Loss is Not Behavior: A Unified Output-Space Analysis of Gradient-Based Machine Unlearning

Xingjian Zhao **Mohammad Mohammadi Amiri** **Malik Magdon-Ismail**
Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
{zhaox8,mohamm11}@rpi.edu, magdon@cs.rpi.edu

Abstract

Gradient-based machine unlearning methods typically "unlearn" by optimizing forget and retain losses, but similar losses do not necessarily imply similar output-space behavior. Prior work has shown that high forget loss alone does not certify that unlearning has produced retraining-like output behavior. We therefore ask: when a gradient-based unlearner differs from the model that would have been learned on only the retain data, what structure does the resulting output-space gap have? We derive closed-form expressions for this gap in linear regression, with an extension to nonlinear models via a generalized Gauss-Newton and local linearity approximations. Our analysis puts various gradient-based methods under a unified framework parametrized by geometry, reweighting, and projection. Two leverage matrices emerge that capture the forget-retain entanglement under the retrained model geometry: forget self-leverage and retain cross-leverage. We show both theoretically and empirically that these two leverage matrices drive the impact of the forget residual on the forget and retain outputs, and existing methods account for this geometry only partially. Our framework yields precomputable leverage diagnostics for auditing forget sets to identify what type of gradient-based unlearning is likely to be successful.

1 Introduction

Modern machine learning increasingly uses sensitive, copyrighted, or safety-critical data. As a result, machine unlearning, the post-hoc removal of a data's influence on the learned model, has become a practical and regulatory concern [2, 7]. The gold standard for machine unlearning is conceptually simple: retrain the model on the retained data after deleting the forget set. For large models and frequent deletion requests, however, exact retraining is usually too expensive, so practical unlearning methods seek more efficient approximations to this retrained gold standard model.

One family of approximations uses Newton-style updates to move directly toward the retrained optimum, but these methods depend on Hessian information and are often not scalable to large models [15, 18, 39]. The dominant practical alternative is gradient-based unlearning: gradient ascent (GA) increases loss on the forget set, while GradDiff, NPO, SimNPO, IMU, and projected variants add retain losses, per-sample weights, or projections to preserve utility [8, 9, 17, 24, 38, 40]. These methods are usually tuned and compared through forget and retain losses.

Recent work has established that loss-space, parameter-space agreement with retraining, and even finite output-imitation tests are insufficient evaluation criteria for unlearning [21, 37]. We verify this empirically in Figure 1: at matched forget loss or similar weights, different gradient-based unlearners can have very different output-space behavior and forget performance, and they can all differ substantially from the retrained model. This motivates output-space deviation from retraining as the evaluation surface, which leads to the open question we study here: what governs the structure of

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

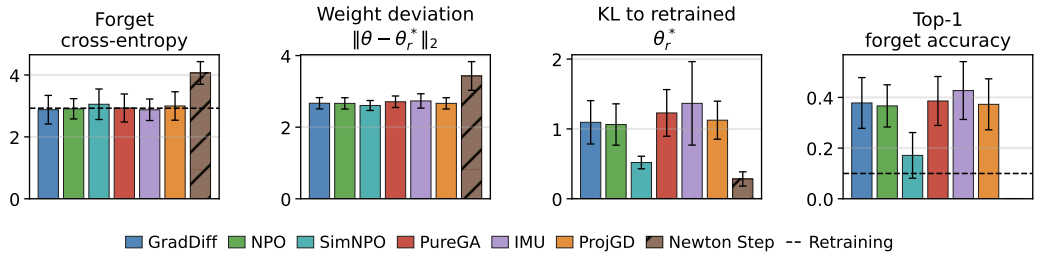


Figure 1: **Loss and weights are not behavior.** MNIST[20] with multi-layer perceptron (MLP) model for class forgetting with 10 seeds for various gradient-based methods and a Newton step unlearning. From left to right: forget cross-entropy (matched to the retrained reference, dashed), weight distance $\|\boldsymbol{\theta} - \boldsymbol{\theta}_r^\star\|_2$, $\mathrm{KL}(p_{\boldsymbol{\theta}_r^\star} \| p_{\boldsymbol{\theta}})$ on $\mathcal{D}_f$, and Top-1 forget accuracy. Setup is detailed in Appendix B.

this deviation for gradient-based unlearners? We show that the answer is a leverage geometry induced by the retrained model, and existing gradient-based variants control only parts of this geometry.

