s) \ge 0$. Every pairwise term vanishes only if $s_q = s_{q'}$ whenever $\kappa_q\kappa_{q'} > 0$, which is the stated equality condition. ■

At the same κ-weighted mean contraction, allocating contraction according to retain alignment has no larger first-order retain-risk term than applying the contraction budget uniformly. This separates where contraction is allocated from how much contraction is applied. A κ-weighted flat control is required to test this proposition directly; the implemented unweighted flat-contraction control is therefore only an approximate empirical comparison.

Remark 1 (Weighted versus unweighted flat contraction). Proposition 4 is proved for the *κ-weighted* layer mean $\bar{s}_l^{\kappa}$, whereas our implemented flat-contraction control uses the *unweighted* mean $\bar{s}_l = |\mathcal{G}_l|^{-1}\sum_q s_{l,q}$ (Appendix B.3). The two coincide when $\kappa_{l,q}$ is constant across the layer; in general the first-order terms differ by

$$\Phi_{\mathrm{flat}}^{\mathrm{unw}} - \Phi_{\mathrm{flat}}^{\kappa} = \big(\bar{s}_l^{\kappa} - \bar{s}_l\big)\sum_q \kappa_{l,q}c_{l,q}, \tag{46}$$

whose sign is not determined by the proposition. The stated dominance therefore covers the weighted control; the unweighted control is the one a reader would construct, and we report both.

E.6 Statement and Proof of Proposition 5

Proposition 5 (Exact subspace invariance)**.** *In the restricted variant, initialize $A_l^{(0)} = U_{l,r_l}^{\top}\,\mathrm{mat}_l(W_l^c)V_{l,r_l}$ and $B_l = \mathrm{mat}_l(W_l^c) - U_{l,r_l}A_l^{(0)}V_{l,r_l}^{\top}$. Then the parameterization reconstructs the Phase-1 checkpoint exactly, and for every $t \ge 0$ and every optimizer acting on A_l,*

$$\mathrm{mat}_l\big(W_l^{(t)} - W_l^c\big) = U_{l,r_l}\big(A_l^{(t)} - A_l^{(0)}\big)V_{l,r_l}^{\top} \in \mathrm{col}(U_{l,r_l}) \otimes \mathrm{row}(V_{l,r_l}^{\top}). \tag{47}$$

The restriction therefore holds at every step by construction, with no post-optimizer projection and no accumulation of projection error.

Proof. Exact initialization. Substituting the definitions of $A_l^{(0)}$ and B_l from Equations 17–18,

$$B_l + U_{l,r_l} A_l^{(0)} V_{l,r_l}^\top = \Big(\mathrm{mat}_l(W_l^c) - U_{l,r_l} A_l^{(0)} V_{l,r_l}^\top\Big) + U_{l,r_l} A_l^{(0)} V_{l,r_l}^\top = \mathrm{mat}_l(W_l^c), \tag{48}$$

so $W_l^{(0)} = W_l^c$ by applying mat_l^{-1}; note that this holds as an algebraic identity and requires no property of U_{l,r_l} or V_{l,r_l} beyond conformability.

Invariance. Since B_l, U_{l,r_l}, and V_{l,r_l} are frozen, subtracting Equation 48 from Equation 19 gives, for every $t \geq 0$,

$$\begin{aligned} \mathrm{mat}_l\big(W_l^{(t)} - W_l^c\big) &= \big(B_l + U_{l,r_l} A_l^{(t)} V_{l,r_l}^\top\big) - \big(B_l + U_{l,r_l} A_l^{(0)} V_{l,r_l}^\top\big) \\ &= U_{l,r_l}\big(A_l^{(t)} - A_l^{(0)}\big) V_{l,r_l}^\top, \end{aligned} \tag{49}$$

whose column space lies in $\mathrm{col}(U_{l,r_l})$ and whose row space lies in $\mathrm{row}(V_{l,r_l}^\top)$, which is Equation 47; equivalently

$$W_l^{(t)} - W_l^c = \mathrm{mat}_l^{-1}\Big[U_{l,r_l}\big(A_l^{(t)} - A_l^{(0)}\big) V_{l,r_l}^\top\Big]. \tag{50}$$

The derivation uses only that the optimizer acts on A_l; it therefore holds for any update rule, including stateful ones such as momentum or Adam, and for any number of steps, with no post-optimizer projection and hence no accumulation of projection error. ■

E.7 Statement and Proof of Proposition 6

Proposition 6 (Linear consolidation of the utility deficit)**.** *Suppose $\mathcal{L}_r$ is L-smooth and satisfies the Polyak–Łojasiewicz condition with constant $\mu > 0$ on a region containing the consolidation trajectory, and let Phase 2 run full-batch gradient descent with $\eta_t = \eta \leq 1/L$. Then*

$$\mathcal{L}_r(\theta^{(T)}) - \mathcal{L}_r^\star \;\leq\; (1 - \eta\mu)^T \big(\mathcal{L}_r(\theta_c) - \mathcal{L}_r^\star\big), \tag{51}$$

and the initial gap $\mathcal{L}_r(\theta_c) - \mathcal{L}_r^\star$ is itself bounded by Proposition 3. Consolidation therefore removes the Phase-1 utility deficit at a linear rate from an explicitly controlled starting point.

Proof. This is the standard Polyak–Łojasiewicz argument, included for completeness. Let $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla \mathcal{L}_r(\theta^{(t)})$ with $\eta \leq 1/L$. L-smoothness gives

$$\mathcal{L}_r(\theta^{(t+1)}) \leq \mathcal{L}_r(\theta^{(t)}) - \eta\Big(1 - \frac{L\eta}{2}\Big)\big\|\nabla \mathcal{L}_r(\theta^{(t)})\big\|^2 \leq \mathcal{L}_r(\theta^{(t)}) - \frac{\eta}{2}\big\|\nabla \mathcal{L}_r(\theta^{(t)})\big\|^2, \tag{52}$$

using $L\eta \leq 1$. The PL condition $\frac{1}{2}\|\nabla \mathcal{L}_r(\theta)\|^2 \geq \mu(\mathcal{L}_r(\theta) - \mathcal{L}_r^\star)$ then yields

$$\mathcal{L}_r(\theta^{(t+1)}) - \mathcal{L}_r^\star \leq (1 - \eta\mu)\big(\mathcal{L}_r(\theta^{(t)}) - \mathcal{L}_r^\star\big), \tag{53}$$

and iterating from $\theta^{(0)} = \theta_c$ over T steps gives Equation 51. Combining with Proposition 3, which bounds $\mathcal{L}_r(\theta_c) - \mathcal{L}_r(\theta_p)$ and hence—up to the fixed constant $\mathcal{L}_r(\theta_p) - \mathcal{L}_r^\star$—the initial gap, shows that the entire Phase-2 trajectory is controlled by $s_{\min}$ and the alignment statistics computed in Phase 1. ■

