The BatchNorm Illusion: Diagnosing Normalization Artifacts in Machine Unlearning Evaluation

Aaryaman Kalani[1] **Murari Mandal**[2] **Dhruv Kumar**[1] **Mohan Kankanhalli**[3] **Yash Sinha**[1]

[1]BITS Pilani [2]KIIT Bhubaneswar [3]National University of Singapore

{f20220488, dhruv.kumar, yash.sinha}@pilani.bits-pilani.ac.in

murari.mandalfcs@kiit.ac.in dcsmsk@nus.edu.sg

Abstract

Approximate machine unlearning aims to remove the influence of specific training data from a trained model without retraining from scratch. We identify a previously undocumented confound in how unlearning is evaluated on BatchNorm-based architectures: a single forward pass over retain data, an operation that modifies no weight, can deterministically rewrite the model's normalization state and reverse the apparent surface-metric forgetting. We formalize this operation as a weight-preserving fixed-point operator and prove that any pre-versus-post gap it induces is provably attributable to BN running statistics rather than to any modification the unlearning method made to the weights. This attribution claim cleanly separates measurement failure (BN artifact) from encoder failure (residual weight-encoded information, recently documented in concurrent work), and the same operator framework yields a unique decomposition of linear-probe elevation into BN-measurement-bias and encoder-geometry components. Empirically, the artifact reverses headline forget accuracy by up to 78pp across nine evaluated methods on standard benchmarks; an attacker with as few as 10 unlabeled images recovers most of the masked accuracy; and a strict GroupNorm control reduces the artifact to zero across all methods. The tested membership-inference attacks change little under recalibration, locating the observed evaluation failure in forget accuracy and linear probing. Code is available at https://github.com/aaryamam/BatchNormIllusion.

1 Introduction

Approximate machine unlearning seeks to remove the influence of specified training data from a model without retraining from scratch [3, 1]. A representative recent method [6] reports forget-class accuracy of 3.0% on CIFAR-10 after unlearning. We load the released checkpoint, change nothing in its weights, run `torch.optim.swa_utils.update_bn` on the retain set, and obtain 65.1% on the same forget class. Which number measures forgetting? The trivial operation that produced the gap is the diagnostic this paper proposes. We are not aware of any systematic report of it as an evaluation control in the unlearning literature, and the headline forgetting numbers in six of nine methods we examine are recoverable when it is applied.

The unlearning literature is structured around two ways approximate methods can fail to satisfy GDPR-style erasure obligations. The first is *encoder failure*: the model's parameters still encode the forgotten content even when classification accuracy on the forget set is near zero, as documented by relearning attacks [14, 29] and concurrent work on linear probing [8, 20]. The second is what we call *measurement failure*: the evaluation procedure reports forgetting where none has occurred. This paper is about the second. The two failure modes are complementary, not competing; for several methods they coexist.

Two phenomena underlie the artifact. *BN corruption* arises when BN is in train mode during unlearning: forget-data forward passes update running statistics through an exponential moving

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

average that lies entirely outside the gradient computation graph, so weight-space gradient projection cannot intercept it. *BN misalignment* arises when BN is frozen but the conv weights drift under projected ascent, so the activation distribution at each BN layer decouples from the still-frozen running statistics. Both surface as low forget accuracy; both vanish under recalibration.

Recalibration's attribution claim, that any pre/post gap is provably about BN running statistics rather than weights, rests on operator-theoretic properties: the operation is idempotent, weight-preserving, and pointwise equivalent to oracle population normalization at the model's existing weights. We extend the same framework to linear-probe measurements, deriving a unique decomposition of LP elevation into a BN-measurement-bias term and an encoder-geometry term. For several methods on CIFAR-100, the BN term is large and *negative*: BN measurement bias *deflates* the surface LP, masking encoder leakage that recalibration reveals. Concurrent work [8] attributing residual forget-class LP to Neural Collapse is consistent with our analysis but cannot quantify the encoder-level severity faithfully without first removing the BN measurement bias. The artifact is also adversarially exploitable: an attacker with local access to the unlearned checkpoint runs the same operation on as few as ten unlabeled images and recovers most of the masked forget-class accuracy.

Contributions. We do not propose a new unlearning method; the contribution is in the genre of measurement corrections [22, 28, 12]. **(C1)** Two mechanically opposite normalization artifacts, BN corruption and BN misalignment, mapping deterministically to BN train-vs-eval mode, with the corruption mechanism formalized as a non-interceptability result for any operator applied to gradients. **(C2)** BN recalibration as a near-zero-cost diagnostic, proven to be a deterministic weight-preserving fixed-point operator (Theorem 1), together with a uniqueness decomposition (Proposition 1) that separately identifies BN-measurement-bias and encoder-geometry components of linear-probe elevation, and shows that BN measurement bias *masks* (rather than creates) encoder leakage for several methods. **(C3)** Empirical evaluation on nine evaluated methods on CIFAR-10/100 showing that the diagnostic restores near-perfect self-consistency between forget accuracy and relearning susceptibility, and that BN vulnerability is structurally entangled with retain preservation. **(C4)** An adversarial recovery experiment converting the artifact into a security finding, plus mechanistic falsifications via GroupNorm (architecture-controlled) and LayerNorm (ViT) controls.

2 Related Work

Approximate unlearning. The unlearning problem was formalized in Cao and Yang [3]; subsequent work spans exact methods (SISA [1]) and approximate methods including data deletion [9, 17], analysis of unlearning factors [34], fine-tuning on retain [10], distillation from competent and incompetent teachers [4], Fisher-weighted forgetting [10], amnesiac learning [11], saliency-based weight erasure [6], SCRUB [18], selective synaptic dampening [7], projected gradient unlearning [13], output-distribution reweighting (RWFT [36]), and representation-level unlearning [19]. Three recent benchmarks evaluate unlearning at scale on BN-based architectures (the NeurIPS unlearning competition [35], Deep Unlearn [2], and MUBox [21]), and to our reading none explicitly reports a BN recalibration control. Recent verification methods such as UMA [37] also rely on BN backbones; our diagnostic is immediately applicable.

Other unlearning settings. Unlearning has also been studied under restricted data access and in tasks beyond image classification. Error-maximizing noise enables fast class unlearning through short impair and repair steps [33]. Zero-shot unlearning removes a class without access to the original training data [5]. Other work extends unlearning to regression [32] and to graph neural networks through knowledge distillation [31]. Further work covers multi-modal recommendation under legal, licensing, and modality constraints [30], and large language models through a multi-agent framework that needs no retraining [26]. Our diagnostic applies to any of these methods that run on a BatchNorm backbone.

Encoder-level evaluation. Gao et al. [8] use linear probes to demonstrate residual forget-class information across approximate methods, attributing the gap to feature–classifier misalignment via Neural Collapse. Lee et al. [20] show that backbone freezing followed by classifier retraining recovers forget accuracy. Relearning attacks [14, 29] establish that any unlearned representation in current methods can be cheaply re-induced by fine-tuning. These are encoder-failure findings: the model's parameters still encode the forgotten content. Our work is mechanistically distinct: it

identifies a measurement failure that operates on top of, and largely orthogonal to, encoder failure. Our decomposition (Section 3) shows that for several methods on CIFAR-100, BN measurement bias *masks* encoder leakage; the post-recalibration LP exceeds retrain by a larger margin than pre-recalibration LP could indicate. The Neural Collapse story is therefore strengthened, not refuted, by our diagnostic.

BatchNorm and recalibration elsewhere. BatchNorm's running statistics play a dual role (training-time stability [15, 25] and a checkpoint-time parametric object that determines inference behavior), which is what makes the artifact possible. Recomputing BN statistics with a target data pass is well known in other contexts: SWA [16] uses `update_bn` to correct statistics of the averaged model; test-time adaptation [27, 23] similarly realigns BN with shifted test distributions. Our contribution is not the mechanism but (i) demonstrating that its absence exposes a systematic measurement failure in unlearning benchmarks, and (ii) the operator-theoretic characterization that makes the attribution unassailable.

3 The Diagnostic and Its Properties

A model with Batch Normalization stores both learned weights and running activation statistics. Unlearning can leave these two parts of the checkpoint inconsistent: low forget-class accuracy may then reflect how the weights are normalized, rather than what they retain. Our diagnostic recomputes the running statistics from retain data at the current weights, using a single forward sweep with gradients disabled. Any resulting change in accuracy is therefore attributable to normalization state. The following results formalize this attribution and extend it to linear probing.

3.1 Two mechanistically opposite phenomena

A BN layer at position ℓ tracks running statistics $(\mu^\ell, (\sigma^\ell)^2)$ updated during forward passes in train mode via exponential moving average with momentum m:

$$\mu^\ell \leftarrow (1-m)\mu^\ell + m\hat{\mu}^\ell_{\text{batch}}, \qquad (\sigma^\ell)^2 \leftarrow (1-m)(\sigma^\ell)^2 + m\hat{v}^\ell_{\text{batch}}. \tag{1}$$

Here $\hat{\mu}^\ell_{\text{batch}}$ and $\hat{v}^\ell_{\text{batch}}$ are the batch moment estimates used for the running-buffer updates. At inference (eval mode), BN normalizes activations using $(\mu^\ell, (\sigma^\ell)^2)$ as fixed parameters.

Phenomenon 1 (BN corruption). If BN is in train mode during unlearning, every forget-data forward pass executes Eq. (1) using batch statistics drawn from $\mathcal{D}_f$. Over T unlearning steps, the running statistics drift toward the forget-class conditional distribution. Because Eq. (1) occurs in the forward pass, no operator applied to gradients can intercept this update; this includes, in particular, projection-based unlearning methods that constrain weight gradients to be orthogonal to retain-class directions. The resulting model has weights consistent with retain-class behavior but BN running statistics that absorb forget-class information; predictions on retain-class inputs are normalized through forget-shifted statistics. We give the formal statement (Theorem 2) and proof in Appendix A.

Phenomenon 2 (BN misalignment). If BN is frozen (eval mode) during unlearning, the running statistics are immutable, but the conv weights θ shift via the unlearning update $\theta \to \theta_T$. The activation distribution $\mathcal{A}^\ell(\theta_T; \mathcal{D}_r)$ produced by the new weights on retain inputs is no longer the distribution against which $(\mu^\ell, (\sigma^\ell)^2)$ were calibrated. The BN layer normalizes incorrectly, producing a measurement-time distortion that depresses forget accuracy without genuine erasure: the weights still encode the forget-class structure, but they are evaluated under stale normalization. The two phenomena are mechanically opposite: the first contaminates the *statistics* relative to fixed weights, the second contaminates the *weights* relative to fixed statistics. Both produce indistinguishable surface metrics, and both vanish under BN recalibration, which we now define.

Why the two phenomena are not interconvertible. BN corruption (Phenomenon 1) and BN misalignment (Phenomenon 2) are not endpoints of a continuum. They are produced by different control choices the unlearning method makes (BN train vs. eval), and they leave different traces in the saved checkpoint. Under Phenomenon 1, θ is approximately consistent with the retain distribution but φ has drifted; the artifact lives in the running statistics. Under Phenomenon 2, θ has drifted but φ

has not; the artifact lives in the mismatch between the two. Both are surfaced by the same diagnostic because $\mathcal{R}^*$ corrects the running statistics to be consistent with the current θ, regardless of which side of the pair was originally moved. A method can in principle exhibit both simultaneously (BN train mode for some steps, frozen for others), and the diagnostic remains a single forward pass.

3.2 The recalibration operator

The model space. A BN-equipped network is a pair $M = (\theta, \varphi)$, where θ collects all weight parameters (conv kernels, BN affine γ, β, classifier weights) and $\varphi = \{(\mu^\ell, (\sigma^\ell)^2)\}_{\ell=1}^{L}$ collects the running statistics. The forward map $f(x; \theta, \varphi)$ is a function of *both*. Standard machine-learning rhetoric treats φ as a passive bookkeeping artifact, but φ is parametric: it is part of the saved checkpoint and it determines normalization at inference. An unlearning method may modify weights without modifying φ, or modify φ without modifying weights, and forget accuracy and linear probing cannot distinguish the two cases without an explicit recalibration step. The membership-inference measurements in Appendix O are largely unchanged by the same operation.