Our key contributions are:

1. **Unified gap formula.** We derive closed-form output-space gaps for gradient-based unlearners, exact in linear regression and extended to nonlinear models under a local generalized Gauss-Newton (GGN) approximation [31]. This gap shows a fundamental objective mismatch between common gradient-based methods and the gold standard.
2. **Two-leverage decomposition.** Our derivation reveals two important components of the gap: forget self-leverage, which captures the uniqueness of forget samples under the retrained model geometry, and retain cross-leverage, which captures how strongly forget-directed changes transfer into retain outputs.
3. **Forget-set diagnostics.** We validate each method's gap formula on MNIST and CIFAR-10, derive a closed-form sufficient rule for ρ, and empirically validate precomputable leverage diagnostics for LLM scale.

2 Related Work

2.1 Machine unlearning methods

Early formulations of machine unlearning formalized data deletion through exact retraining, certified removal, or influence-style approximations to retraining [2, 15, 18]. Newton-style methods are attractive because they target the retrained optimum directly, but they require Hessian or inverse-Hessian operations and rely on local approximations that can be costly or fragile at modern scale [11, 39]. A related line linearizes the network around a reference point (pretrained or core-trained weights) and forgets exactly in the resulting linear, typically squared-loss, model. Examples include NTK-based scrubbing [12], mixed-privacy forgetting [13], and parameter-efficient variants [22], made practical by fast empirical-NTK approximations [30, 35]. Within that linear model they are exact retraining, i.e. the Newton Step row of Table 1, so their output gap to $\boldsymbol{\theta}_r^\star$ is governed by linearization error and the locality limits of Section 4.3. While Newton-style methods are limited in practice, the retrained-model Hessian they use as a preconditioner is the same geometric object that, as we show, implicitly governs the output-space gap of gradient-based unlearners.

Most practical large language model (LLM) unlearning methods instead optimize surrogate objectives. GA on the forget loss is the simplest example [17]; GradDiff adds retain descent to improve utility preservation [24]; NPO and SimNPO stabilize GradDiff-style unlearning through preference-style losses and reweighting [8, 38]; IMU introduces influence-guided sample weights [9]; and GU-style projected methods reduce retain drift by projecting updates away from retain-gradient directions [40]. Another LLM baseline is Representation Misdirection Unlearning (RMU)[23], but it operates on intermediate representations rather than the output-loss gradient-based family analyzed in this work.

2.2 Limits of loss-space and parameter-space evaluation

Several lines of existing work show from different angles that loss, output imitation on a limited evaluation set, and parameter agreement are not enough to certify unlearning. In general prediction problems, predictive multiplicity shows that competing models with similar loss can assign conflicting predictions [28]. In unlearning, robustness studies show that superficial "forgetting" can be reversed by relearning or adversarial evaluation [5, 16, 21, 25, 26]. Other work argues that parameter-space or algorithmic equivalence criteria are themselves delicate: verification-error analyses motivate proximity to retraining, while auditable definitions and retrain-equivalence impossibility results expose limits of parameter- or path-oblivious formulations [33, 34, 37]. In the adjacent setting of catastrophic forgetting, Shenfeld et al. [32] show that the forward KL to the original model on the new task predicts the loss of prior-task performance. We share the output-space viewpoint, but measure per-example gaps to the *retrained* reference and decompose them by forget–retain geometry.