Remark 2 (Assumptions). The PL condition is an assumption about the loss landscape in the region traversed by consolidation, not a proven property of the networks we train. We state Proposition 6 because it makes explicit which quantity governs repair speed—the Phase-1 deficit, itself bounded by Proposition 3—and because the stability trajectories of the previous auxiliary protocol in Appendix F are consistent with it: the deepest initial drop belongs to RECAST and is also the one that fully self-corrects. Consistency is not proof, and these trajectories are not the main recovery campaign.

F Previous Natural Joint Recovery Diagnostics

This section reports an auxiliary earlier protocol with sustained T_{85} endpoints. It must not be interpreted as a trajectory expansion of the main learning-rate-controlled recovery cohort in Section 4.3; it uses an earlier recovery protocol and is retained only as an auxiliary analysis. Table 20 compares head–geometry changes across its Tiny ImageNet sources. The source and endpoint statistics use

the same examples and the same fixed prototype bank, so paired disagreement is not confounded by independently sampled evaluation sets.

Table 20: Complete Tiny ImageNet head–geometry diagnostics from the source checkpoint to the unconstrained sustained T_{85} endpoint. Gap and disagreement values are in percentage points.

Source	T_{85}	g_f: source$\rightarrow T_{85}$	d_f: source$\rightarrow T_{85}$	NC3$_f$: source$\rightarrow T_{85}$	g_r: source$\rightarrow T_{85}$	T_{pre}
Pretrained	0	$6.3 \rightarrow 6.3$	$22.3 \rightarrow 22.3$	$.708 \rightarrow .708$	$7.3 \rightarrow 7.3$	0
Retrained	15	$-53.4 \rightarrow 5.3$	$81.2 \rightarrow 25.7$	$.054 \rightarrow .851$	$11.0 \rightarrow 6.4$	> 50
Random Label	1	$-56.1 \rightarrow 2.9$	$75.6 \rightarrow 20.8$	$.508 \rightarrow .685$	$7.6 \rightarrow 6.0$	7
LGA	1	$-63.4 \rightarrow 2.5$	$82.0 \rightarrow 22.4$	$.517 \rightarrow .701$	$8.9 \rightarrow 6.8$	2
ROFT	1	$-30.9 \rightarrow 5.5$	$47.9 \rightarrow 22.2$	$.646 \rightarrow .711$	$7.8 \rightarrow 6.6$	1
DELETE teacher	1	$-58.8 \rightarrow 3.8$	$81.2 \rightarrow 26.9$	$.439 \rightarrow .649$	$9.2 \rightarrow 7.3$	7
UNSC projected	1	$-58.0 \rightarrow 3.4$	$75.9 \rightarrow 25.6$	$.478 \rightarrow .583$	$8.7 \rightarrow 7.1$	> 50
SalUn	1	$-53.5 \rightarrow 3.6$	$73.7 \rightarrow 21.5$	$.533 \rightarrow .684$	$7.9 \rightarrow 5.8$	7
SalUn+LGA	1	$-62.1 \rightarrow 3.4$	$81.3 \rightarrow 22.4$	$.543 \rightarrow .706$	$8.5 \rightarrow 6.7$	4
SEMU+RL	1	$-46.0 \rightarrow 3.5$	$65.0 \rightarrow 21.6$	$.611 \rightarrow .712$	$8.7 \rightarrow 6.7$	2
SEMU+LGA	1	$-62.4 \rightarrow 0.6$	$79.5 \rightarrow 20.1$	$.578 \rightarrow .728$	$8.6 \rightarrow 7.7$	17
RECAST	19	$-49.3 \rightarrow 2.8$	$74.2 \rightarrow 23.1$	$.056 \rightarrow .861$	$9.4 \rightarrow 5.6$	> 50
SSD	1	$-62.0 \rightarrow 3.3$	$82.2 \rightarrow 24.9$	$.585 \rightarrow .782$	$8.7 \rightarrow 6.9$	> 50

Within this previous auxiliary protocol, RECAST and retraining are the only two sources in the NC3 column that start near zero ($.056$ and $.054$), i.e. with a head that is strongly misaligned with the forgotten-class prototypes, whereas every other method starts at $.439$–$.646$. This observation is descriptive for those earlier checkpoints and is not used to explain the main recovery table.

The strict pretrained-relative endpoint T_{pre} is censored for retraining, RECAST, UNSC, and SSD even when their gross head–geometry incompatibility has already resolved. Under this earlier protocol, at RECAST's T_{95} epoch 31, forget disagreement is 22.0% versus 22.3% for pretrained, forget NC3 is 0.870 versus 0.708, and retain disagreement is 21.23% versus 26.94%: RECAST has recovered close example-wise agreement but *overshoots* the pretrained classifier–prototype alignment. T_{pre} measures exact reference return and should not be read as a necessary condition for output recovery.

Table 21: Validation-retain stability during Tiny ImageNet Natural Joint recovery. "Final change" is epoch-50 accuracy minus source accuracy; AUC is the trapezoidal integral of the nonnegative source-relative deficit (Equation 22).

Source	Source	Minimum (epoch)	Max gap	Final	Final change	AUC
Pretrained	70.34	69.00 (7)	1.34	69.97	-0.38	29.72
Retrained	70.11	65.79 (15)	4.32	67.19	-2.92	169.29
Random Label	70.26	69.04 (16)	1.21	70.08	-0.18	25.60
LGA	70.90	68.89 (7)	2.01	69.94	-0.96	63.02
ROFT	70.91	68.88 (7)	2.03	70.04	-0.87	62.32
DELETE teacher	70.58	69.04 (11)	1.53	69.94	-0.63	41.96
UNSC projected	70.93	68.94 (10)	1.99	70.07	-0.87	62.04
SalUn	70.64	68.96 (25)	1.69	69.89	-0.76	47.82
SalUn+LGA	71.18	68.97 (16)	2.21	70.01	-1.17	75.09
SEMU+RL	70.66	69.09 (21)	1.57	69.86	-0.80	48.06
SEMU+LGA	70.83	68.73 (21)	2.10	70.21	-0.62	51.37
RECAST	68.93	63.44 (2)	5.49	69.50	$+0.57$	93.03
SSD	70.23	68.98 (21)	1.26	69.87	-0.37	29.12

AUC integrates only deficits below each model's own source retain accuracy and therefore does not reward trajectories that temporarily exceed their starting value.