The recalibration operator. For any model $M = (\theta, \varphi)$ and retain distribution $\mathcal{D}_r$, define

$$\mathcal{R}^*_{\mathcal{D}_r} : (\theta, \varphi) \mapsto (\theta, \varphi^*(\theta; \mathcal{D}_r)), \tag{2}$$

where $\varphi^*(\theta; \mathcal{D}_r)$ are the population activation moments at every BN layer when the model with weights θ is evaluated on inputs drawn from $\mathcal{D}_r$ under oracle layer-wise normalization. Empirically, we use $\widehat{\mathcal{R}}_{N_r,b}$: a single forward sweep over N_r retain images in train mode with gradients disabled and mini-batch size b (PyTorch's `torch.optim.swa_utils.update_bn`). Its error separates sampling variation, $O_p(N_r^{-1/2})$, from a finite-batch term, $O_L(b^{-1})$, under the regularity conditions in Appendix B; the latter constant may depend on network depth L. The population operator below is exactly idempotent. The empirical pass is also idempotent when the input tensors, batch partition, and forward computation are held fixed.

Theorem 1 (Recalibration as a fixed-point operator)**.** *$\mathcal{R}^*_{\mathcal{D}_r}$ satisfies:* ***(i)*** idempotence*: $\mathcal{R}^* \circ \mathcal{R}^* = \mathcal{R}^*$;* ***(ii)*** weight invariance*: weights θ are unchanged; pre-BN activations at the first BN layer are identical to those of M, while activations at $\ell \geq 2$ are deterministic functions of unchanged θ and updated φ^* (and may differ from M's);* ***(iii)*** pointwise oracle equivalence*: $f(x; \mathcal{R}^*(M))$ equals the network with θ and oracle population $\mathcal{D}_r$ moments at every BN layer, deterministically for every input;* ***(iv)*** empirical concentration*: under the moment and second-order regularity assumptions of Appendix B, $\widehat{\mathcal{R}}_{N_r,b}$ deviates from $\mathcal{R}^*$ by $O_p(N_r^{-1/2}) + O_L(b^{-1})$ in any fixed norm on φ, for a fixed finite-depth network and independent retain images.*

The proof and finite-batch derivation are in Appendix B. The pass changes normalization statistics but no weight, which gives the following attribution:

Corollary 1 (BN attribution)**.** *For any metric $g(M)$ depending on the forward map, the gap $g(\mathcal{R}^*(M)) - g(M)$ is provably attributable to BN state and not to weights. Equivalently, $\mathcal{R}^*$ cannot create forgetting and cannot destroy weight-encoded forgetting; it can only* unmask *which regime the model was in.*

Any metric change induced by the pass comes from normalization state, with the learned parameters fixed. "Weight invariance" refers to encoder weights θ (including BN affine γ, β), not to forward-pass features themselves; features are deterministic functions of θ and φ, so an update to φ predictably modifies them.

3.3 Decomposition of linear-probe elevation

The previous subsection frames the artifact for the surface metric (forget accuracy). We now use the operator framework to disambiguate *linear-probe* measurements, which concurrent work [8] interprets via Neural Collapse.

For an unlearned model M, let LP^M denote forget-class linear-probe accuracy, $M_{\mathrm{cal}} = \mathcal{R}^*(M)$ the recalibrated model, and M_{retr} the retrain-from-scratch reference. The next result splits the gap relative to retrain into a normalization-state contribution and a contribution that remains after recalibration.

Proposition 1 (Unique BN/NC attribution decomposition)**.** *For any M, the decomposition of the LP gap relative to retrain*

$$\mathrm{LP}^{M} - \mathrm{LP}^{M_{\mathrm{retr}}} = \underbrace{\left(\mathrm{LP}^{M_{\mathrm{cal}}} - \mathrm{LP}^{M_{\mathrm{retr}}}\right)}_{\text{NC-residual}} + \underbrace{\left(\mathrm{LP}^{M} - \mathrm{LP}^{M_{\mathrm{cal}}}\right)}_{\text{BN-residual}} \tag{3}$$

is the unique decomposition into terms satisfying (i) the BN-residual vanishes whenever $\varphi = \varphi^*(\theta; \mathcal{D}_r)$, and (ii) the NC-residual is invariant under $\mathcal{R}^*$.

The proof (Appendix C) is short: existence is the algebraic identity $A - C = (A - B) + (B - C)$; uniqueness follows by evaluating any candidate split on M_{cal} and using the two criteria. Uniqueness matters because it precludes alternative attributions consistent with the same two criteria: any decomposition that respects "BN-residual vanishes at φ^*" and "NC-residual is recalibration-invariant" must coincide with Eq. (3).

Corollary 2 (BN measurement bias is identifiable independently of encoder geometry)**.** *A nonzero BN-residual is a measurement bias attributable to the running statistics alone, not to feature–classifier misalignment in the sense of Neural Collapse. By Theorem 1 (weight invariance), $\mathcal{R}^*$ leaves encoder and classifier weights untouched, so the encoder-level forget-class structure is identical before and after recalibration; any LP change reflects a correction to the surface measurement of that fixed structure. A positive BN-residual indicates the surface LP* overstates *encoder leakage; a negative BN-residual indicates the surface LP* understates *it. The post-recalibration LP,* $\mathrm{LP}^{M_{\mathrm{cal}}}$*, is the unbiased measure of the encoder's residual forget-class linear separability.*

The operational consequence is direction-dependent. The BN-residual on CIFAR-100 for GA, SalUn, and SSD is large and *negative* (Table 2). BN measurement bias *deflates* the pre-recalibration LP, hiding encoder-level leakage that recalibration reveals. The NC-residuals for the same methods are positive and substantial: the recalibrated LP exceeds retrain by 14–17pp, indicating the encoder retains *more* forget-class structure than retrain. The Neural Collapse story of Gao et al. [8] is consistent with our analysis on these methods, but their pre-recalibration measurement could not see that the BN measurement bias was masking the encoder-level severity by a magnitude that exceeds the (uncorrupted) NC signal itself.

We caution that an additive squared-energy (Pythagorean) form does not hold: an empirical cross-correlation analysis (Appendix D) finds mean $|\rho| = 0.40$ across train-mode methods, with maximum 0.68, ruling out $\|\delta_W\|^2 + \|\delta_{\mathrm{BN}}\|^2 = \|\delta_W + \delta_{\mathrm{BN}}\|^2$ as a quantitative attribution principle. Eq. (3) is exact at the level of LP accuracies (an algebraic identity), not at the level of squared norms. The negative correlation in BN-corrupted methods (gradient ascent shifts BN running statistics toward forget while weight updates shift representations away from forget) is consistent with the two-mechanisms framing of §3.1 but rules out additive squared-energy decomposition.

Together, weight invariance and the decomposition isolate two questions: how much of the measured leakage depends on normalization state, and how much remains after that state is recalibrated. We now examine these components across unlearning methods.

4 Empirical Evaluation

Setup. We evaluate 9 unlearning methods on CIFAR-10 (single-class, two-class) and CIFAR-100 (single-class, five-class), with ResNet-18 (CIFAR adaptation: 3×3 stem, no max-pool). Each unlearning method uses a single set of hyperparameters per dataset/forget-fraction condition, drawn from the reference implementation of each method or its closest available re-implementation; full configurations are in Appendix F. Linear probes use `sklearn.LogisticRegression(solver='lbfgs', max_iter=1000)` with 50 samples per class. The probe-budget choice is biased: Adam-trained probes underestimate LP-50 by 14–22pp for methods that genuinely modified the encoder (Appendix K), so we use lbfgs throughout. Relearning AUC is computed by fine-tuning on 50 forget-class samples for 50 epochs and integrating the normalized forget-accuracy curve. BN recalibration uses $N_r \geq 5000$ retain samples in train mode with gradients disabled. All multi-seed numbers are mean$\pm$std across three random seeds.

Table 1: Main diagnostic on CIFAR-10 (ResNet-18, single-class forget, class 0, three seeds; mean±std). $\Delta F = \text{Post-F}\% - \text{Pre-F}\%$ is the BN illusion magnitude, attributable to BN state by Corollary 1. Pre-R / Post-R = retain accuracy before / after recalibration. PGU† is our cleaner-covariance re-implementation; the official two-phase protocol yields $\Delta F = +14.5$pp (Appendix G).

Method	BN	Ret-aw.	Pre-F%	Post-F%	ΔF	Pre-R%	Post-R%	LP-50	ReAUC
Retrain	train	✓	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	99.6	99.6	75.5 ± 0.7	0.000
SCRUB	train	✓	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	88.2	90.9	82.1 ± 0.0	0.000
GA	train	–	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	57.2	81.5	79.4 ± 0.2	0.000
GA+FT	train	✓	67.9 ± 0.5	67.7 ± 0.3	-0.2 ± 0.2	96.3	96.3	89.7 ± 0.0	0.746
SalUn	train	–	3.0 ± 0.1	65.1 ± 0.3	$\mathbf{+62.1 \pm 0.3}$	85.0	95.1	88.9 ± 0.1	0.723
BadTeacher	eval	✓	18.9 ± 6.7	97.2 ± 0.1	$\mathbf{+78.2 \pm 6.7}$	95.7	95.9	91.4 ± 0.1	0.976
IncompTeacher	train	✓	39.1 ± 22.6	87.1 ± 11.7	$\mathbf{+48.0 \pm 11.4}$	90.1	89.1	89.8 ± 0.4	0.928
NoiseInject*	train	–	79.8 ± 0.5	98.7 ± 0.0	$\mathbf{+18.9 \pm 0.5}$	95.0	95.2	91.1 ± 0.0	0.988
SSD	train	–	25.7 ± 3.7	88.1 ± 1.0	$\mathbf{+62.4 \pm 4.2}$	79.1	95.7	90.3 ± 0.3	0.901
PGU†	eval	–	1.1 ± 0.0	62.0 ± 0.7	$\mathbf{+60.9 \pm 0.7}$	93.0	94.6	91.0 ± 0.0	0.641

*NoiseInject is a simplified Gaussian noise-injection baseline implemented for this work. It does not implement the Fisher forgetting procedure of Golatkar et al. [10]. Ret-aw. = retain-aware objective or training stage; ReAUC = relearning AUC.

4.1 Main diagnostic and self-consistency

Across the ten rows of Table 1, including Retrain, the association between pre-recalibration forget accuracy and leakage depends on the metric and test (Appendix N). Rank association with relearning is strong, but near-zero forget accuracy still leaves the relevant operating points unresolved. Among the four unlearning methods and Retrain with Pre-F $\leq 3\%$, relearning AUC spans 0.000–0.723 and LP-50 spans 75.5%–91.0%, against a retrain reference of 75.5%. Thus near-zero forget accuracy does not separate low-leakage checkpoints from residual leakage. After recalibration, association with relearning is $r = 0.998$ (Pearson, $p < 10^{-6}$), $\rho_S = 0.988$ (Spearman), and $\tau_K = 0.952$ (Kendall). Association with LP-50 is significant under all three tests ($p < 0.005$).

Table 1 reports the diagnostic results (three seeds; mean±std). Six of nine methods exhibit $\Delta F \geq 18.9$pp, with the largest reversal of +78pp on BadTeacher: forget-class accuracy rises from 19% to 97% after a single weight-preserving forward pass.

The four rows with $|\Delta F| \leq 0.2$pp split three ways. SCRUB reaches $\Delta F = 0$ via genuine erasure with retain preserved at 88%. GA reaches $\Delta F = 0$ at 500 steps by destroying retain (Pre-R = 57.2%, the lowest in the table); at fewer steps, ΔF traces a smooth phase transition from +76pp at 100 steps to 0pp at 500 steps (Appendix H). GA+FT's fine-tune step self-recalibrates BN. PGU (despite eval mode) exhibits $\Delta F \approx 61$pp under our re-implementation: projection controls *which* directions weights move, not the *magnitude* of activation drift relative to frozen running statistics, and once weights move enough to depress Pre-F% to near zero, BN misalignment is fully exposed (Appendix G). **Pre-R% reveals which $\Delta F = 0$ rows correspond to viable forgetting:** Retrain, SCRUB, and GA+FT preserve retain; GA does not, a distinction invisible from ΔF alone.