2.3 Retain-forget entanglement and geometry

The closest work to ours studies why gradient-based unlearning fails when forget and retain data interact. Mavrothalassitis et al. [29] attribute failures of ascent-based methods to statistical dependence between forget and retain sets. Other recent analyses study model resistance, stability, semantic entanglement, gradient conflict, and retain-gradient geometry [1, 3, 4, 36, 40]. These works diagnose retain-forget interaction through loss, gradient alignment, empirical instability, coherence, or semantic similarity. The main distinction of our work is the focus on output space: if an unlearned model is meant to approximate gold-standard retraining, then its outputs, not only its losses or parameters, are the relevant behavioral object. Our unified analysis reveals the matrices through which this interaction propagates to outputs: forget self-leverage and retain cross-leverage under the retrained model geometry.

3 Problem Setup

In this section, we formally define the unlearning and the gold-standard retraining objectives. We consider a training set $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$ with $|\mathcal{D}| = n$ and $|\mathcal{D}_f| = m$, where $\mathcal{D}_f$ is the *forget set* whose influence is to be removed and $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ is the *retain set*. Each sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ pairs an input $\boldsymbol{x}_i$ with a label $\boldsymbol{y}_i$. A model $f(\boldsymbol{\theta}, \boldsymbol{x}) \in \mathbb{R}^c$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$ is trained on $\mathcal{D}$ under the ℓ_2-regularized objective

$$\mathcal{L}(\boldsymbol{\theta}) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2, \qquad \lambda > 0, \tag{1}$$

where λ is the regularization strength and $\ell(\cdot)$ is the loss function. The full-data optimum is $\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

The gold standard. The gold-standard unlearned model is the minimizer of an analogous objective restricted to $\mathcal{D}_r$. We adopt the full-data-normalized retain objective

$$\mathcal{R}(\boldsymbol{\theta}) \;:=\; \frac{1}{n}\sum_{i\in\mathcal{D}_r} \ell(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2, \qquad \boldsymbol{\theta}_r^\star \;:=\; \arg\min_{\boldsymbol{\theta}} \mathcal{R}(\boldsymbol{\theta}), \tag{2}$$

which has the same minimizer as the standard $\frac{1}{n-m}$-normalized retain objective $\frac{1}{n-m}\sum_{i\in\mathcal{D}_r} \ell(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\lambda_r}{2}\|\boldsymbol{\theta}\|^2$ with a scaled regularization strength $\lambda_r = \frac{n}{n-m}\lambda$. We refer to $\boldsymbol{\theta}_r^\star$ as the *gold standard*: any unlearning method ideally returns a model whose induced behavior on arbitrary inputs matches that of $\boldsymbol{\theta}_r^\star$. Define also the full-data-normalized forget loss

$$\mathcal{F}(\boldsymbol{\theta}) \;:=\; \frac{1}{n}\sum_{j\in\mathcal{D}_f} \ell(\boldsymbol{\theta}, \boldsymbol{x}_j, \boldsymbol{y}_j), \tag{3}$$

so that $\mathcal{L} = \mathcal{R} + \mathcal{F}$.

Gradient-objective unlearning. Most existing optimization-based unlearners minimize a controlled retain–forget trade-off

$$\boldsymbol{\theta}_{\rho}^{\mathrm{GDiff}} \;:=\; \arg\min_{\boldsymbol{\theta}} \; \mathcal{R}(\boldsymbol{\theta}) - \rho\,\mathcal{F}(\boldsymbol{\theta}), \qquad \rho > 0, \tag{4}$$