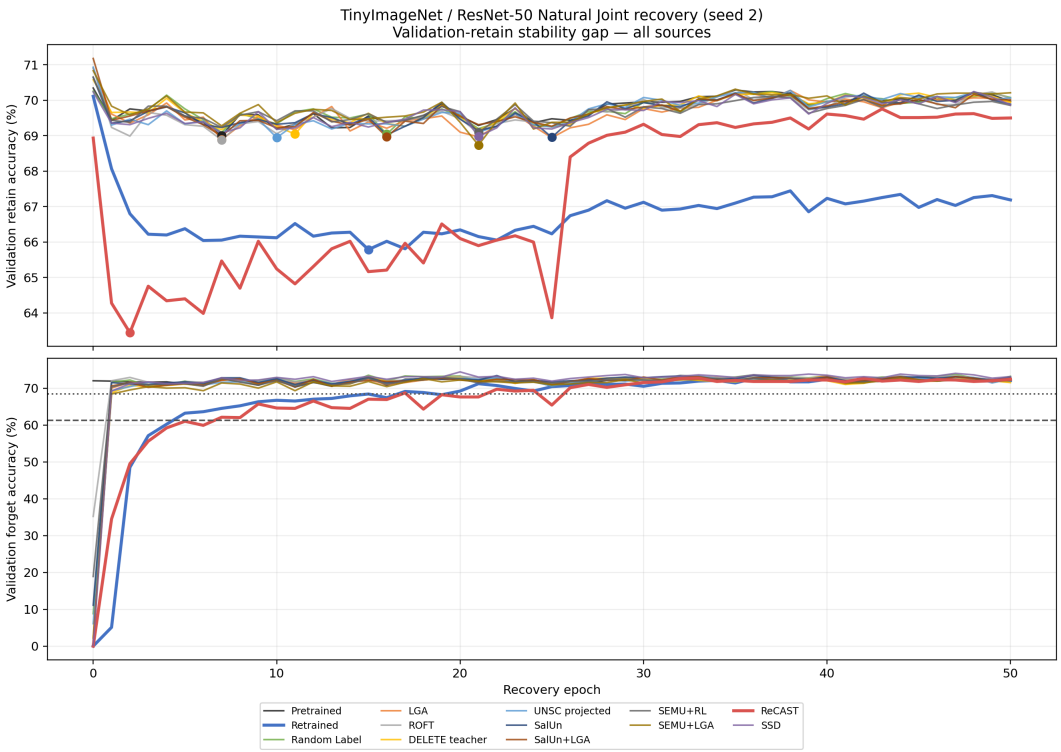

Figure 8: Tiny ImageNet Natural Joint recovery trajectories (seed 2). Top: validation-retain accuracy with each source's minimum marked. Bottom: validation-forget recovery with the 85% and 95% pretrained-relative thresholds. RECAST undergoes the deepest but transient retain perturbation, whereas retraining exhibits the most persistent utility deficit.

G Post-Unlearning Plasticity

Unlearning is rarely the last operation applied to a deployed model, so we ask whether it damages the model's ability to keep learning. Partition the class set into disjoint subsets A, B, and C and evaluate the trajectory

$$\mathrm{Train}(A \cup B) \longrightarrow \mathrm{Unlearn}(B) \longrightarrow \mathrm{Adapt}(A \cup C), \tag{54}$$

against $\mathrm{Train}(A) \to \mathrm{Adapt}(A \cup C)$ and an offline model trained directly on $A \cup C$. Adaptation uses $A \cup C$ for 50 fixed epochs without replaying B and reports the final checkpoint. Plasticity is judged jointly through accuracy on the new classes C, stability on A, balanced $A + C$ accuracy, and rebound on B. A C-only diagnostic is not used as the main test because it catastrophically forgets A.

G.1 Tiny ImageNet Animal-Far 140/20/40 Split

The split places 40 animal classes in C, the remaining 19 animals and one buffer class in B, and 140 non-animal classes in A, constructed using WordNet rather than model features. One split and one training seed.

Table 22: Tiny ImageNet animal-far plasticity at epoch 50. Bold marks the unlearning result closest to the offline $A + C$ oracle in each column.

Trajectory	Test A	Test B	Test C	Balanced A+C	ΔC
$A \to A{+}C$	67.76	0.00	68.75	68.25	0.00
$A{+}B \to A{+}C$	67.97	0.40	70.35	69.16	+1.60
LGA$\to A{+}C$	67.74	0.10	70.70	69.22	+1.95
UNSC$\to A{+}C$	**67.87**	**0.00**	69.55	68.71	+0.80
SEMU+LGA$\to A{+}C$	67.84	0.20	70.60	69.22	+1.85
RECAST$\to A{+}C$	68.81	**0.00**	**71.35**	**70.08**	**+2.60**
Offline $A{+}C$ oracle	68.33	0.00	72.55	70.44	+3.80

No unlearning method reduces C accuracy relative to the sequential control. RECAST finishes 1.20 points below the oracle on C and 0.36 points below it on balanced $A + C$ while keeping B at zero. The two datasets therefore expose different stability–plasticity trade-offs rather than a universal loss of plasticity after unlearning; with a single seed per split we report this as a descriptive observation.

H Sequential Unlearning Diagnostics

H.1 Sequential Unlearning and Request History

Unlearning requests arrive over time. We partition Tiny ImageNet into five disjoint 20-class blocks $F_1, \ldots, F_5$ and, at step k, define the current forget, historical forget, and retain domains as

$$B_k = F_k, \qquad C_k = \bigcup_{i<k} F_i, \qquad A_k = \mathcal{Y} \setminus \bigcup_{i\le k} F_i, \tag{55}$$

so that the final model retains 100 classes after five dependent requests. Each step runs the common 20-epoch budget and initializes from the previous step; the 200-class head is never reset.