At fixed batch composition, a 3×3 learning-rate/step-count grid on BadTeacher and SalUn (Appendix I) gives $\Delta F \geq 57.8$pp for every configuration preserving retain accuracy at $\geq 85\%$. The GA step-count sweep (Appendix H) likewise links smaller gaps to degraded retain utility. Across the main comparison, the diagnostic reverses apparent forgetting in six of nine methods by 19–78 percentage points; retain accuracy and the auxiliary leakage measures identify the different zero-gap regimes.

4.2 2D decomposition: BN- vs. NC-residual

Table 2 instantiates Eq. (3) on both datasets. The BN-residual is the pre-minus-post LP difference, $-\Delta\text{LP}$, with weights fixed; its sign indicates whether BN measurement bias inflates (+) or deflates (−) the surface LP. The NC-residual is the post-recalibration LP elevation over retrain and is attributable to the encoder.

The empirical content of Corollary 2 is now visible. GA, SalUn, and SSD on CIFAR-100 sit far in the BN-deflation quadrant: $|\text{BN-residual}| \approx 20$–30pp, dwarfing the $|\text{NC-residual}| \approx 14$–17pp. The pre-recalibration LP for these methods sits *below* the retrain reference; a reader looking only at the surface LP would conclude these methods stripped forget-class linear separability below the level of a model that never saw the class. The post-recalibration LP exceeds retrain by 14–17pp, indicating

Table 2: 2D LP-elevation decomposition (single seed): NC-residual = $\mathrm{LP}^{M_{\mathrm{cal}}} - \mathrm{LP}^{M_{\mathrm{retr}}}$, BN-residual = $\mathrm{LP}^{M} - \mathrm{LP}^{M_{\mathrm{cal}}}$. CIFAR-10: $\mathrm{LP}^{M_{\mathrm{retr}}} = 76.10$. CIFAR-100: $\mathrm{LP}^{M_{\mathrm{retr}}} = 73.33$. The CIFAR-100 column for GA, SalUn, and SSD shows large *negative* BN-residuals: BN deflates the pre-recalibration LP, masking the substantial positive NC-residual. Separately rounded entries can differ by 0.01pp when subtracted.

	CIFAR-10				CIFAR-100			
Method	LP^{M}	$\mathrm{LP}^{M_{\mathrm{cal}}}$	NC-res	BN-res	LP^{M}	$\mathrm{LP}^{M_{\mathrm{cal}}}$	NC-res	BN-res
Retrain	76.10	76.10	0.00	0.00	73.33	73.33	0.00	0.00
SCRUB	79.23	79.30	+3.20	−0.07	81.67	86.67	+13.33	−5.00
GA	67.67	77.93	+1.83	−10.27	58.00	87.67	+14.33	−29.67
SalUn	77.47	86.23	+10.13	−8.77	60.67	88.00	+14.67	−27.33
SSD	82.33	88.37	+12.27	−6.03	70.00	90.67	+17.33	−20.67
GA+FT	89.60	87.30	+11.20	+2.30	88.67	88.67	+15.33	0.00
BadTeacher	89.27	88.90	+12.80	+0.37	82.00	89.00	+15.67	−7.00
NoiseInject	89.97	89.07	+12.97	+0.90	83.67	86.33	+13.00	−2.67
IncompTeacher	90.63	87.57	+11.47	+3.07	87.00	90.00	+16.67	−3.00
PGU	89.10	88.33	+12.23	+0.77	93.67	94.33	+21.00	−0.66

the encoder retains substantially more forget-class structure than retrain, hidden by BN measurement bias from concurrent work [8]. The Neural Collapse story is therefore strengthened, not refuted: encoder-level severity is larger than concurrent work could measure, but requires recalibration to expose. For methods with small BN-residual on CIFAR-10 (BadTeacher, NoiseInject, IncompTeacher, PGU), the NC-residual is the dominant LP-elevation component and the surface LP only slightly overstates it; the qualitative attribution to encoder geometry is unchanged.

Quadrant geometry. The 9 methods cluster into four regions of the (BN-residual, NC-residual) plane. **(I) Origin (genuine forgetting):** only Retrain occupies this region with both residuals at zero. **(II) Positive BN-residual, positive NC-residual:** CIFAR-10 BadTeacher, NoiseInject, IncompTeacher, GA+FT, and PGU; the surface LP overstates encoder leakage but is qualitatively in the right direction. **(III) Negative BN-residual, positive NC-residual (BN-deflation):** CIFAR-100 GA, SalUn, SSD; the surface LP *understates* encoder leakage, sometimes by a factor that reverses the sign relative to retrain. This is the regime in which the BN illusion is most dangerous to interpret, because the surface measurement looks safer than retrain. **(IV) Sign-flipped:** no method we tested falls here, but the framework predicts a method could in principle have positive BN-residual and negative NC-residual (the surface LP overstates encoder leakage, which is itself below retrain). The four-region picture is what the unique decomposition of Proposition 1 actually buys: a rigorous taxonomy of how surface-level LP departs from encoder-level LP, applicable to any future method evaluated on a BN backbone.

The decomposition therefore distinguishes normalization that masks leakage from normalization that inflates it, while the post-recalibration residual measures elevation relative to retrain.

Cross-dataset replication on CIFAR-100. We write $\Delta\mathrm{LP}$ and $\Delta\mathrm{NCC}$ for the post-minus-pre changes in linear-probe and nearest-class-mean accuracy. We replicate the diagnostic on CIFAR-100 in single-class ($p = 0.01$) and five-class ($p = 0.05$) forget conditions (Appendix E). Two findings. First, $|\Delta\mathrm{LP}|$ scales inversely with forget fraction p: for GA, CIFAR-100 1-class $\Delta\mathrm{LP}$=+29.7pp drops to +2.1pp at 5-class; CIFAR-10 1-class (p=0.10) is +11.9pp, 2-class (p=0.20) is +5.0pp, consistent with EMA absorption arithmetic. Second, only Retrain on CIFAR-100 5-class achieves negative $\Delta\mathrm{LP}$ (−6.0pp) and negative $\Delta\mathrm{NCC}$ (−6.9pp). Negative $\Delta\mathrm{LP}$ post-recalibration is consistent with genuine forgetting, but it is a coarse indicator and not a reliable diagnostic on its own: SCRUB, which exhibits genuine forgetting by every other measure, produces $\Delta\mathrm{LP} > 0$ in three of four CIFAR conditions we test.

4.3 Threat model

The adversary's objective is to restore forget-class capability, not to reconstruct training examples. The adversary holds the released checkpoint (weights and BN running statistics), has at least ten unlabeled natural images, and can run the model locally. The attack requires neither labels nor

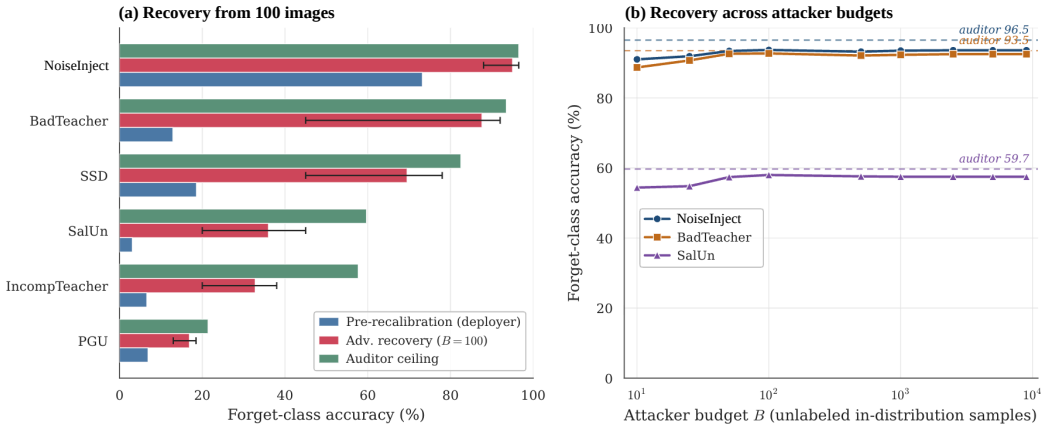

Figure 1: Adversarial recovery on BN-vulnerable methods (CIFAR-10, ResNet-18). **(a)** Forget-class accuracy before recalibration (blue), after recovery using $B = 100$ in-distribution unlabeled images (red; three-seed mean and seed-range error bars), and after full-retain auditor recalibration (green). **(b)** Forget-class accuracy versus attacker budget B for three representative methods; dashed horizontal lines mark their auditor references. Substantial recovery is already visible at $B = 10$.

gradient updates nor access to the original model or designated forget set. We measure success by post-recalibration forget-class accuracy and its change from the released checkpoint; these are the quantities reported in Section 4.4. Membership inference is evaluated separately in Appendix O.

Setting	Actor's access	Recalibration	Relation to model release
Model release	Checkpoint and local execution	Directly applies the recovery pass	Primary attack setting
Hosted API	Inference queries only	Cannot update BN statistics through the interface	Disclosure of the checkpoint enables the primary attack
Auditor / regulator	Checkpoint and retain data	Runs the same pass as an evaluation control	Same operation, different purpose

The recovery procedure acts directly on a locally accessible checkpoint. An inference-only interface blocks this procedure, while an auditor uses it to test whether low forget accuracy survives normalization-state correction.

4.4 Adversarial recovery: the artifact is exploitable

The diagnostic results so far are evaluation-side. The same operation, run by an attacker in the model-release setting, becomes a recovery procedure. We vary recalibration data along two axes: (i) distribution: original retain set (auditor view), CIFAR-10 test (in-distribution adversary), or CIFAR-100 raw pixels with CIFAR-10 normalization (out-of-distribution adversary); (ii) budget $B \in \{10, 25, 50, 100, 500, 1000, 2500, 5000, 9000\}$ with three random seeds. The attacker's procedure is identical to the diagnostic: a single forward pass with running statistics updated and weights frozen. No labels, no gradients, no model queries beyond the forward sweep.

Findings. At $B = 100$, the in-distribution attack substantially restores forget-class accuracy across the vulnerable methods in Figure 1a. The sub-100-budget results in Appendix J show that BadTeacher and NoiseInject reach 88.7% and 91.0% forget-class accuracy at $B = 10$. SalUn reaches 54.4% at $B = 10$ and 58.0% at $B = 100$. These are absolute accuracies, rather than fractions of the original model's accuracy. At $B = 100$, recovery with CIFAR-100 images under CIFAR-10 normalization reaches 78%–95% of the corresponding in-distribution accuracy (Appendix J.2). Thus a ten-image budget already exposes the artifact in the tested model-release setting.

4.5 Mechanistic falsification and scope

GroupNorm has no running statistics, so the recalibration analogue must be a no-op. We retrain ResNet-18 on CIFAR-10 with all `BatchNorm2d` replaced by `GroupNorm`, and re-run eight unlearning methods plus Retrain (PGU omitted on GN due to compute). ΔF collapses to exactly 0.00pp uniformly (Appendix F, Table 6), falsifying any architecture-, dataset-, or hyperparameter-level confound. Relearn-AUC remains uniformly high (95%–98%) on GN-ResNet; the measurement artifact disappears, but encoder-level failure [8, 20] is normalization-independent. The artifact also transfers to ResNet-50 on Tiny-ImageNet: BadTeacher $\Delta F{=}{+}82.2$pp (vs. $+78.2$ on CIFAR-10), PGU $+29.6$pp, with the GA step-count phase transition replicating with the cliff edge shifted earlier by 200-class competitive pressure (Appendix L). On ViT-S/16 (LayerNorm) the recalibration operator is the identity by Theorem 1; we verify $\Delta F{=}0.00$pp across 5 methods tuned into a forgetting regime (Appendix M), confirming the artifact is normalization-specific. At ImageNet resolution, a pretrained ResNet-50 on ImageNet-100 gives BadTeacher $\Delta F = +92.0$pp with retain accuracy preserved, and ten-image recovery reaches 82.0% forget accuracy (Appendix T). Together, the controls isolate the recalibratable BN state, while the larger-image experiments demonstrate that the recovery effect extends beyond CIFAR.

4.6 Discussion

Tuning at fixed batch composition. In the tested 3×3 learning-rate/step-count grid, every BadTeacher or SalUn configuration preserving retain accuracy at $\geq 85\%$ retains a recalibration gap of at least 57.8pp (Appendix I). This result concerns that grid and fixed batch composition. Changing the retain fraction is a separate intervention: retain-mixed batches mitigate SalUn's corruption component in Appendix Q.