recovering retraining at $\rho = 0$ and adding active forget-ascent for $\rho > 0$. Variants such as NPO and SimNPO replace $\mathcal{F}$ by a per-sample reweighted version, IMU incorporates per-sample influence weighting on the forget term, and Pure GA omits $\mathcal{R}$ entirely. Newton-style updates instead approximate $\boldsymbol{\theta}_r^\star$ in a single second-order step. Throughout, we treat $\boldsymbol{\theta}_r^\star$ as the reference solution and ask how each method's output behavior deviates from it. For nonconvex models the minimizers in (2) and (4) are not unique. We read them as local stationary points. In our MNIST and CIFAR-10 experiments $\boldsymbol{\theta}_r^\star$ is the warm-start retrain from $\boldsymbol{\theta}^\star$ (on TOFU, the released retain model), and in the fixed-point validation of Section 5.2 each method's solution is the stationary point on the branch through $\boldsymbol{\theta}_r^\star$. For a parameter $\boldsymbol{\theta}$ and evaluation inputs $\boldsymbol{X}_a$, we call $f(\boldsymbol{\theta}, \boldsymbol{X}_a) - f(\boldsymbol{\theta}_r^\star, \boldsymbol{X}_a)$ the *output-space gap* of $\boldsymbol{\theta}$ on $\boldsymbol{X}_a$, and summarize its size by per-coordinate RMS or output KL.

4 Output-Space Gap Analysis

In this section, we derive closed-form expressions for the output-space gap between gradient-based unlearners and the gold-standard retrained model $\boldsymbol{\theta}_r^\star$, first under a linear squared-loss setting and then under a local nonlinear approximation. Our following analysis reveals the key role of the retrained-model-geometry-based leverage matrices in structuring this gap, and it subsumes various gradient-based methods under a unified framework parametrized by geometry, reweighting, and projection.

4.1 Linear squared-loss analysis

We first analyze the output-space gap in a linear regression setting, which admits a closed-form solution and clean geometric interpretation that reveals two key factors influencing the gap. To make the gap explicit, we begin with the squared-loss linear setting $f(\boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\theta}$, for $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^d$, and $\ell(\boldsymbol{\theta}, \boldsymbol{x}, y) = \frac{1}{2}(y - \boldsymbol{x}^\top \boldsymbol{\theta})^2$. Let $\boldsymbol{X}_r \in \mathbb{R}^{(n-m)\times d}$ and $\boldsymbol{X}_f \in \mathbb{R}^{m\times d}$ be the retain and forget data matrices, respectively, with labels $\boldsymbol{y}_r \in \mathbb{R}^{n-m}$, $\boldsymbol{y}_f \in \mathbb{R}^m$. The full-data objective is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2n}\|\boldsymbol{y}_r - \boldsymbol{X}_r\boldsymbol{\theta}\|_2^2 + \frac{1}{2n}\|\boldsymbol{y}_f - \boldsymbol{X}_f\boldsymbol{\theta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2. \tag{5}$$

Define the *retain-Hessian* and the two natural *forget residuals*, evaluated at the full-data optimum and the retrained optimum, respectively, as

$$\boldsymbol{H}_r \;:=\; \tfrac{1}{n}\boldsymbol{X}_r^\top \boldsymbol{X}_r + \lambda \boldsymbol{I}_d, \qquad \boldsymbol{e}_f^{+} \;:=\; \boldsymbol{X}_f\boldsymbol{\theta}^\star - \boldsymbol{y}_f, \qquad \boldsymbol{e}_f^{-} \;:=\; \boldsymbol{X}_f\boldsymbol{\theta}_r^\star - \boldsymbol{y}_f. \tag{6}$$

The residual $\boldsymbol{e}_f^{+}$ is precomputable from $\boldsymbol{\theta}^\star$ alone, while $\boldsymbol{e}_f^{-}$ is oracle-evaluable from the retrained model. For any arbitrary evaluation data $\boldsymbol{X}_a$, we define the *leverage matrix*

$$\boldsymbol{K}_{af} \;:=\; \boldsymbol{X}_a\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top, \tag{7}$$

where as Theorem 1 shows, the leverage matrix $\boldsymbol{K}_{af}$ captures the influence of unlearning the forget samples on the output on arbitrary data $\boldsymbol{X}_a$ under the retrained model geometry. Two special cases of the leverage matrix are (i) *forget self-leverage* $\boldsymbol{K}_{ff} = \boldsymbol{X}_f\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top$: its diagonal entries capture the uniqueness of individual forget samples under this geometry, while its off-diagonal entries capture overlap among forget samples; and (ii) *retain cross-leverage* $\boldsymbol{K}_{rf} = \boldsymbol{X}_r\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top$ which measures how forget-directed changes transfer into outputs on the retain set.