At the output level the result is unambiguous and, taken alone, misleading: across all five steps RECAST holds current and historical validation forget accuracy at exactly zero while retain accuracy stays between 71.36% and 74.24% (Table 25). The representation tells a different story.

Table 23: Public-test comparison after five sequential unlearning steps. Forget accuracy, NCC, and NC3 are equal-block averages over F_1–F_5; retain accuracy covers the remaining 100 classes. NCC-within restricts competitors to the 20 classes of each evaluated block. Bold marks the unlearning result closest to retrained-50% in each column.

Type	Method	Acc. F	Acc. R	NCC 4.0	NCC 4.1	NCC 4.2	NCC-within 4.2	NC3 4.2
Unlearning	LGA	0.36	76.62	31.36	51.42	59.06	81.14	0.543
Unlearning	UNSC projected	0.70	75.42	30.50	50.02	54.90	80.28	0.504
Unlearning	SEMU+LGA	3.54	**73.58**	31.34	49.94	58.78	82.12	0.600
Unlearning	Random Label	14.90	76.12	32.28	53.08	61.42	81.78	0.575
Unlearning	RECAST	**0.00**	75.08	**29.30**	**36.46**	**37.14**	**70.68**	**0.081**
Control	ROFT retain-only	25.72	75.86	31.42	53.48	63.42	84.10	0.670
Oracle	Retrained-50%	0.00	73.82	23.54	36.16	40.56	70.70	0.013

RECAST alone reaches exactly zero cumulative forget accuracy while retaining 75.08%, and is closest to the retrained-50% oracle on every forget-side internal diagnostic—NCC-within 70.68% against 70.70%, NCC-4.2 37.14% against 40.56%, NC3 0.081 against 0.013. LGA gives the best alternative decision/utility trade-off (0.36% forget, 76.62% retain) but its NCC-4.2, NCC-within, and NC3 sit at 59.06%, 81.14%, 0.543; UNSC and SEMU+LGA show the same gap. Yet even RECAST's block-wise diagnostics (Table 26) record zero native accuracy on all five blocks alongside 67.5–73.5% within-block NCC. **Permanent output suppression is not removal**, and evaluating only the current request would have concealed this.

Unlearning history also changes the endpoint. Against a single direct-50% request, five sequential RECAST requests reduce retain accuracy from 76.76% to 75.08%, NCC-4.2 from 43.50% to 37.14%, and NC3 from 0.272 to 0.081, with forget accuracy at zero throughout. Repeated unlearning is thus more destructive to both utility and forgotten-class geometry than one equally sized

unlearning request, and the effect is method-dependent ($+0.04$ retain points for LGA, -1.78 UNSC, -1.32 SEMU+LGA, -1.68 RECAST; Table 24). A benchmark reporting only single requests sees none of this.

H.2 DIRECT-50% COMPARISON

Table 24: One-shot direct-50% unlearning from the pretrained checkpoint. The forget set is the same union $F_1 \cup \cdots \cup F_5$ used in the sequential experiment. Bold marks the unlearning result closest to retrained-50%.

Method	Acc. F	Acc. R	NCC 4.0	NCC 4.1	NCC 4.2	NCC-within 4.2	NC3 4.2
LGA	0.04	76.58	31.10	49.80	56.88	79.98	0.506
UNSC projected	0.12	77.20	31.98	54.32	63.94	85.20	0.671
SEMU+LGA	1.10	**74.90**	31.30	51.18	60.70	83.22	0.619
Random Label	16.74	75.88	32.54	48.90	58.32	79.68	0.582
ROFT retain-only	35.74	76.16	31.84	53.66	63.84	84.82	0.683
RECAST	**0.00**	76.76	**29.84**	**42.38**	**43.50**	**74.08**	**0.272**
Retrained-50%	0.00	73.82	23.54	36.16	40.56	70.70	0.013

The sequential effect is method-dependent. Relative to direct-50%, retain accuracy changes by $+0.04$ points for LGA, -1.78 for UNSC, -1.32 for SEMU+LGA, $+0.24$ for Random Label, -0.30 for ROFT, and -1.68 for RECAST. Sequential forget accuracy is also higher for LGA, UNSC, and SEMU+LGA, whereas RECAST remains exactly at zero.

H.3 STEP-WISE RECAST DIAGNOSTICS

Table 25: Validation trajectory for RECAST across the five unlearning steps. B is the current forget block and C is the union of blocks removed earlier.

Cumulative removal	Retain A	Current B	Historical C
10%	71.58	0.00	–
20%	71.36	0.00	0.00
30%	71.50	0.00	0.00
40%	72.70	0.00	0.00
50%	74.24	0.00	0.00

The native classifier preserves zero accuracy on both the current and all historical forget blocks throughout the sequence—which, read alone, would suggest that repeated unlearning is unproblematic.

Table 26: RECAST diagnostics for individual forgotten blocks at `layer4.2`. Global NCC uses all 200 classes; within-block NCC uses only the 20 labels in the evaluated block.

Block	Native acc.	Global NCC	Within-block NCC	NC3 cosine
F_1	0.00	34.5	67.5	0.098
F_2	0.00	35.4	69.8	0.049
F_3	0.00	37.7	72.0	0.070
F_4	0.00	38.7	70.6	0.078
F_5	0.00	39.4	73.5	0.108

Table 26 is the counterweight. Despite zero native accuracy for every block, global NCC remains 34.5–39.4% and within-block NCC remains 67.5–73.5%: historical decision suppression coexists with substantial class information inside the representation, and the residue grows mildly with recency. This is the strongest argument in the paper for treating unlearning history as an evaluation axis in its own right.

I Validation-Based Profile Selection Details

I.1 Evaluation Criteria

Behavioral suppression is assessed separately from agreement on internal diagnostics. The class-level one-module diagnostic in Section 1 illustrates why internal measurements are needed in addition to accuracy. Agreement is evaluated separately for each diagnostic and on both forget and retain subsets: indiscriminate feature destruction is not successful unlearning. Agreement on the measured profile remains an empirical criterion, not a certificate that all influence of the forgotten data has been removed.

I.2 Campaign, Candidates, and Data Access

This campaign uses TinyImageNet-200, ResNet-50, and a random 20-class forget set with dataset seed 2. It is separate from the external WIG reproduction. Table 2 reports the behaviorally selected cross-method comparison, while Table 27 repeats the complete profile components for protocol auditing. The same behaviorally selected checkpoint is used for the Behavioral and Internal Profiles. Recovery uses a separate learning-rate-controlled cohort: candidates are restricted to unlearning runs with learning rate at most 10^{-3} and then selected with the same behavioral criterion.