Why methods resist, and the corresponding defenses. A small ΔF has different causes. SCRUB combines a zero gap with low relearning susceptibility and 88% retain accuracy. GA reaches a zero gap at 500 steps with substantially degraded retain accuracy; Appendix H shows the transition within the same method. GA+FT instead self-recalibrates during retain fine-tuning, giving $\Delta F = -0.2$pp. This suggests three controls. Architecturally, GN/LN remove the running statistics responsible for the artifact. Procedurally, freezing BN during unlearning prevents corruption, while retain recalibration corrects any remaining weight–normalization misalignment before evaluation. At the data level, a retain fraction of at least 25% removes the observed SalUn gap in the tested mixed-batch sweep. This mixing result is specific to the evaluated pipeline; BadTeacher's eval-mode misalignment persists at its reference mixing ratio.

Three-regime decomposition of $\Delta F = 0$ at scale. At CIFAR-10 scale, $\Delta F = 0$ admitted a clean reading. At Tiny-ImageNet scale (Appendix L) it splits three ways: *genuine forgetting with retain preserved* (SCRUB; high breakthrough epoch in relearning); *encoder-level failure* where the head suppresses the forget class but the encoder does not move (GA+FT, SSD: high LP, high relearn-AUC); and *saturated weight erasure with destroyed retain* (GA, IncompTeacher: $\leq 3\%$ retain). The auxiliary columns (Pre-R, LP, ReAUC) are required to distinguish these.

Practical recommendations and scope. Deployers with retain access should recalibrate post-unlearning and serve the recalibrated model; without retain access, any unlabeled natural-image proxy fixes most of the artifact. New methods should prefer GN/LN architectures or freeze BN and recalibrate before evaluation; benchmarks should require pre/post-recalibration reporting alongside encoder-level checks [8, 20]. Post-recalibration $F\%{=}0$ is not a privacy guarantee; pairing the diagnostic with MIA, MIA-NN, or UMA-style verifiers yields a more trustworthy evaluation.

Composition with concurrent encoder-level diagnostics. The decomposition in Section 3 positions BN recalibration as a prerequisite for, not a competitor to, encoder-level evaluation. Linear probing [8], nearest-class-mean (NCC) accuracy, and backbone-freezing classifier retraining [20] all measure properties of the encoder, properties that, on BN-based architectures, are obscured by a measurement bias they were not designed to control for. Reporting LP^{M} alone risks the kind of sign-error illustrated in Table 2 for GA/SalUn/SSD on CIFAR-100, where the surface measurement places these methods *below* retrain on linear separability while the corrected measurement places them above. The remedy is to report $\mathrm{LP}^{M_{\mathrm{cal}}}$ or, when the BN-residual is itself of interest, both. A

small additional cost (one forward sweep) buys a quantitative reading of an effect that is otherwise invisible.

What the diagnostic does and does not establish. The diagnostic does not establish that any of the methods we evaluate are wrong, or that their authors made an error in their own evaluation protocols; it establishes that the standard surface-metric reading of forget accuracy on BN architectures is unreliable, and that the apparent forgetting in six of nine methods is recoverable by a transformation that modifies no weight. Whether a given method "forgets" is a question about the encoder; the diagnostic reframes it as a question about the recalibrated checkpoint, M_{cal}, which is the unique representative of the model's residual forget-class capability under the unique decomposition of Proposition 1. Honest evaluation requires reporting ΔF alongside Pre-R and Post-R. The joint value distinguishes genuine forgetting (Pre-R high, $\Delta F = 0$) from saturated weight erasure (Pre-R low, $\Delta F = 0$) from the BN illusion (Pre-R high, ΔF large) in a way no single column can.

Limitations. We evaluate classification benchmarks (CIFAR-10/100, Tiny-ImageNet, and ImageNet-100), primarily with class-level forget sets. Instance-wise unlearning, with forget samples drawn randomly across all classes, is evaluated in one condition (Appendix P). Full ImageNet-1k evaluation was not feasible within the available compute budget. We do not benchmark on language models, where the analogue of BatchNorm is LayerNorm and Theorem 1 predicts the diagnostic is the identity (the ViT-S/16 result of Appendix M confirms this for vision LayerNorm; we do not test text). The threat model assumes local checkpoint access; defending against an attacker with checkpoint access requires architectural changes (GN/LN) rather than evaluation-side controls.

Why the artifact has gone undocumented. The recalibration operation is standard practice in test-time adaptation [27, 23] and SWA [16]; the mechanism is well-known. We attribute the absence of recalibration as an unlearning evaluation control to three factors. First, retain-set BN recalibration is not commonly listed as a step in the standard unlearning pipeline; authors evaluate on the model produced by the method, exactly as released. Second, the artifact is direction-specific: BN corruption *lowers* forget accuracy on the surface, which looks like the desired outcome rather than the opposite of it. Third, the most-cited unlearning benchmarks [35, 2, 21] score methods on metrics that, on BN backbones, the standard recalibration would shift; in the absence of an explicit control, one cannot tell whether a high-scoring method is genuinely forgetting or successfully exploiting the artifact. Our diagnostic adds a single line of code per evaluated checkpoint and recovers the missing control. The cost is one forward pass with gradients disabled; the value is a quantitative, weight-preserving, mechanistically grounded reading of an effect that previously had no name.

5 Conclusion

A single forward pass over retain data, modifying no weight, reverses the apparent forgetting in six of nine evaluated unlearning methods, is exploitable by an attacker with as few as ten unlabeled images, and is absent under GroupNorm. The operation is a deterministic, weight-preserving fixed-point on the model's normalization state, so any pre/post gap is provably attributable to BatchNorm rather than to weights. Headline forgetting numbers on BN architectures should be re-reported with the diagnostic applied; the artifact is mechanistically distinct from, and complementary to, the encoder-level failures concurrent work has documented, with our decomposition revealing that BN measurement bias *masks* encoder leakage rather than creating it for several methods. Both normalization-state and encoder-level controls are therefore needed. The membership-inference experiment reinforces the metric-specific scope: recalibration changes attack AUC by at most 0.012, despite forget-accuracy changes of up to approximately 98pp on those checkpoints (Appendix O). The diagnostic is one function call. We name the effect the *BN illusion*. Future BN-based unlearning evaluations should report ΔF alongside Pre-R and Post-R as standard practice; the recalibrated checkpoint M_{cal} is the unbiased reading of the model's residual forget-class capability.

Acknowledgments and Disclosure of Funding

This work was supported by the SPARKLE program (Student Program for Advancing Research, KnowLedge, and Entrepreneurship) of BITS Pilani, SPARKLE ID SPP2608.

References

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

[2] Xavier F Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep Unlearn: Benchmarking machine unlearning for image classification. *arXiv preprint arXiv:2410.01276*, 2024.

[3] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, 2015.

[4] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI Conference on Artificial Intelligence*, 2023.

[5] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 2023.

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

[7] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI Conference on Artificial Intelligence*, 2024.

[8] Yichen Gao, Altay Unal, Akshay Rangamani, and Zhihui Zhu. An illusion of unlearning? Assessing machine unlearning through internal representations. *arXiv preprint arXiv:2604.08271*, 2026.

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

[10] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI Conference on Artificial Intelligence*, 2021.

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[13] Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. *arXiv preprint arXiv:2312.04095*, 2024. WACV 2024.

[14] Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Jogging the memory of unlearned model through targeted relearning attack. *arXiv preprint arXiv:2406.13356v1*, 2024.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[16] Pavel Izmailov, Dmitry Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[17] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, 2021.

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

[19] Anjie Le, Can Peng, Yuyuan Liu, and J. Alison Noble. POUR: A provably optimal method for unlearning representations via neural collapse. *arXiv preprint arXiv:2511.19339*, 2025.

[20] Jaewon Lee, Yongwoo Kim, and Donghyun Kim. Erase at the Core: Representation unlearning for machine unlearning. *arXiv preprint arXiv:2602.05375*, 2026.

[21] Xiang Li, Bhavani Thuraisingham, and Wenqi Wei. MUBox: A critical evaluation framework of deep machine unlearning. *arXiv preprint arXiv:2505.08576*, 2025.

[22] Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022.

[23] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[24] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, 2019. doi: 10.1109/SP.2019.00065.

[25] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, 2018.

[26] Debdeep Sanyal and Murari Mandal. Agents are all you need for LLM unlearning. In *Conference on Language Modeling*, 2025.

[27] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems*, 2020.

[28] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop, NeurIPS*, 2018.

[29] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, and Stephen Casper. Latent adversarial training improves robustness to persistent harmful behaviors in LLMs. *arXiv preprint arXiv:2407.15549*, 2024.

[30] Yash Sinha, Murari Mandal, and Mohan Kankanhalli. Multi-modal recommendation unlearning for legal, licensing, and modality constraints. In *AAAI Conference on Artificial Intelligence*, 2025.

[31] Yash Sinha, Murari Mandal, and Mohan Kankanhalli. Distill to delete: Unlearning in graph networks with knowledge distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

[32] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Deep regression unlearning. In *International Conference on Machine Learning*, 2023.

[33] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[34] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: Understanding factors influencing machine unlearning. *arXiv preprint arXiv:2109.13398*, 2021.

[35] Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio C. S. Jacques Junior, et al. Are we making progress in unlearning? findings from the first NeurIPS unlearning competition. *arXiv preprint arXiv:2406.09073*, 2024.

[36] Yian Wang, Ali Ebrahimpour-Boroojeny, and Hari Sundaram. On the necessity of output distribution reweighting for effective class unlearning. *arXiv preprint arXiv:2506.20893v1*, 2025.

[37] Hao Xuan and Xingyu Li. Verifying robust unlearning: Probing residual knowledge in unlearned models. *arXiv preprint arXiv:2504.14798v1*, 2025.

A BN Corruption: Formal Statement and Proof

In training mode, batch statistics update the running buffers during the forward pass. An operation applied subsequently to weight gradients does not intercept this update. The quantitative effect depends on the separation of the activation moments, rather than on class labels alone.

Theorem 2 (BN corruption / non-interceptability)**.** *Let u_t be the running mean at a BN layer. Assume **(A1)** that the forget and retain pre-normalization mean vectors at the weights under consideration satisfy*

$$d_\ell := \frac{\|\mu_f^\ell - \mu_r^\ell\|_2}{\|\sigma_r^\ell\|_2} \geq \delta > 0, \qquad \|\sigma_r^\ell\|_2 > 0, \tag{4}$$

*and **(A2)** that BN runs in training mode with EMA momentum $m \in (0, 1]$. Every unlearning-batch forward pass executes $u_{t+1} = (1-m)u_t + m\hat{\mu}_t$, independently of any operator applied only to weight gradients. If, additionally, the activation-moment targets remain fixed, the batches have expected mean μ_f^ℓ, and $u_0 = \mu_r^\ell$, then*

$$\frac{\|\mathbb{E}u_T - \mu_r^\ell\|_2}{\|\sigma_r^\ell\|_2} = [1-(1-m)^T]d_\ell \geq [1-(1-m)^T]\delta. \tag{5}$$

This bound concerns normalization-state drift, not a lower bound on forget accuracy.

Proof. Unrolling the forward-pass update gives

$$u_T = (1-m)^T u_0 + m\sum_{t=0}^{T-1}(1-m)^{T-1-t}\hat{\mu}_t. \tag{6}$$

Under the fixed-target assumptions, take expectations and use $m\sum_{t=0}^{T-1}(1-m)^{T-1-t} = 1-(1-m)^T$ to obtain $\mathbb{E}u_T - \mu_r^\ell = [1-(1-m)^T](\mu_f^\ell - \mu_r^\ell)$. Equation (5) follows from (A1). The EMA update uses the current batch statistics, not the weight gradients. A gradient projection can alter later weights and hence later activation distributions, but cannot prevent the buffer update already executed by the forward pass. □

Changing weights and moving targets. For an arbitrary weight trajectory, let a_t be the conditional expected mean of the current unlearning batch, q_t the contemporaneous retain mean, and $\xi_t = \hat{\mu}_t - a_t$. Writing $w_t = (1-m)^{T-1-t}$ gives the exact identity

$$\begin{aligned} u_T - q_T = (1-m)^T(u_0 - q_0) + m\sum_{t=0}^{T-1} w_t(a_t - q_t) \\ + \sum_{t=0}^{T-1} w_t(q_t - q_{t+1}) + m\sum_{t=0}^{T-1} w_t\xi_t. \end{aligned} \tag{7}$$

The terms separate initial mismatch, activation separation, movement of the retain target, and batch noise. A cumulative lower bound for changing weights requires control of target motion and cancellation between the separation vectors. The forward-pass non-interceptability statement itself holds throughout the trajectory. Appendix R measures activation separation at the pre-unlearning weights.