The two residuals $\boldsymbol{e}_f^{+}$ and $\boldsymbol{e}_f^{-}$ are not independent quantities; they are linked by an explicit invertible transformation through $\boldsymbol{K}_{ff}$, which we use throughout to translate between equivalent forms.

Lemma 1 (Forget residual conversion)**.** *Suppose $\boldsymbol{H}_r$ is positive definite. Then the two forget residuals defined in* (6) *are linked by an explicit invertible transformation:*

$$\boldsymbol{e}_f^{-} \;=\; \left(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\right)\boldsymbol{e}_f^{+}, \qquad \boldsymbol{e}_f^{+} \;=\; \left(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\right)^{-1}\boldsymbol{e}_f^{-}. \tag{8}$$

In particular, $\boldsymbol{e}_f^{+} = \boldsymbol{0} \iff \boldsymbol{e}_f^{-} = \boldsymbol{0}$.

The proof is in Appendix A.2; it is a one-line consequence of the normal-equation difference between $\boldsymbol{\theta}^\star$ and $\boldsymbol{\theta}_r^\star$, evaluated at $\boldsymbol{X}_f$.

Theorem 1 (Retrained output deviation)**.** *Suppose $\boldsymbol{H}_r$ is positive definite. Then for any evaluation data $\boldsymbol{X}_a$, the output gap of the retrained model from the full-data optimum admits two equivalent closed forms:*

$$\boldsymbol{X}_a\boldsymbol{\theta}_r^\star - \boldsymbol{X}_a\boldsymbol{\theta}^\star \;=\; \tfrac{1}{n}\,\boldsymbol{K}_{af}\,\boldsymbol{e}_f^+ \tag{9}$$

$$\;=\; \tfrac{1}{n}\,\boldsymbol{K}_{af}\big(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\big)^{-1}\boldsymbol{e}_f^-, \tag{10}$$

where the second form follows from the first by Lemma 1.

The proof is at Appendix A.3. Equation (9) reveals that $\boldsymbol{K}_{af}$ captures how much of the forget residual $\boldsymbol{e}_f^+$ is transferred to the change in output on $\boldsymbol{X}_a$ when we retrain on $\mathcal{D}_r$, and the $\boldsymbol{e}_f^-$ form is its dual through $\boldsymbol{K}_{ff}$.

Theorem 2 (Linear GradDiff output gap)**.** *Suppose $\boldsymbol{H}_r - \frac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f$ is positive definite (equivalently $\frac{\rho}{n}\lambda_{\max}(\boldsymbol{K}_{ff}) < 1$, where $\lambda_{\max}(\cdot)$ captures the maximum eigenvalue, see Appendix A.5). Then for any evaluation data $\boldsymbol{X}_a$, the output gap of GradDiff from $\boldsymbol{\theta}_r^\star$ has the closed form*

$$\boldsymbol{X}_a\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{X}_a\boldsymbol{\theta}_r^\star \;=\; \frac{\rho}{n}\,\boldsymbol{K}_{af}\Big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\Big)^{-1}\Big(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\Big)\boldsymbol{e}_f^+. \tag{11}$$

Equivalently, in terms of the retrained-model forget residual $\boldsymbol{e}_f^- = \boldsymbol{X}_f\boldsymbol{\theta}_r^\star - \boldsymbol{y}_f$,

$$\boldsymbol{X}_a\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{X}_a\boldsymbol{\theta}_r^\star \;=\; \frac{\rho}{n}\,\boldsymbol{K}_{af}\Big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\Big)^{-1}\boldsymbol{e}_f^-. \tag{12}$$