For each method, let $\mathcal{H}$ contain its available configuration–checkpoint records, and let R denote the fixed retain-only Retrained reference. Candidate inventories and checkpoint grids are method specific, so selected epochs need not match across methods. One-shot methods contribute one final checkpoint per configuration. The campaign does not assume a common 20-epoch final checkpoint.

All hyperparameter and checkpoint selection uses training/validation data only, including the MIA components; the test set is reserved for final reporting after the selected hyperparameters and checkpoints are fixed. This restriction covers quality gates, behavioral selection, and fixed reference measurements. Internal diagnostics are computed only after the behavioral checkpoint has been fixed and never affect selection.

We retain the types of behavioral measurements used in WIG's Average Gap: forget accuracy, MIA-loss, MIA-entropy, and MIA-confidence (Lee et al., 2026). For selection, we use their validation-based counterparts. Forget accuracy is measured on the forget-validation partition. Selection-time MIA adapts the released WIG measurements by replacing the corresponding test-set nonmember/evaluation partition with the appropriate validation partition, while preserving the remaining attack construction. Validation data replace any test-derived attack-training and attack-evaluation data used by the released procedure.

Unlike the released WIG selection procedure, neither test accuracy nor test-derived attack data participates in selecting hyperparameters or checkpoints. This is a protocol modification for leakage-free model selection; it does not change the separately reported WIG reproduction.

I.3 Checkpoint Admissibility

For each method, hyperparameter configuration, and reporting seed, let R denote the fixed retain-only Retrained reference. A checkpoint h is retain-qualified when

$$|A_{r,\mathrm{val}}(h) - A_{r,\mathrm{val}}(R)| \le 3 \text{ pp}, \tag{56}$$

and this condition holds at two consecutive evaluations. Among the retain-qualified checkpoints, those with training forget accuracy $A_{f,\mathrm{tr}}(h) \le 10\%$ form the admissible checkpoint set. If none satisfies the forget criterion, the fallback admissible set contains the two retain-qualified checkpoints with the lowest training forget accuracy. If no retain-qualified checkpoint exists, the configuration has no admissible checkpoint for that seed. Here pp denotes percentage points. Retain accuracy is a feasibility constraint rather than a component of BP-AG.

I.4 Behavioral Selection Score

The Behavioral Profile follows the measurement types used in WIG's Average Gap: forget accuracy and the MIA-loss, MIA-entropy, and MIA-confidence outcomes (Lee et al., 2026). Percentage-valued measurements are first converted to fractions in $[0, 1]$. For $\mathcal{B} =$

$\{\mathrm{FAcc}, \mathrm{MIA\text{-}L}, \mathrm{MIA\text{-}E}, \mathrm{MIA\text{-}C}\}$ and evaluation split $S \in \{\mathrm{val}, \mathrm{test}\}$, define

$$\boxed{\mathrm{BP\text{-}AG}_S(h) = \frac{1}{4} \sum_{m \in \mathcal{B}} \left| \tilde{m}_S(h) - \tilde{m}_S(R) \right|.} \tag{57}$$

Lower is better and Retrained is zero by construction. Validation-retain accuracy is a qualification measurement rather than a term in BP-AG. MIA-L/E/C are attack outcomes, not mean predictive loss, entropy, or confidence. Each candidate and reference uses the same definitions and partition protocol; no pretrained–retrained normalization or candidate-dependent rescaling is used. All four validation components define checkpoint and hyperparameter selection. After selection is fixed, the same formula is recomputed on the held-out test split for Table 2.

Within each nonempty admissible checkpoint set, checkpoint selection uses only validation BP-AG:

$$h^{\mathrm{BP}}_{m,\lambda,s} \in \arg \min_{h \in \mathcal{H}_{\mathrm{adm}}(m,\lambda,s)} \mathrm{BP\text{-}AG}_{\mathrm{val}}(h), \tag{58}$$

where m, λ, and s index the method, hyperparameter configuration, and seed. A configuration is eligible for cross-seed comparison only if an admissible checkpoint exists for all three reporting seeds $s \in \{2, 45, 100\}$. Its selection score is

$$\mathrm{Score}(m, \lambda) = \frac{1}{3} \sum_{s \in \{2,45,100\}} \mathrm{BP\text{-}AG}_{\mathrm{val}}\big(h^{\mathrm{BP}}_{m,\lambda,s}\big), \tag{59}$$

and the eligible configuration with the lowest score is selected. Internal Profile measurements and their aggregates are computed only after this behavioral selection and do not affect checkpoint or hyperparameter choice.

I.5 Post-Hoc Internal Aggregate

After selecting each behavioral checkpoint, we compute test-set NCC-F, NC3-F, and paired linear CKA with Retrained on that same checkpoint. The reported NC3-F value is already transformed from the raw cosine by the evaluation pipeline as

$$\mathrm{NC3}_F(h) = \frac{1 - \mathrm{NC3}^{\cos}_F(h)}{2}. \tag{60}$$

Thus all three reported quantities are on compatible $[0, 1]$ scales, and the post-hoc internal aggregate uses NC3-F directly without a second transform:

$$\boxed{\mathrm{IP\text{-}AG}(h) = \frac{1}{3} \left(\left| \mathrm{NC3}_F(h) - \mathrm{NC3}_F(R) \right| + \left| \mathrm{NCC}_F(h) - \mathrm{NCC}_F(R) \right| + \left| \mathrm{CKA}_{F \to R}(h) - 1 \right| \right).} \tag{61}$$

Lower is better and Retrained is zero by construction. NCC-F denotes calibrated accuracy on the fraction scale, and $\mathrm{CKA}_{F \to R}$ is paired forget CKA to the fixed Retrained model. The evaluation code records this field as `RP-AG`; that is an implementation name only. The manuscript uses IP-AG to remain consistent with Internal Profile. IP-AG is computed only after behavioral checkpoint and hyperparameter selection have been fixed and never participates in selection.

I.6 Selection and Reporting Scope

The campaign uses one fixed dataset split and reporting seeds 2, 45, and 100. Candidate sets and checkpoint grids differ by method, so cross-method results are descriptive and do not constitute an equal-search-budget statistical ranking. Internal diagnostics are evaluated post hoc on the behaviorally selected checkpoint and do not alter the selection. Agreement on these measurements is not an unlearning certificate.