Mixed batches and loss independence. For a mixed loader, a_t in Eq. (7) is the actual pre-BN mean induced by that loader. When the feature map before the layer is fixed independently of batch composition, a retain fraction r_{mix} gives $a_{\text{mix}} = (1-r_{\text{mix}})\mu_f + r_{\text{mix}}\mu_r$, reducing the mean separation by the factor $1 - r_{\text{mix}}$. Deeper train-mode BN layers can themselves change this feature map through earlier batch normalizations. The mixed-batch experiment in Appendix Q therefore measures the resulting effect directly. The update does not depend on which loss generated the gradients: the structural statement applies to gradient-ascent and data-based procedures alike, including RandomLabeling (Appendix S).

Limits of the static-target quantitative prediction. A quasi-stationary approximation predicts that relative excess BN drift, normalized by the natural retain offset, scales as $[1-(1-m)^T](1-p)/p$ when weight-induced activation drift is small. Across five CIFAR-10/100 conditions, the fitted slope relating this prediction to measured drift was 0.019, compared with the identity-prediction slope of 1.0. The approximation did not provide a useful quantitative prediction: during unlearning, weight updates change the activation moments that the EMA tracks, as Eq. (7) makes explicit. We retain the structural mechanism but do not use this static-target approximation to predict drift magnitudes. Recalibration in Theorem 1 is applied after unlearning, at fixed weights, so its idempotence concerns a different operation.

B Recalibration: Proof, Finite-Batch Error, and Layerwise Measurements

B.1 Population properties and empirical idempotence

Proof of Theorem 1(i)–(iii). **(i)** The state $\varphi^*(\theta;\mathcal{D}_r)$ depends only on the weights and retain distribution. Since $\mathcal{R}^*$ leaves θ unchanged, a second application returns the same state. **(ii)** Weight invariance follows from Eq. (2). Pre-BN activations at the first layer are unchanged. At later layers they depend on the updated normalization of earlier layers and may change; the attribution uses weight invariance, not activation invariance. **(iii)** At each BN layer, eval-mode normalization with its oracle moments computes $y_c = \gamma_c(h_c - \mu_c^*)/\sqrt{v_c^* + \epsilon} + \beta_c$. Starting from the first layer and propagating through the network, this is exactly the oracle-normalized forward map at the fixed weights. □

The empirical pass resets the running statistics and uses train-mode batch statistics for forward normalization. For a fixed sequence of input tensors and batches $\mathcal{B}$, deterministic computation therefore produces a state $\widehat{\varphi}_{\mathcal{B}}(\theta)$ independent of the incoming buffers:

$$\widehat{\mathcal{R}}_{\mathcal{B}}(\theta,\varphi) = (\theta, \widehat{\varphi}_{\mathcal{B}}(\theta)), \qquad \widehat{\mathcal{R}}_{\mathcal{B}} \circ \widehat{\mathcal{R}}_{\mathcal{B}} = \widehat{\mathcal{R}}_{\mathcal{B}}. \tag{8}$$

Repeated passes with the same inputs and partition yielded bit-identical running statistics. Resampling the partition changes the empirical operator; the observed variation was approximately 10^{-2}. Stochastic input transformations and any other forward-pass randomness must also be fixed for exact repeatability.

B.2 Finite-batch derivation

Assumptions. Hold the weights and finite network depth L fixed. Let $P = \mathcal{D}_r$ be a distribution of independent input images, with sufficient moments for the mean and variance estimators below; spatial sites within an image need not be independent. BN uses a fixed $\epsilon > 0$. Assume the recursively defined activation-moment functionals admit the second-order expansion in Eq. (10), with the stated moment and remainder bounds. For ReLU networks this is regularity of the expected moment functionals, including activation-threshold crossings, rather than pointwise twice differentiability of ReLU. First consider $N_r = Kb$ images in K equal-sized batches.

The local expansion and the Jensen term. At one scalar BN coordinate write $F(h;u,v) = \gamma(h-u)(v+\epsilon)^{-1/2} + \beta$ and $a = v + \epsilon$. For moment errors $\delta_u = \hat{u} - u$ and $\delta_v = \hat{v} - v$, Taylor expansion in the estimated moments gives

$$\begin{aligned} F(h;\hat{u},\hat{v}) - F(h;u,v) = & -\frac{\gamma\delta_u}{\sqrt{a}} - \frac{\gamma(h-u)\delta_v}{2a^{3/2}} \\ & + \frac{\gamma\delta_u\delta_v}{2a^{3/2}} + \frac{3\gamma(h-u)\delta_v^2}{8a^{5/2}} + \mathcal{E}_b. \end{aligned} \tag{9}$$

The reciprocal-standard-deviation map $q(v) = (v+\epsilon)^{-1/2}$ is convex, since $q''(v) = 3(v+\epsilon)^{-5/2}/4 > 0$. Jensen's inequality applies to the random batch variance $\hat{v}$: $\mathbb{E}q(\hat{v}) \geq q(\mathbb{E}\hat{v})$. The quadratic terms in Eq. (9) have order b^{-1} when moment fluctuations have size $b^{-1/2}$ and the corresponding moments and remainder are controlled. This does not assign a universal sign to the complete output bias: h, $\hat{u}$, and $\hat{v}$ come from the same batch and are dependent. The following calculation retains that dependence.

From one batch to the full network. Let $\mathcal{T}_\ell(P)$ be the pre-BN population mean and variance at layer ℓ when preceding layers use their oracle P moments. It includes within-image spatial averaging. For the empirical image distribution $P_b = b^{-1}\sum_{i=1}^{b}\delta_{X_i}$, train-mode normalization uses the corresponding batch moments $\mathcal{T}_\ell(P_b)$ throughout the preceding layers. Write the assumed second-order expansion as

$$\mathcal{T}_\ell(P_b) - \mathcal{T}_\ell(P) = \frac{1}{b}\sum_{i=1}^{b}\psi_\ell(X_i) + \frac{1}{2b^2}\sum_{i,j=1}^{b}\Psi_\ell(X_i, X_j) + r_{\ell,b}. \tag{10}$$

Here $\mathbb{E}\psi_\ell(X) = 0$ and $\mathbb{E}\Psi_\ell(x, X) = \mathbb{E}\Psi_\ell(X, x) = 0$ for every x; the kernels have finite second moments, including on the diagonal. Assume $\mathbb{E}\|r_{\ell,b}\| = o(b^{-1})$ and $\mathbb{E}\|r_{\ell,b}\|^2 = o(b^{-1})$. The first-order expectation vanishes. Independence of images eliminates the off-diagonal expectations in the double sum, leaving its b diagonal terms:

$$\mathbb{E}\mathcal{T}_\ell(P_b) - \mathcal{T}_\ell(P) = \frac{\mathbb{E}\Psi_\ell(X, X)}{2b} + o(b^{-1}), \qquad \mathbb{E}\|\mathcal{T}_\ell(P_b) - \mathbb{E}\mathcal{T}_\ell(P_b)\|^2 = O_L(b^{-1}). \tag{11}$$

This yields the finite-batch offset without treating a sample and its batch statistics as independent.

PyTorch uses the biased batch variance for forward normalization and a Bessel-corrected variance for its running-variance update.[1] For bS_ℓ scalar sites per channel, the factor $bS_\ell/(bS_\ell - 1)$ contributes another $O(b^{-1})$ term at fixed spatial size S_ℓ; it does not make spatial sites independent. Thus the actual statistic $S_{\ell,b}$ written into the running average satisfies

$$\mathbb{E}S_{\ell,b} = \varphi^*_\ell + \frac{c_\ell}{b} + o(b^{-1}), \qquad \mathbb{E}\|S_{\ell,b} - \mathbb{E}S_{\ell,b}\|^2 = O_L(b^{-1}). \tag{12}$$

Averaging retain batches. For an arithmetic average of $K = N_r/b$ independent batches, $\widehat{\varphi}_{\ell,N_r,b} = K^{-1}\sum_{k=1}^{K} S^{(k)}_{\ell,b}$. Equation (12) gives a mean-squared sampling fluctuation $K^{-1}O_L(b^{-1}) = O_L(N_r^{-1})$, while the expectation retains the finite-batch offset. Over the finite collection of layers and channels,

$$\boxed{\|\widehat{\varphi}_{N_r,b} - \varphi^*\| = O_p(N_r^{-1/2}) + O_L(b^{-1}).} \tag{13}$$

For unequal batch sizes b_k and normalized averaging weights w_k, the corresponding terms are $O_p((\sum_k w_k^2/b_k)^{1/2})$ and $O_L(\sum_k w_k/b_k)$, using the actual size and weight of the final batch.

The functionals $\mathcal{T}_\ell$ already include propagation through earlier layers, so their constants depend on depth, weights, and activation sensitivities. Equation (13) is a fixed-depth result, not a depth-uniform bound. Increasing N_r at fixed b reduces sampling variation but does not remove the finite-batch term.

B.3 Empirical reference agreement and layerwise mismatch

Agreement between recalibration configurations. We compare statistics estimated from $5{,}000$ retain images with an empirical reference estimated from all $45{,}000$ retain images at batch size 256, using `update_bn` at the same fixed unlearned weights. At the default batch size of 128, the layer-averaged normalized running-mean discrepancy,

$$\frac{1}{L}\sum_{\ell=1}^{L}\frac{\|\widehat{\mu}^\ell_{5000,128} - \widehat{\mu}^\ell_{45000,256}\|_2}{\|\widehat{\sigma}^\ell_{45000,256}\|_2}, \tag{14}$$

is 0.0016–0.0018 across GA, SalUn, SCRUB, and BadTeacher. These measurements quantify agreement with the full-retain empirical reference, which itself uses finite batches, rather than error against the oracle φ^*.

Layerwise normalization mismatch. Figure 2 compares each unlearned checkpoint with its retain-recalibrated copy. Define

$$\delta^\ell_\mu = \frac{\|\mu^\ell - \widehat{\mu}^\ell_{\mathrm{cal}}\|_2}{\|\widehat{\sigma}^\ell_{\mathrm{cal}}\|_2}, \qquad \delta^\ell_v = \frac{\|(\sigma^\ell)^2 - (\widehat{\sigma}^\ell_{\mathrm{cal}})^2\|_2}{\|(\widehat{\sigma}^\ell_{\mathrm{cal}})^2\|_2}. \tag{15}$$

[1] https://docs.pytorch.org/docs/2.10/generated/torch.nn.BatchNorm2d.html

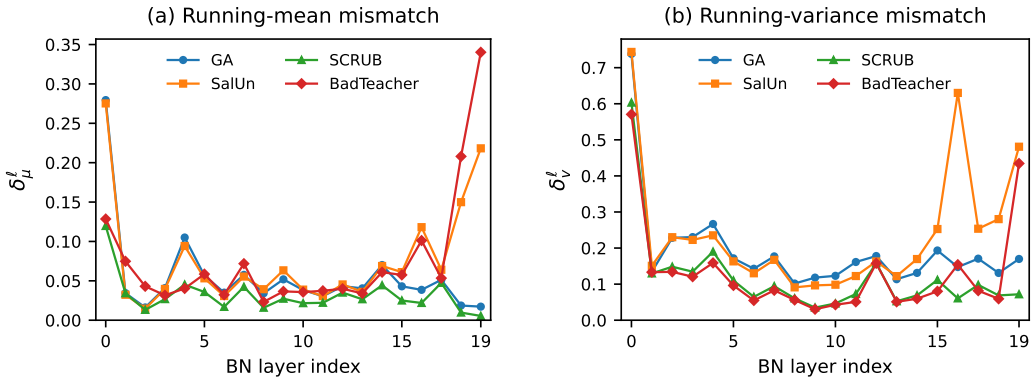

Figure 2: Layerwise normalization mismatch on CIFAR-10/ResNet-18, forgetting class 0. Each checkpoint is compared with its own retain-recalibrated copy ($N_r = 5000$, $b = 128$, fixed subset and batch order). **(a)** Normalized running-mean mismatch. **(b)** Relative running-variance mismatch, as defined in Eq. (15). GA, SalUn, and SCRUB use train-mode BN during unlearning; BadTeacher uses eval mode. All 20 BN layers are shown in module order. The experiment uses seed 0; no across-seed error bars are shown.