The proof is at Appendix A.4. The two forms (11) and (12) are equivalent by the residual conversion identity in Lemma 1. Eq. (12) makes the structural decomposition cleanest: the retrained-model forget residual $\boldsymbol{e}_f^-$ is amplified by the inverse stability factor $(\boldsymbol{I}_m - \frac{\rho}{n}\boldsymbol{K}_{ff})^{-1}$, which captures how forget-directed perturbations are amplified under the retrained model geometry, and then transferred to the output gap on $\boldsymbol{X}_a$ by the leverage matrix $\boldsymbol{K}_{af}$. Eq. (11), on the other hand, reveals that the gap is computable entirely from $\boldsymbol{\theta}^\star$ without any retraining since we can compute the retain Hessian $\boldsymbol{H}_r = \boldsymbol{H} - \boldsymbol{H}_f$, where $\boldsymbol{H} = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}_d$ is the full Hessian and $\boldsymbol{H}_f = \frac{1}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f$ is the forget Hessian, making (11) an *a priori* diagnostic that bounds the output-space gap on any candidate forget set $\mathcal{D}_f$ before any unlearning is run. We adopt the Eq. (12) form in the following analysis for readability; all bounds and takeaways translate immediately to $\boldsymbol{e}_f^+$ via the same identity.

The two corollaries below summarize the practical norm bound and the separation between forget and retain output deviations.

Corollary 1 (Stability of the effective residual)**.** *Under the assumption of Theorem 2,*

$$\big\|\boldsymbol{X}_a\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{X}_a\boldsymbol{\theta}_r^\star\big\|_2 \;\le\; \frac{\rho}{n}\,\frac{\|\boldsymbol{K}_{af}\|_2\,\|\boldsymbol{e}_f^-\|_2}{1 - \frac{\rho}{n}\lambda_{\max}(\boldsymbol{K}_{ff})}, \tag{13}$$

or equivalently, in terms of the precomputable full-data residual,

$$\big\|\boldsymbol{X}_a\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{X}_a\boldsymbol{\theta}_r^\star\big\|_2 \;\le\; \frac{\rho}{n}\,\frac{\|\boldsymbol{K}_{af}\|_2\,\big(1 + \frac{1}{n}\lambda_{\max}(\boldsymbol{K}_{ff})\big)\,\|\boldsymbol{e}_f^+\|_2}{1 - \frac{\rho}{n}\lambda_{\max}(\boldsymbol{K}_{ff})}. \tag{14}$$

Corollary 2 (Forget and retain output deviations)**.** *Under the assumption of Theorem 2, the GradDiff output gap on the forget and retain sets is*

$$\boldsymbol{X}_f\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{X}_f\boldsymbol{\theta}_r^\star = \tfrac{\rho}{n}\,\boldsymbol{K}_{ff}\big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\big)^{-1}\boldsymbol{e}_f^-, \tag{15}$$

$$\boldsymbol{X}_r\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{X}_r\boldsymbol{\theta}_r^\star = \tfrac{\rho}{n}\,\boldsymbol{K}_{rf}\big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\big)^{-1}\boldsymbol{e}_f^-. \tag{16}$$

The corresponding $\boldsymbol{e}_f^+$ forms follow from Lemma 1.

Corollary 2 delivers the central design principle of this paper: $\boldsymbol{K}_{ff}$ **and** $\boldsymbol{K}_{rf}$ **capture qualitatively different geometric information about the forget set and act asymmetrically on the output**