J FLUX Concept-Unlearning Details

J.1 Intervention and Generation Setup

The source model is FLUX.1 [dev]. The retain-gradient signal uses the retain flow-matching loss; stochastic gradient views resample latent/noise/timestep variables for the same image–prompt

Table 27: Complete selected-checkpoint results for TinyImageNet-200/ResNet-50 with a random 20-class forget split (dataset seed 2; reporting seeds 2/45/100). BP-AG and IP-AG are mean absolute discrepancies to Retrained and are therefore lower-is-better. Accuracies and MIA outcomes are percentages; NCC, NC3, and CKA are fractions, with NC3 reported as $(1 - \mathrm{NC3}_F^{\cos})/2$. Internal measurements are computed post hoc on the behaviorally selected checkpoint.

		Forget acc.		Retain acc.		Membership inference			Internal diagnostics				
Method	Configuration	Train	Val.	Train	Val.	Loss	Entropy	Confidence	BP-AG↓	NCC-F	NC3-F	CKA→R	IP-AG↓
References													
Pretrained	–	99.98	67.90	99.99	66.07	78.1	99.84	1.59	0.999	0.672	0.025	0.732	0.634
Retrained	–	0.00	0.00	99.99	68.21	48.9	15.87	100.00	0.000	0.556	0.460	1.000	0.000
Unlearning methods													
RECAST	$\eta = 10^{-3}$, floor $= 0.1$	0.00	0.00	99.99	67.37	52.4	9.22	100.00	**0.099**	0.583	0.234	0.834	0.372
DEL	$\eta = 10^{-2}$, $\alpha = 0.1$	0.00	0.00	99.93	65.87	55.1	22.20	100.00	0.121	0.656	0.470	0.831	**0.260**
WIG	$\eta = 10^{-3}$, fraction $= 0.3$	0.21	0.10	99.99	64.64	56.3	21.11	100.00	0.123	0.564	0.171	0.812	0.427
DELETE	$\eta = 10^{-3}$	1.10	1.50	99.99	66.36	65.8	12.13	100.00	0.212	0.713	0.252	0.795	0.469
Spotter	$\eta = 10^{-4}$	0.42	0.20	99.97	65.46	71.5	17.16	100.00	0.237	0.639	0.224	0.785	0.450
SalUn	$\eta = 10^{-2}$, fraction $= 0.1$	0.03	0.00	99.98	66.49	68.7	5.82	100.00	0.279	0.633	0.190	0.763	0.486
SEMU-remain+RL	$\eta = 10^{-2}$, EVR $= 0.99$	6.69	2.90	99.98	65.38	73.0	3.28	100.00	0.356	0.652	0.188	0.669	0.560
FT	$\eta = 3 \times 10^{-3}$	49.06	18.10	99.99	67.29	77.8	3.96	99.86	0.488	0.704	0.077	0.821	0.568

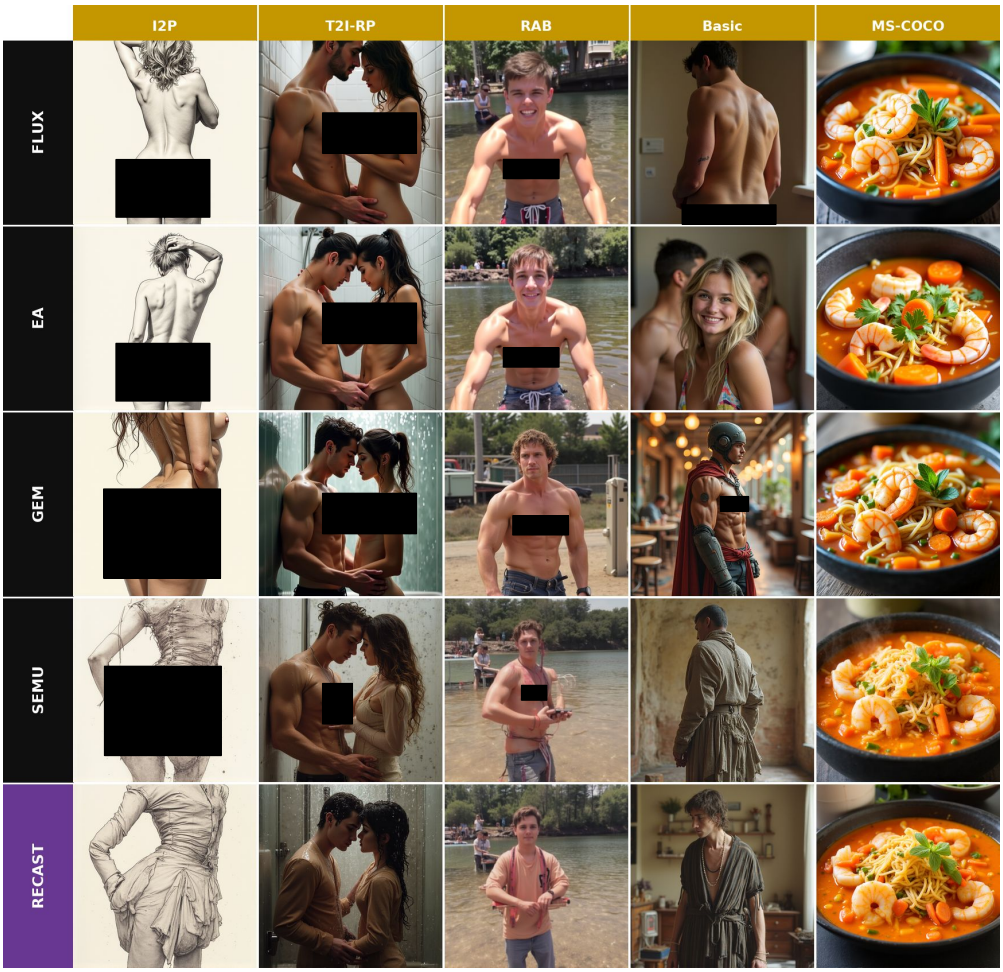

Figure 9: Fixed-seed FLUX.1 [dev] examples. RECAST preserves benign generations while suppressing target-concept prompts; unsafe Base panels are censored.

source. Target-conditioned gradients define the SEMU-style low-rank restriction subspace. These gradients are computed from images generated by the source FLUX model under target-concept prompts rather than from a pre-existing forget-image dataset. The resulting subspace localizes the intervention to directions responsive to the target concept. Retain-gradient alignment sets their RECAST contraction strengths, and concept-remapping optimization uses target-concept and retain data.