Across the 20 BN layers, the mean-mismatch profiles are method-dependent and non-monotonic. Fitted slopes per layer are -0.0040 for GA, -0.0017 for SCRUB, $+0.0018$ for SalUn, and $+0.0056$ for BadTeacher. The largest observed layerwise mean mismatch is 0.3402, at BadTeacher's final BN layer. These profiles locate the normalization mismatch corrected by recalibration; they measure a different quantity from finite-batch estimation bias.

C Proof of Proposition 1

Proof. The decomposition is the identity $A - C = (B - C) + (A - B)$ for $A = \mathrm{LP}^M, B = \mathrm{LP}^{M_{\mathrm{cal}}}, C = \mathrm{LP}^{M_{\mathrm{retr}}}$. Existence: when $\varphi = \varphi^*$, then $M = M_{\mathrm{cal}}$ so the BN-residual $A - B = 0$; under $\mathcal{R}^*$, $A \to B, B \to B, C \to C$, so the NC-residual $B - C$ is invariant. Uniqueness: suppose $\mathrm{LP}^M - \mathrm{LP}^{M_{\mathrm{retr}}} = X + Y$ with X vanishing at $\varphi = \varphi^*$ and Y invariant under $\mathcal{R}^*$. Evaluate the identity on M_{cal}: the left side becomes $B - C$, X vanishes, and Y is unchanged by invariance. So $Y = B - C$, hence $X = A - B$. Corollary 2 follows: the BN-residual is induced by an operator leaving encoder weights unchanged, so any nonzero BN-residual reflects a measurement bias, not an encoder-geometry change. □

D Cross-Correlation: Why the Pythagorean Form Fails

For each method we measured the empirical correlation ρ between weight-induced (δ_W) and BN-state-induced (δ_{BN}) penultimate-layer feature perturbations on forget-class inputs. The implementation computes

$$\delta_W = h(\theta_T, \varphi^*) - h(\theta_0, \varphi^*(\theta_0)),$$
$$\delta_{\mathrm{BN}} = h(\theta_T, \varphi_T) - h(\theta_T, \varphi^*).$$

The first compares weights with both models recalibrated; the second isolates the statistics effect at fixed final weights. Here (θ_0, φ_0) is the pre-unlearning checkpoint, (θ_T, φ_T) is the post-unlearning checkpoint, and $\varphi^* = \varphi^*(\theta_T; \mathcal{D}_r)$ is the recalibrated state defined in Section 3.2. Computing δ_{BN} against φ^* rather than φ_0 means δ_{BN} measures the deviation of the post-unlearning running stats from the population fixed point at fixed final weights—the same quantity that drives the diagnostic ΔF in the main paper.

Regime-dependence. For methods that update BN running statistics during unlearning (train-mode methods: GA, SCRUB, SalUn, IncompTeacher, NoiseInject, SSD, GA+FT), φ_T diverges from the population fixed point during unlearning, and δ_{BN} captures the corruption-induced perturbation. For methods that freeze BN during unlearning (eval-mode methods: BadTeacher, and PGU in our

re-implementation), running statistics never move: $\varphi_T = \varphi_0$ exactly, verified by direct checkpoint comparison (max $\|\varphi_T - \varphi_0\|_\infty = 0$ across all 20 BN layers for BadTeacher). For these methods δ_{BN} reduces to $h(\theta_T, \varphi_0) - h(\theta_T, \varphi^*)$, a calibration gap between full-data normalization (φ_0, computed from 50k training samples over all classes) and retain-only normalization (φ^*, computed from 45k retain samples over 9 classes). This gap is small in LP space (BadTeacher's BN-residual in Table 2 is $+0.37$pp on CIFAR-10) but has a fixed magnitude in feature space because both stat sets are well-calibrated to their respective distributions.

Table 3: Cross-correlation analysis on CIFAR-10 (single-class). $E_W = \|\delta_W\|^2$, $E_{\mathrm{BN}} = \|\delta_{\mathrm{BN}}\|^2$, $E_{\mathrm{total}} = \|\delta_W + \delta_{\mathrm{BN}}\|^2$. Pyth-error $= |E_W + E_{\mathrm{BN}} - E_{\mathrm{total}}|/E_{\mathrm{total}}$. Methods grouped by BN mode during unlearning.

Method	BN mode	E_W	E_{BN}	E_{total}	ρ	Pyth-error
Train-mode methods						
GA	train	57.4	24.4	64.1	-0.24	27.6%
SCRUB	train	46.9	3.1	49.6	-0.02	0.7%
SalUn	train	50.3	34.2	46.2	-0.46	83.2%
NoiseInject	train	58.8	7.3	41.6	-0.59	59.1%
IncompTeacher	train	42.6	4.8	27.9	-0.68	69.8%
GA+FT	train	49.7	1.1	53.4	$+0.17$	4.8%
SSD	train	47.2	21.5	27.7	-0.64	147.6%
Eval-mode methods						
BadTeacher†	eval	42.6	20.1	24.8	-0.65	152.6%

†BN frozen during unlearning; δ_{BN} measures the retain-data calibration gap between φ_0 and φ^* rather than unlearning-induced corruption (see prose).

Headline reading. Across the seven train-mode methods, mean $|\rho| = 0.40$ with maximum 0.68. The threshold for a Pythagorean form ($|\rho| < 0.2$ across all rows) is not met. The Pythagorean error exceeds 50% on three of seven train-mode methods and reaches 147.6% on SSD. This is the load-bearing finding: an additive squared-energy decomposition $\|\delta_W\|^2 + \|\delta_{\mathrm{BN}}\|^2 = \|\delta_W + \delta_{\mathrm{BN}}\|^2$ fails empirically, which is why the LP-space decomposition of Eq. (3) is presented as an algebraic identity rather than as an orthogonal Pythagorean split. The negative correlation pattern in train-mode BN-corrupted methods (gradient ascent shifts running stats toward forget while weight updates shift representations away from forget) is mechanistically consistent with the two-phenomena framing of Section 3.1—the weight-induced and stat-induced perturbations partially cancel, which is what produces the systematically negative ρ values on the three large-Pyth-error rows (SalUn, NoiseInject, SSD).

The eval-mode row carries a different interpretation. BadTeacher's row reflects the calibration gap between φ_0 and φ^*, not unlearning-induced corruption (since BadTeacher's stats never moved). The large E_{BN} and Pyth-error for BadTeacher are not evidence of BN corruption; they are evidence that recalibrating to retain-only stats induces a feature-space perturbation whose direction is anti-correlated with the unlearning weight perturbation ($\rho = -0.65$). We retain the row in Table 3 for completeness and to make this dependence explicit.

Why the LP-space decomposition is the load-bearing one. The decomposition of Eq. (3)—BN-residual $= \mathrm{LP}^{M} - \mathrm{LP}^{M_{\mathrm{cal}}}$, NC-residual $= \mathrm{LP}^{M_{\mathrm{cal}}} - \mathrm{LP}^{M_{\mathrm{retr}}}$—is uniform across BN modes. For eval-mode BadTeacher, the BN-residual on CIFAR-10 is $+0.37$pp (Table 2), correctly reflecting that BadTeacher's surface LP measurement is barely biased by its (frozen, retain-aligned) BN stats. For eval-mode PGU on CIFAR-10, the BN-residual is $+0.77$pp (Table 2). For train-mode SalUn, GA, and SSD on CIFAR-100, the BN-residuals are large and negative (-27.33, -29.67, -20.67), reflecting that BN measurement bias deflates the surface LP for these methods. The LP-space decomposition cleanly distinguishes regime from method, while the feature-space cross-correlation in Table 3 does not. This is one reason the LP-space form is the primary decomposition in the paper; the feature-space analysis is presented in this appendix solely to justify the algebraic-identity (rather than energy) framing of Eq. (3).

Table 4: CIFAR-100 single-class forget (class 0, $p = 0.01$; single seed). $\Delta\text{LP} = \text{LP}^{M_{\text{cal}}} - \text{LP}^{M} = -\text{BN-residual}$.

Method	Post-F%	LP-pre	LP-post	ΔLP	NCC-pre	NCC-post	Relearn AUC
Retrain	0.0	73.3	74.0	+0.7	72.3	69.0	0.633
SCRUB	0.0	81.7	86.7	+5.0	57.3	78.7	0.306
GA	1.0	58.0	87.7	+29.7	17.0	81.3	0.811
SalUn	1.0	60.7	88.0	+27.3	19.0	83.3	0.815
SSD	0.0	70.0	90.7	+20.7	49.3	89.1	0.862
GA+FT	1.0	88.7	88.7	0.0	–	–	0.824
IncompTeacher	75.0	87.0	90.0	+3.0	70.0	76.3	0.969
BadTeacher	28.0	75.0	89.0	+14.0	47.0	58.2	0.965
NoiseInject	85.0	88.4	91.1	+2.7	64.7	71.4	0.987
PGU	4.0	–	94.3	–	–	–	0.991

Table 5: CIFAR-100 5-class forget (classes 0–4, $p = 0.05$; single seed).

Method	Post-F%	LP-pre	LP-post	ΔLP	NCC-pre	NCC-post	Relearn AUC
Retrain	0.0	54.9	48.9	−6.0	51.5	44.6	0.596
SCRUB	0.0	65.6	66.6	+1.0	59.3	60.7	0.828
GA	0.0	43.8	45.9	+2.1	34.5	37.1	0.790
SalUn	0.0	42.1	44.4	+2.3	32.9	36.1	0.800

E CIFAR-100 Full Results

F Hyperparameters and Implementation Details

Architecture. ResNet-18 with CIFAR adaptation: 3×3 stem conv, no max-pool, $512 \to 10$ or 100 classifier. GroupNorm variant: all `nn.BatchNorm2d(c)` replaced by `nn.GroupNorm(num_groups=32, num_channels=c)`.

Method hyperparameters (CIFAR-10 single-class). SCRUB: lr=10^{-3}, steps=500, retain_steps=2, kl_weight=1.0. IncompTeacher: lr=10^{-3}, steps=500, $\tau_{\text{dist}} = 4.0$, $\alpha = 0.95$. SalUn: lr=10^{-3}, steps=200, threshold=0.5. GA: lr=10^{-3}, steps=500. SSD: $\lambda = 10000$. NoiseInject (Gaussian noise added to weights, implemented for this work): $\sigma_{\text{noise}} = 5.0$. GA+FT: GA-steps=200, FT-epochs=1, FT-lr=10^{-4}. PGU: see Appendix G.

Diagnostic protocol. BN recalibration: `torch.optim.swa_utils.update_bn` on retain loader, train mode, no gradients, mini-batch size $b = 128$. LP-50: `sklearn.linear_model.LogisticRegression(solver='lbfgs', max_iter=1000)`, 50 samples per class. NCC: class-mean features on 50 samples per class, ℓ_2 assignment. Relearn-AUC: fine-tune on 50 forget-class samples for 50 epochs (lr=10^{-4}), record forget accuracy curve, integrate normalized.

Additional experiment cohorts. The normalization diagnostics in Appendix B and the CIFAR experiments in Appendices O–S use a separately trained base model from the main evaluation. Comparisons are paired within each experiment; absolute values across cohorts are not directly comparable. Appendix N instead recomputes correlations from Table 1. The ImageNet-100 experiment uses its own pretrained ResNet-50.

Compute. All experiments on a single NVIDIA A6000 GPU ($\approx$ 30GB usable VRAM). Total wall-clock $\approx$ 90 GPU-hours.