Method	Output gap $\boldsymbol{X}_a(\boldsymbol{\theta}_\rho^{\text{method}} - \boldsymbol{\theta}_r^\star)$	Geometry
GradDiff	$\frac{\rho}{n}\boldsymbol{K}_{af}\left(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{K}_{ff}\right)^{-1}\boldsymbol{e}_f^-$	retrained Hessian
NPO	$\frac{\rho}{n}\boldsymbol{K}_{af}\left(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{W}_{\text{NPO}}^-\boldsymbol{K}_{ff}\right)^{-1}\boldsymbol{W}_{\text{NPO}}^-\boldsymbol{e}_f^-$	retrained Hessian
SimNPO	$\frac{\rho}{n}\boldsymbol{K}_{af}\left(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{W}_{\text{SimNPO}}^-\boldsymbol{K}_{ff}\right)^{-1}\boldsymbol{W}_{\text{SimNPO}}^-\boldsymbol{e}_f^-$	retrained Hessian
Pure GA	$-\boldsymbol{X}_a\boldsymbol{\theta}_r^\star - \frac{\rho}{n\lambda}\boldsymbol{G}_{af}\left(\boldsymbol{I} - \frac{\rho}{n\lambda}\boldsymbol{G}_{ff}\right)^{-1}\boldsymbol{y}_f$	Euclidean Gram
IMU	$-\boldsymbol{X}_a\boldsymbol{\theta}_r^\star - \frac{\rho}{n\lambda}\boldsymbol{G}_{af}\left(\boldsymbol{I} - \frac{\rho}{n\lambda}\boldsymbol{A}_{\text{IMU}}\boldsymbol{G}_{ff}\right)^{-1}\boldsymbol{A}_{\text{IMU}}\boldsymbol{y}_f$	Euclidean Gram
ProjGD (fixed)	$\frac{\rho}{n}\boldsymbol{K}_{af}^{\text{P}}\left(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{K}_{ff}^{\text{P}}\right)^{-1}\boldsymbol{e}_f^-$	retrained Hessian (projected)
Newton Step	$\mathbf{0}$	retrained Hessian (Newton)

Table 1: Theoretical output gaps from $\boldsymbol{\theta}_r^\star$ in the linear setting. Each row assumes that its displayed inverse exists. The GradDiff, Pure GA, fixed-weight IMU, fixed-projector ProjGD, and Newton Step rows are exact for the displayed linear objectives; NPO and SimNPO are frozen-weight local forms, with weights evaluated at $\boldsymbol{\theta}_r^\star$. The geometry column distinguishes methods that route forget perturbations through $\boldsymbol{H}_r^{-1}$ (top group: GradDiff, NPO, SimNPO, fixed-projector ProjGD) from those that use raw Euclidean Gram structure (Pure GA, IMU). Newton Step recovers $\boldsymbol{\theta}_r^\star$ exactly in linear regression. Proofs in Appendix A.6.

gap. At a high level, $\boldsymbol{K}_{ff} = \boldsymbol{X}_f\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top$ is an *intrinsic* property of the forget set under the retrained-model geometry: its diagonal entries measure how distinctive each forget sample is once the retain data is accounted for, and its eigenvalues control how strongly forget-directed perturbations are amplified by $(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{K}_{ff})^{-1}$. By contrast, $\boldsymbol{K}_{rf} = \boldsymbol{X}_r\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top$ is a *cross-set* alignment quantity, capturing how forget-directed updates project onto retain directions; it carries no information about the forget set in isolation. In the next section, we show that most existing gradient-based methods only partially account for the leverage geometry: they control the effective forget residual through reweighting or projection, but they do not fully eliminate the underlying forget self-leverage or retain cross-leverage.

4.2 Unified method view

So far our analysis has focused on the GradDiff method. In this section, we extend the analysis to a broader class of gradient-based unlearning methods. Each method differs only in (i) its *effective leverage matrix* (retrained-model Hessian $\boldsymbol{H}_r^{-1}$ vs. Euclidean $\lambda^{-1}\boldsymbol{I}$) and (ii) a per-sample *reweighting* matrix or fixed projector. Define the diagonal weighting matrices