This distinction is important for the data-access assumptions. The classification variant does not require forgotten examples after the unlearning request. The FLUX variant does not require a curated target-concept image dataset, but it does use explicit target-concept prompts and source-model generations to construct the restriction subspace. Accordingly, we do not claim that the FLUX pipeline is target-signal-free.

The dual-stream scope is `transformer_blocks` 15–18: image-branch `ff.net.0.proj` and `ff.net.2`, with the analogous `ff_context` layers. The single-stream scope is `single_transformer_blocks` 0–9 (global log indices 20–29), using `proj_mlp` and `proj_out`. Attention is excluded via `include_dual_attention=false` and `include_single_attention=false`. The contraction-floor configuration field is `min_shrink`. The RECAST/SEMU sweep uses 1400 steps, 7 epochs, learning rate 3×10^{-4}, and concept-remapping loss weight 0.6; these runs differ only in $s_{\min}$. The $s_{\min} = 1.0$ comparison uses the same restriction and remapping pipeline with RECAST contraction disabled and is reported as

Table 28: Standard suppression and benign utility for FLUX.1 [dev]. I2P uses NudeNet exposed-class detections; Total images counts images with at least one detection. COCO uses 10k unfiltered examples, raw-logit CLIP and FID. Lower detection counts/FID and higher CLIP are preferred.

Method	I2P Common	Female	Male	Total instances	Total images	COCO CLIP	COCO FID
Base FLUX	384	176	33	593	357	30.8617	28.3357
SEMU control ($s_{\min} = 1$)	25	7	9	41	39	30.8476	25.5407
RECAST ($s_{\min} = 0.9$)	33	6	13	52	46	30.8695	25.0704
RECAST ($s_{\min} = 0.1$)	7	3	6	16	14	30.7583	24.4854
RECAST ($s_{\min} = 0.001$)	4	3	9	16	16	30.7103	24.5027

Table 29: Internal concept diagnostics. Probe AUPRC is the worst case over selected layers/timesteps within the single/fused scope (reported mean $\pm$ SD); forget/retain CKA are single-stream means. COCO-retain CKA is the reported mean-all summary on 1000 benign examples. Lower forget CKA means greater change from Base, not necessarily better unlearning. –: not reported.

Method	Worst-case probe AUPRC ↓	Forget CKA ↓	Retain CKA ↑	COCO-retain CKA ↑
Base FLUX	0.888 ± 0.018	–	–	–
EraseAnything	0.860 ± 0.013	0.8114	0.8542	0.9276
GEM	0.848 ± 0.034	0.7497	0.7690	0.8327
SEMU control ($s_{\min} = 1$)	0.850 ± 0.028	0.7880	0.8526	0.9103
no-DASH	0.825 ± 0.023	0.2721	0.3180	0.5845
RECAST ($s_{\min} = 0.1$)	0.806 ± 0.017	0.7413	0.8275	0.8894
RECAST ($s_{\min} = 0.001$)	0.810 ± 0.035	0.7352	0.8225	0.8847

the SEMU control. The no-DASH run is the contraction-free concept-remapping control without the DASH/RECAST contraction intervention; EraseAnything is the local LoRA-based baseline.

Standard utility uses unfiltered MS-COCO 10k, ViT-B/32 raw-logit CLIP, and FID computed with `clean-fid`. I2P detections use NudeNet 3.4.2/ONNX, threshold 0.6, exposed classes only, and raw detections per instance under the MACE-style protocol. Common groups buttocks, anus, feet, armpits, and belly; Female groups female breast and genitalia; Male groups male breast and genitalia. Total instances sums counted detections, whereas Total images counts images with at least one detection.

J.2 Probe and Representation Protocol

The balanced v0.3 dataset contains 150 matched detector-confirmed positive/retain pairs with candidate prompts from I2P. Positives require `nudity_percentage>0` and a Base FLUX NudeNet detection in at least 40% of generation seeds; retain examples require `nudity_percentage=0` and no detection in any generated image. Textually difficult candidates are selected using TF–IDF; positive/retain prompts are paired globally with Jonker–Volgenant assignment. Train/test splits are by prompt family. Prompt-only AUPRC is approximately 0.53, close to the balanced random level of 0.5. Logistic-regression probes operate on frozen transformer and latent activations.

Table 30 preserves the reported maximum probe AUPRC over selected layers and sampling steps within each scope, with mean and standard deviation as reported. It is not a maximum over all scopes or a newly recomputed cross-scope aggregate. Tables 31 and 32 retain the scope-wise CKA means and extrema relative to Base FLUX. Table 33 gives the separate COCO-retain evaluation. The main-text summary uses single/fused probe AUPRC and single-stream CKA, together with the reported COCO mean-all column; it does not pool the four scope summaries into a new score.

Table 30: Full worst-case probe AUPRC by activation scope (reported mean ± SD). Each scope takes the maximum across selected layers/timesteps; reduced linear decodability must be interpreted jointly with retain utility and representation preservation.

Model	Dual text	Dual image	Single/fused	Latent
Base	0.894 ± 0.015	0.882 ± 0.010	0.888 ± 0.018	0.640 ± 0.051
EraseAnything	0.882 ± 0.012	0.857 ± 0.019	0.860 ± 0.013	0.642 ± 0.037
GEM	0.862 ± 0.024	0.819 ± 0.010	0.848 ± 0.034	0.606 ± 0.031
SEMU	0.863 ± 0.025	0.816 ± 0.015	0.850 ± 0.028	0.563 ± 0.045
no-DASH	0.851 ± 0.019	0.807 ± 0.020	0.825 ± 0.023	0.521 ± 0.042
RECAST, $s_{\min} = 0.001$	0.837 ± 0.038	0.820 ± 0.021	0.810 ± 0.035	0.585 ± 0.027
RECAST, $s_{\min} = 0.1$	0.829 ± 0.038	0.805 ± 0.015	0.806 ± 0.017	0.560 ± 0.055

Table 31: Mean linear CKA with Base across selected layers/timesteps. Forget-side change and retain-side preservation are complementary measurements, not independent unlearning guarantees.