GN per-method retain/LP detail. GN-ResNet retrains from scratch to 93.97% test accuracy. GN retain / GN LP-50: GA 94.2/91.2; SalUn 94.4/89.2; BadTeacher 58.9/87.1; IncompTeacher 93.4/95.7; SCRUB 54.9/67.8; GA+FT 94.3/92.0; NoiseInject 94.0/93.5; SSD 94.0/93.4; Retrain 90.8/67.2. BadTeacher and SCRUB exhibit retain degradation on *both* BN- and GN-ResNet—a method-level limitation, not a normalization confound. Five of the eight GN methods (GA, SalUn, GA+FT, NoiseInject, SSD) did not reach a forgetting regime at default hyperparameters on the GN

Table 6: GroupNorm control on CIFAR-10. BN ΔF reproduces Table 1's 3-seed means; GN ΔF is the same metric on a GN-ResNet retrained from scratch (single seed; mechanistic guarantee yields zero variance).

Method	BN ΔF (pp)	GN Pre-F%	GN ΔF (pp)
GA	$+0.0$	89.9	0.00
SalUn	$+62.1$	86.4	0.00
BadTeacher	$+78.2$	0.2	0.00
IncompTeacher	$+48.0$	0.1	0.00
SCRUB	$+0.0$	0.0	0.00
GA+FT	-0.2	91.4	0.00
NoiseInject	$+18.9$	93.6	0.00
SSD	$+62.4$	93.6	0.00
Retrain	$+0.0$	0.0	0.00

backbone (Pre-F $\geq 86\%$); for these methods $\Delta F = 0$ is mechanistically guaranteed by the absence of any φ to update, but is empirically tested only on the methods that did reach forgetting.

G PGU Re-Implementation Notes

We re-implement PGU [13] from the official reference. The official two-phase protocol (phase 1: 99 epochs; phase 2: up to 200 epochs with early stop at 3pp forget accuracy, weight decay 5×10^{-4}, entropy weight 0.1) estimates the retain activation covariance from a 5,000-sample augmented subset. Our re-implementation estimates that covariance using the full 45,000-sample retain set without augmentation—a strictly more accurate estimator yielding better-conditioned projection bases. The cleaner-covariance variant uses the single 99-epoch phase, whereas the official protocol includes the second phase and its early-stop criterion.

The two implementations produce different operating points on the same method. The official two-phase checkpoint (used in Table 2) reaches Pre-F%=6.9, Post-F%=21.4, $\Delta F = +14.5$pp—a partially-converged regime. The cleaner-covariance single-phase implementation (PGU† in Table 1) reaches Pre-F%=1.1, Post-F%=62.0, $\Delta F = +60.9$pp in 99 epochs: better projection bases drive forget accuracy almost to zero, exposing the BN misalignment phenomenon fully. This mirrors the GA step-count sweep: a method's apparent BN safety reflects how far into the weight-erasure regime its operating point has been pushed.

H GA Step-Count Sweep

Table 7: GA on CIFAR-10 (single-class, class 0). At 100 steps GA is BadTeacher-like (+76pp); at 500 steps GA is Retrain-like. Same method, entire BN-vulnerability spectrum as a function of one knob.

Steps	Pre-F%	Pre-Retain%	Post-F%	ΔF
100	9.1	78.0	85.3	$+76.2$
200	4.5	76.8	54.9	$+50.4$
300	0.8	74.0	15.8	$+15.0$
400	0.1	69.2	0.7	$+0.6$
500	0.0	65.2	0.0	0.0
600	0.0	61.4	0.0	0.0

GA's appearance in Table 1 as a $\Delta F = 0$ method reflects its 500-step operating point rather than architectural immunity. Within this fixed-composition sweep, increasing the number of ascent steps reduces the recalibration gap while degrading retain accuracy.

I Hyperparameter Sweep

At fixed batch composition, every tested configuration preserving retain accuracy at $\geq 85\%$ produces $\Delta F \in [57.8, 87.8]$pp. The low-gap configurations in this grid degrade retain accuracy. Thus learning-

Table 8: BadTeacher (left) and SalUn (right) hyperparameter sweep, ΔF in pp. BadTeacher: all 9 configurations preserve retain $\geq 85\%$; mean 78.3pp and population standard deviation 9.4pp across configurations. SalUn: $\diamondsuit$ retain-collapse boundary; † degenerate (retain 36.6%).

BadTeacher				SalUn			
LR×\Steps×	0.5 (250)	1 (500)	2 (1000)	LR×\Steps×	0.5 (100)	1 (200)	2 (400)
0.5×	+72.0	+57.8	+70.0	0.5×	+81.9	+83.0	+62.2
1×	+77.9	+83.0	+87.4	1×	+83.3	+62.0	$+0.4^{\diamondsuit}$
2×	+82.5	+86.0	+87.8	2×	+63.3	†DEGEN	$+0.3^{\diamondsuit}$

rate and step-count tuning within the tested grid does not remove the artifact while preserving retain utility. The separate retain-mixing intervention is reported in Appendix Q.

J Adversarial Recovery: Additional Results

J.1 Sub-100 budgets

Table 9: Sub-100 adversarial recovery on three BN-vulnerable methods (CIFAR-10, in-distribution unlabeled samples from CIFAR-10 test). BadTeacher and NoiseInject recovered to $\geq 88\%$ at B=10.

Method	B=10	B=25	B=50	B=100	B=500
BadTeacher	88.7	90.7	92.6	92.7	92.1
SalUn	54.4	54.8	57.4	58.0	57.6
NoiseInject	91.0	91.9	93.4	93.7	93.2

Even the smallest tested budget, ten unlabeled images, exposes the artifact in these BN-vulnerable checkpoints.

J.2 OOD recovery

We compare recovery using in-distribution unlabeled data (CIFAR-10 test set) against recovery using out-of-distribution data (CIFAR-100 raw pixels normalized with the CIFAR-10 mean/std the unlearned model was trained against). Both are evaluated at attacker budget B=100. The OOD/in-dist ratio is the OOD recovery accuracy divided by the in-dist recovery accuracy on the same method.

Table 10: OOD adversarial recovery (CIFAR-10, ResNet-18, B=100). *Pre-F*: surface forget-class accuracy on the unlearned checkpoint as released. *Auditor*: recalibration with full retain set (the in-house ceiling for an authorized auditor). *In-Dist*: attacker recovery using CIFAR-10 test images. *OOD*: attacker recovery using CIFAR-100 raw pixels under CIFAR-10 normalization. *Ratio*: OOD/In-Dist. The ratio range is 78%–95%, indicating that BN update is largely insensitive to the source of the attacker's unlabeled data: per-channel natural-image statistics correlate strongly across CIFAR-10 and CIFAR-100. GPM-W is a weight-projection unlearning variant evaluated only in this experiment.

Method	Pre-F%	Auditor%	In-Dist%	OOD%	OOD/In-Dist
GA+FT	61.6	60.0	59.1	47.4	80%
SalUn	3.1	58.5	57.6	45.1	78%
GPM-W	6.5	81.1	80.3	73.9	92%
BadTeacher	12.9	92.4	91.8	85.0	92%
IncompTeacher	6.6	54.2	52.7	47.5	90%
NoiseInject	69.5	93.5	93.0	88.5	95%
SSD	18.6	81.4	80.7	72.3	90%
PGU	6.9	21.4	22.1	17.8	81%

The ratio is bounded below by 78% (SalUn—a method whose surface Pre-F is already near zero, leaving the recoverable ceiling lower in absolute terms) and above by 95% (NoiseInject). No method falls below the 78% ratio. This rules out the natural defense "the attacker doesn't have access to anything close enough to the retain distribution": CIFAR-100 raw pixels are demonstrably close enough on the relevant axis (per-channel BN statistics), even though the two datasets share no class

labels and have different resolutions. The OOD attack is cheaper than the in-distribution attack only by a small margin; the threat model of Section 4.3 need not assume strong distributional knowledge.

K LP-Probe Training-Budget Bias

Table 11: LP probe training budget bias on CIFAR-10. Adam-50 underestimates LP-50 by 14–22pp for genuine encoder modifications, but is approximately unbiased for intact encoders.

Method	Adam-50	Adam-500	sklearn-lbfgs	Conv. budget	Gap
Retrain	58.7	76.9	76.5	500 ep	−17.8
SCRUB	67.9	81.8	81.9	500 ep	−14.0
GA	57.6	79.8	79.7	500 ep	−22.1
GA+FT	87.6	89.6	89.6	50 ep	−2.0
SalUn	87.0	88.9	89.0	50 ep	−2.0
BadTeacher	92.2	91.2	91.2	10 ep	+1.0

The bias is one-directional and favors methods with intact encoders—it makes intact-encoder methods look more separable and modified-encoder methods look more thoroughly forgotten than they are. We use sklearn-lbfgs throughout.

L Scale: ResNet-50 / Tiny-ImageNet

Table 12: Tiny-ImageNet / ResNet-50, single-class forget, three seeds. "Break" = breakthrough epoch in relearning fine-tune. "BN-Drift" = $\|\varphi - \varphi^*(\theta; \mathcal{D}_r)\|$ summed across BN layers. Relearn protocol: 50 epochs at LR=3×10^{-3} (calibrated for 200-class competitive threshold).

Method	Pre-F%	Post-F%	ΔF	Pre-R%	LP-50	ReAUC	Break	BN-Drift
Retrain	0.0	0.0	0.0	89.9	80.7	0.849	10	0.282
SCRUB	0.0	0.0	0.0	76.8	89.3	0.300	50	2.822
GA	0.0	0.0	0.0	2.1	82.7	0.300	50	9.384
GA+FT	0.8	0.8	0.0	95.0	91.3	0.791	10	0.273
SalUn	0.0	0.4	+0.4	89.0	89.3	0.788	10	8.989
IncompTeacher	0.0	10.0	+10.0	2.4	88.0	0.703	20	6.182
SSD	0.0	0.0	0.0	92.4	88.7	0.861	10	4.625
NoiseInject	84.2	99.0	**+14.8**	92.6	89.3	0.999	0	3.482
BadTeacher	0.6	82.8	**+82.2**	88.7	88.0	0.990	0	2.083
PGU	68.4	98.0	**+29.6**	57.9	88.0	0.999	0	2.020

Three regimes of $\Delta F = 0$. *Genuine forgetting:* SCRUB (Pre-R = 76.8%, breakthrough = 50). *Encoder-level failure (Gao–Lee regime):* GA+FT, SSD with high retain, high LP, and high ReAUC—head suppresses forget but encoder still encodes it. *Saturated weight erasure:* GA, IncompTeacher with destroyed retain. The SalUn anomaly ($\Delta F = 0.4$ vs. +62 on CIFAR-10) is a method-by-method scale variation we do not resolve. The headline reading is unchanged: where the BN illusion appears, it transfers.

Table 13: GA step-count sweep on Tiny-ImageNet/ResNet-50 (single seed, single-class).

Steps	Pre-F%	Post-F%	ΔF (pp)	Pre-R%	BN-Drift
50	0.0	84.6	+84.6	46.0	7.956
100	0.0	42.4	+42.4	44.4	7.962
200	0.0	0.2	0.2	40.8	7.950
400	0.0	0.0	0.0	26.8	7.860
800	0.0	0.0	0.0	0.5	8.421

The phase transition replicates: cliff edge moves earlier (500 steps → 200 steps) because 200-class competitive pressure destroys retain faster. BN-Drift saturates early (7.95–8.42) and stays roughly constant; ΔF tracks weight-encoded forget-class signal availability for recal to expose, not BN-Drift magnitude.

M LayerNorm Verification: ViT-S/16

LayerNorm computes statistics per-token per-instance at forward time; there are no running statistics. The recalibration operator is the identity on LN-based models by construction. We confirmed: ViT-S/16 (ImageNet-pretrained, fine-tuned on Tiny-ImageNet) tuned into a forgetting regime on 5 methods (GA, SalUn, BadTeacher, SCRUB, Retrain) yields $\Delta F = 0.00$pp uniformly. Per-step weight equality verified (parameter tensors bit-identical before/after no-op pass). PGU dropped (projection geometry is BN-specific). The GroupNorm control of Section 4.5 fixes architecture and varies normalization, providing the strict architecture-controlled falsification; the ViT/LN result is complementary, confirming the artifact is normalization-specific rather than dataset- or architecture-specific.

N Rank-Based Correlation Tests

Table 14 recomputes the associations in Table 1 with Pearson, Spearman, and Kendall tests, using the same ten rows, including Retrain. The pre-recalibration associations are descriptive: both are significant under Spearman, while the linear-probe pair is not significant under Kendall. The relevant conditional observation is that the four unlearning methods and Retrain with Pre-F $\leq 3\%$ have substantially different downstream leakage. Post-recalibration forget accuracy resolves these operating points and is strongly associated with both leakage measures.