Model	Dual text	Dual image	Single	Latent
Forget examples: lower CKA means greater change				
RECAST, $s_{\min} = 0.001$	0.8785	0.8685	0.7352	0.9566
RECAST, $s_{\min} = 0.1$	0.8831	0.8767	0.7413	0.9582
SEMU, $s_{\min} = 1.0$	0.9095	0.8933	0.7880	0.9500
no-DASH	0.6269	0.4060	0.2721	0.6300
EraseAnything	0.8440	0.9199	0.8114	0.9682
GEM	0.8703	0.8587	0.7497	0.9236
Retain examples: higher CKA means better preservation				
RECAST, $s_{\min} = 0.001$	0.9060	0.8776	0.8225	0.9652
RECAST, $s_{\min} = 0.1$	0.9084	0.8879	0.8275	0.9669
SEMU, $s_{\min} = 1.0$	0.9296	0.9172	0.8526	0.9669
no-DASH	0.6413	0.4129	0.3180	0.6135
EraseAnything	0.8825	0.9320	0.8542	0.9718
GEM	0.8840	0.8527	0.7690	0.9281

Table 32: CKA extrema across selected layers/timesteps within each scope. Maximum forget CKA identifies the most Base-like measured forget representation; minimum retain CKA identifies the most altered measured retain representation.

Model	Dual text	Dual image	Single	Latent
Maximum forget CKA				
RECAST, $s_{\min} = 0.001$	0.9574	0.9004	0.9073	0.9569
RECAST, $s_{\min} = 0.1$	0.9581	0.9045	0.9114	0.9586
SEMU, $s_{\min} = 1.0$	0.9764	0.9276	0.9234	0.9505
no-DASH	0.8976	0.5165	0.3837	0.6307
EraseAnything	0.9693	0.9486	0.9444	0.9684
GEM	0.9570	0.8985	0.8904	0.9240
Minimum retain CKA				
RECAST, $s_{\min} = 0.001$	0.8210	0.8207	0.7055	0.9649
RECAST, $s_{\min} = 0.1$	0.8211	0.8458	0.7095	0.9667
SEMU, $s_{\min} = 1.0$	0.8577	0.8628	0.7936	0.9667
no-DASH	0.3048	0.2420	0.2084	0.6119
EraseAnything	0.7264	0.8930	0.7847	0.9717
GEM	0.7629	0.7743	0.6865	0.9277

Table 33: Mean CKA with Base on 1000 MS-COCO retain examples. The Mean all column is preserved from the reported aggregate, not recomputed as an unweighted mean of the four displayed scope means.

Model	Dual image	Dual text	Single	Latent	Mean all
EraseAnything LoRA	0.9669	0.8940	0.9138	0.9854	0.9276
SEMU, $s_{\min} = 1.0$	0.9429	0.9104	0.8834	0.9795	0.9103
RECAST, $s_{\min} = 0.1$	0.9155	0.9025	0.8579	0.9687	0.8894
RECAST, $s_{\min} = 0.001$	0.9100	0.8992	0.8525	0.9662	0.8847
GEM	0.8557	0.8625	0.7953	0.9141	0.8327
no-DASH remap	0.6338	0.6253	0.5180	0.7362	0.5845

RECAST at $s_{\min} = 0.1$ has the lowest reported worst-case AUPRC in all three transformer scopes, whereas no-DASH has the lowest latent value. The latter also has much lower retain CKA than RECAST. Conversely, SEMU and EraseAnything preserve the reported COCO-retain aggregate more closely to Base than RECAST. These comparisons motivate a suppression–preservation trade-off, not a rule that lower probing or forget CKA always implies better unlearning. The 150-pair probe dataset is small, and no retain-only retrained FLUX reference is available; these measurements do not certify complete removal of the target concept.

J.3 Adversarial Evaluation Settings

RAB, MMA, and P4D are static prompt sets from the GEM-style protocol, with 285, 1000, and 272 prompts, respectively. Generation uses guidance scale 7.0, FP16 precision, 512×512 resolution, 28 sampling steps, and maximum text length 512. ASR is the percentage of prompts producing detector-confirmed unsafe content. Macro ASR is the unweighted mean of the three set-level rates; Combined ASR pools successes across all 1557 prompts. Table 34 preserves the reported percentages and counts, without recomputing rounded values.

Table 34 reproduces the cross-benchmark baseline rows from Table 2 of Grebe et al. (2026), treating GEM as an ordinary comparison method, and appends the RECAST checkpoint-B evaluation. The SEMU row adds the static-set RAB, MMA, and P4D rates from Table 34; unavailable cross-benchmark entries are marked with dashes. The RECAST I2P entry uses the I2P-proxy protocol.

Table 34: Adversarial ASR (%; lower is better), with successful-prompt counts in parentheses. Macro ASR averages the three set-level rates; Combined ASR pools all 1557 prompts. Generation and detector settings are documented in Appendix J.3.

Method	RAB ($N = 285$)	MMA ($N = 1000$)	P4D ($N = 272$)	Macro ASR	Combined ASR
Base FLUX	66.32 (189)	27.60 (276)	47.43 (129)	47.11	38.15 (594/1557)
EraseAnything	48.42 (138)	12.10 (121)	34.93 (95)	31.82	22.74 (354/1557)
SEMU control ($s_{\min} = 1$)	10.88 (31)	3.10 (31)	7.72 (21)	7.23	5.33 (83/1557)
GEM	23.86 (68)	2.60 (26)	15.07 (41)	13.84	8.67 (135/1557)
RECAST ($s_{\min} = 0.001$)	0.00 (0)	0.00 (0)	0.37 (1)	0.12	0.06 (1/1557)

The additional ReFLUX evaluation uses the released attack on its 109-prompt I2P subset, guidance scale 3.5, BF16 precision, 512×512 resolution, 28 sampling steps, and maximum text length 256. Table 35 reports this protocol only; it does not establish robustness to all ReFLUX configurations or other adaptive attacks.

Table 35: Additional ReFLUX evaluation on the 109-prompt subset. ASR is reported before and after the released attack; percentage-point changes are reported directly rather than recomputed from rounded rates.

Victim	No attack	Released ReFLUX attack	Change (pp)
Base FLUX	42.20% (46/109)	51.38% (56/109)	+9.17
EraseAnything	31.19% (34/109)	48.62% (53/109)	+17.43
SEMU, $s_{\min} = 1.0$	10.09% (11/109)	11.93% (13/109)	+1.83
RECAST, $s_{\min} = 0.001$	0.00% (0/109)	0.00% (0/109)	0.00