Table 14: Associations between forget accuracy and leakage measures, calculated from Table 1 ($n = 10$, including Retrain). Each cell gives a coefficient and its two-sided p-value. Pre-recalibration association with relearning is strong under rank tests; post-recalibration association is significant for both leakage measures under all three tests.

Pair	Pearson (p)	Spearman (p)	Kendall (p)
Pre-F vs. ReAUC	0.63 (0.052)	0.89 (0.0006)	0.76 (0.003)
Pre-F vs. LP-50	0.53 (0.11)	0.66 (0.039)	0.46 (0.069)
Post-F vs. ReAUC	0.998 ($<10^{-6}$)	0.988 ($<10^{-6}$)	0.952 (0.00023)
Post-F vs. LP-50	0.928 (0.00011)	0.85 (0.002)	0.736 (0.0037)

O Membership Inference Before and After Recalibration

We evaluated four membership-inference attacks on four unlearning methods before and after recalibration: three black-box attacks based on confidence, entropy, and modified entropy, and a white-box gradient-norm attack following Nasr et al. [24]. On these checkpoints, recalibration changed forget accuracy by up to approximately 98 percentage points, while every attack AUC changed by at most 0.012 and remained within the confidence interval of the retrain reference, approximately 0.50. The original-model attack ceiling was 0.61. Within this measured range, recalibration has little effect on membership inference despite its large effect on forget accuracy. This supports the metric-specific interpretation of the BN illusion: the tested attacks do not exhibit the same normalization sensitivity as forget accuracy and linear probing.

P Instance-Wise Unlearning

We evaluated instance-wise forgetting with samples drawn randomly across all classes, rather than conditioned on a forget class. SalUn's recalibration gap falls to $+3.1$pp, compared with approximately $+78$pp for class-conditional forgetting on the same pipeline. BadTeacher retains a gap of $+89.6$pp. Random sampling removes the systematic class-conditional moment shift driving corruption, while changes in the weights can still cause misalignment with frozen BN statistics. Thus the class-conditional setting amplifies the corruption component, but is not required for the normalization artifact as a whole.

Q Batch Composition

We swept the retain fraction of the unlearning loader from 0 to 0.9. For SalUn, retain fractions of at least 0.25 reduced the recalibration gap from approximately $+75$pp to approximately 0pp in this sweep. BadTeacher retained a gap of $+89.5$pp at its reference-implementation mixing ratio. Retain mixing therefore mitigates the observed train-mode corruption, while eval-mode misalignment persists in the mixed-batch setting tested here. The learning-rate/step-count results in Appendix I hold batch composition fixed and address a different intervention.

R Activation Divergence Across BN Layers

We measured normalized class-conditional activation divergence in the pre-unlearning model at all 20 BN layers, for three forget classes including cat. The measured divergence was nonzero for every class and layer, with a minimum of 0.023. Early-layer values were 0.07–0.17, and late-layer values reached approximately 0.45. These measurements motivate the activation-moment separation condition in Theorem 2, including for early layers and visually correlated classes. They characterize the initial model; Eq. (7) describes how the targets subsequently move during unlearning. The theorem's assumption concerns activation moments, not an inequality between class labels or input distributions alone.

S A Purely Data-Based Method

RandomLabeling fine-tunes on forget samples with uniformly resampled labels, without a gradient-ascent component. It produces $\Delta F = +75.3$pp with train-mode BN and $+58.7$pp with eval-mode BN. The effect therefore extends beyond gradient-ascent methods. In train mode, the forward-pass EMA update is independent of the loss used for the weight update, as formalized by Theorem 2; freezing the buffers removes this update but does not prevent weight–normalization misalignment.

T ImageNet Scale

Table 15: ImageNet-100 at 224-pixel resolution, using an ImageNet-pretrained ResNet-50 and single-class forgetting (single seed; no retrain reference). Values are forget and retain accuracies, with ΔF in percentage points.

BadTeacher setting	Forget before	Forget after	ΔF	Retain before	Retain after
Full-retain recalibration	0.0%	92.0%	$+92.0$	71.6%	72.5%
Ten-image recovery	0.0%	82.0%	$+82.0$	71.6%	preserved

The normalization artifact persists at ImageNet resolution: full-retain recalibration raises forget accuracy from 0.0% to 92.0%, while retain accuracy rises from 71.6% to 72.5%. Ten unlabeled images recover 82.0% forget accuracy. Together with the $+82.2$pp BadTeacher result on Tiny-ImageNet (Appendix L), this extends the observed effect to larger-image ResNet-50 settings.

U Notation

Recalibration changes ΔF, $\Delta\mathrm{LP}$, and $\Delta\mathrm{NCC}$ are post minus pre. The decomposition uses $\text{BN-residual} = \mathrm{LP}^{M} - \mathrm{LP}^{M_{\mathrm{cal}}} = -\Delta\mathrm{LP}$, so a negative BN-residual denotes masked leakage. The mini-batch size b is distinct from the attacker image budget B, and distillation temperature τ_{dist} is distinct from Kendall's τ_{K} and the step count T.

Symbol	Meaning
$M = (\theta, \varphi)$	Model weights paired with BN running statistics
θ	All weights, including convolution kernels, BN affine parameters γ, β, and classifier weights
φ	Running means and variances stored in the checkpoint
$\mu^{\ell}, (\sigma^{\ell})^2$	Running mean and variance at BN layer ℓ
$\hat{\mu}^{\ell}, \hat{v}^{\ell}$	Mini-batch mean and variance at that layer; the variance estimator is specified in Appendix B.2
ℓ, L	BN layer index and number of BN layers
m	Exponential-moving-average momentum
$\mathcal{A}^{\ell}(\theta; \mathcal{D})$	Activation distribution at layer ℓ under weights θ on data $\mathcal{D}$
$f(x; \theta, \varphi)$	Forward map, depending on both weights and running statistics
$\mathcal{R}^{*}_{\mathcal{D}_r}$	Oracle retain-distribution recalibration operator
$\widehat{\mathcal{R}}_{N_r, b}$	Empirical recalibration on N_r retain images in batches of size b
$\varphi^{*}(\theta; \mathcal{D}_r)$	Oracle population activation moments at the current weights
$\widehat{\varphi}_{\mathrm{cal}}$	Empirically recalibrated running-statistic state
M_{cal}	Recalibrated model; the theoretical definition is $\mathcal{R}^{*}(M)$
M_{retr}	Retrain-from-scratch reference
$g(M)$	Metric that depends on the model's forward map
$\mathcal{D}_f, \mathcal{D}_r$	Forget and retain data, or their distributions when taking expectations
p	Fraction of training data designated for forgetting
N_r	Retain images used in recalibration
b, B	Recalibration mini-batch size; attacker's unlabeled-image budget
$T, \mathcal{L}, \mathcal{P}_W$	Unlearning step count; training loss; weight-gradient projection
τ_{dist}	Distillation temperature
Pre-F, Post-F	Forget accuracy before and after recalibration
ΔF	Post-F minus Pre-F, in percentage points
Pre-R, Post-R	Retain accuracy before and after recalibration
Ret-aw.	Retain-aware training objective or training stage
LP^{M}, LP-50	Forget-class linear-probe accuracy using 50 training samples per class, evaluated on the indicated model
$\Delta\mathrm{LP}$	$\mathrm{LP}^{M_{\mathrm{cal}}} - \mathrm{LP}^{M}$
BN-residual	$\mathrm{LP}^{M} - \mathrm{LP}^{M_{\mathrm{cal}}}$
NC-residual	$\mathrm{LP}^{M_{\mathrm{cal}}} - \mathrm{LP}^{M_{\mathrm{retr}}}$
ReAUC	Relearning area under the normalized accuracy curve
NCC, $\Delta\mathrm{NCC}$	Nearest-class-mean accuracy; its post-minus-pre change
$r, \rho_{\mathrm{S}}, \tau_{\mathrm{K}}$	Pearson, Spearman, and Kendall correlation coefficients
MIA	Membership-inference attack

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction explicitly enumerate three claims (mechanistic, diagnostic + security, empirical) restated in the conclusion, each tied to a section in the body. We are explicit about scope (BN-based architectures only, vision benchmarks) and about the metric-specific result (Appendix O), with no privacy guarantee inferred from $F\% = 0$.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4.6 is dedicated to limitations: graded findings rather than a clean partition, scope restricted to BN, what post-recalibration $F\% = 0$ does not establish, and the BadTeacher/SCRUB retain-degradation confound on GroupNorm. Appendix A additionally documents the unsuccessful static-target prediction and the distinction between structural non-interceptability and quantitative drift.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 1 is proved in Appendix B, including explicit finite-batch regularity assumptions and a distinction between empirical reference agreement and oracle error. Proposition 1 and its uniqueness argument are proved in Appendix C. Theorem 2 is restated and proved in Appendix A, with an activation-moment separation condition, a fixed-target bound, and a moving-target identity.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix F provides architecture, per-method hyperparameters, the diagnostic protocol, the adversarial recovery protocol, and the compute budget. Appendix G documents the PGU re-implementation. All baselines are public. Code is available at `https://github.com/aaryamam/BatchNormIllusion`

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/aaryamam/BatchNormIllusion`. All datasets used (CIFAR-10, CIFAR-100, Tiny-ImageNet, ImageNet-100) are public, and all baseline methods are described in the cited papers.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies the architecture, datasets, forget-class conditions, linear-probe protocol, and relearning-AUC protocol. Per-method hyperparameters are in Appendix F. The adversarial recovery protocol is fully specified in Section 4.4 and Appendix F.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports three-seed mean$\pm$std for the headline diagnostic on CIFAR-10 single-class. Tables 2 and the CIFAR-100 tables in Appendix E are single-seed but the underlying mechanism is the same and effect sizes are an order of magnitude larger than the seed-level variance observed in the multi-seed CIFAR-10 table. Appendix N reports Pearson, Spearman, and Kendall correlations with p-values. The near-zero-forget-accuracy comparison is stated conditionally. The depth-profile experiment in Appendix B and the ImageNet-100 experiment are single-seed. Adversarial recovery uses three random seeds per condition over an extended sweep $B \in \{10, 25, 50, 100, 500, 1000, 2500, 5000, 9000\}$; full results in Appendix J. The GroupNorm control is mechanistically guaranteed to produce zero artifact (no recalibratable state), and we observe exactly 0.00pp uniformly across eight unlearning methods and Retrain, consistent with that guarantee.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments run on a single shared NVIDIA A6000 GPU ($\approx$ 30GB usable VRAM); total wall-clock for the full reported set is approximately 90 GPU-hours (Appendix F).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The work uses only public benchmark datasets (CIFAR-10, CIFAR-100) and analyzes published methods. We do not introduce any model that produces harmful content. The security finding regarding adversarial recovery is disclosed in the spirit of identifying weaknesses in privacy-related claims, in line with responsible disclosure norms in adjacent literature (e.g., membership inference, model extraction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive: the paper identifies a measurement failure in evaluations of unlearning methods deployed for GDPR-style regulatory compliance, providing a low-cost diagnostic that future evaluations can adopt. Negative: the adversarial recovery experiment publishes a recipe for reversing forgetting in BN-vulnerable methods; we judge that disclosure is net-positive because (i) the underlying operation (`update_bn`) is a standard PyTorch primitive, (ii) the artifact is trivially discoverable by any motivated party, and (iii) regulators and deployers benefit from knowing the privacy promise is invalid before relying on it.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A]

    Justification: The paper does not release new datasets or models. The diagnostic and adversarial recovery procedures are based on standard PyTorch primitives applied to public methods on public datasets.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All baseline methods are cited to their original publications. CIFAR-10 and CIFAR-100 are publicly available datasets distributed by their original authors. The PGU re-implementation refers to the official repository at https://github.com/hnanhtuan/projected_gradient_unlearning.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new datasets or pretrained models. Code is available at https://github.com/aaryamam/BatchNormIllusion

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [N/A]

Justification: LLMs are not a component of the core methodology. Any use of LLMs in this work is limited to writing, editing, and formatting assistance, which per NeurIPS policy does not require declaration.