$$\boldsymbol{W}_{\text{NPO}}^- = \operatorname{diag}\big(\sigma(z_j^-)\big)_{j=1}^m, \quad \boldsymbol{W}_{\text{SimNPO}}^- = \operatorname{diag}\big(\sigma(\tilde{z}_j^-)\big)_{j=1}^m, \quad \boldsymbol{A}_{\text{IMU}} = \operatorname{diag}(\alpha_j)_{j=1}^m, \tag{17}$$

where $\sigma(t) = (1+e^{-t})^{-1}$ is the logistic sigmoid, z_j^-, $\tilde{z}_j^-$ are the NPO/SimNPO logit arguments evaluated at $\boldsymbol{\theta}_r^\star$, and $\alpha_j = \|\boldsymbol{H}_{\boldsymbol{\theta}^\star}^{-1}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}^\star, \boldsymbol{x}_j, y_j)\|_2$ is IMU's per-sample influence norm, where $\boldsymbol{H}_{\boldsymbol{\theta}^\star}$ is the full-data Hessian at $\boldsymbol{\theta}^\star$. Explicitly, for the squared-loss surrogate, $z_j^- = -\beta_{\text{NPO}}\big(\ell_j(\boldsymbol{\theta}_r^\star) - \ell_j(\boldsymbol{\theta}_{\text{ref}})\big)$ and $\tilde{z}_j^- = -\beta_{\text{SimNPO}}\tilde{\ell}_j(\boldsymbol{\theta}_r^\star)$. We adopt the convention that the (Sim)NPO objectives use the prefactor ρ/β in front of the inner $\log\sigma(\beta\,\cdot)$ term (Appendix A.6), so that the inverse-temperature $\beta > 0$ that arises from differentiating the inner sigmoid cancels with the prefactor and the effective coefficient appearing in Table 1 is exactly ρ with the β-free weights above; under any other convention β would simply be absorbed into ρ. Let also $\boldsymbol{G}_{ff} = \boldsymbol{X}_f\boldsymbol{X}_f^\top$ and $\boldsymbol{G}_{af} = \boldsymbol{X}_a\boldsymbol{X}_f^\top$ be Euclidean Gram matrices. For ProjGD, let $\boldsymbol{P}_r$ be a fixed projector that removes retain-gradient-aligned directions from the forget gradient, and define the projected leverage matrices as

$$\boldsymbol{K}_{af}^{\text{P}} := \boldsymbol{X}_a\boldsymbol{H}_r^{-1}\boldsymbol{P}_r\boldsymbol{X}_f^\top, \qquad \boldsymbol{K}_{ff}^{\text{P}} := \boldsymbol{X}_f\boldsymbol{H}_r^{-1}\boldsymbol{P}_r\boldsymbol{X}_f^\top.$$

Proposition 1 (Unified linear gaps)**.** *For the fixed-weight or frozen-local linear surrogate specified above, whenever the displayed inverse exists, each method's stationary output gap from $\boldsymbol{\theta}_r^\star$ admits the closed form in Table 1. Invertibility ensures a unique stationary solution; positive-definite Hessians additionally ensure unique minima for the quadratic objectives.*

Table 1 shows that all methods share the same basic structure: an effective forget residual is formed by an inverse term on the forget set, then transferred to evaluation outputs through a method-specific geometry. GradDiff, NPO, SimNPO, and fixed-projector ProjGD use the retrained-Hessian geometry

through $\boldsymbol{K}$-matrices, while Pure GA and IMU use Euclidean Gram matrices $\boldsymbol{G}$. Reweighting and projection change the inverse term and transfer map, but they do not remove the output-space gap in general. This distinction governs which leverage diagnostic should predict each method's behavior.

Concretely, for a fixed forget residual:

- **Forget self-leverage governs amplification on *any* evaluation set.** In Eq. (12), $\boldsymbol{K}_{ff}$ enters the inverse stability factor $(\boldsymbol{I}_m - \frac{\rho}{n}\boldsymbol{K}_{ff})^{-1}$. At fixed ρ, increasing $\|\boldsymbol{K}_{ff}\|_2$ increases its worst-case amplification within the stable region; realized gaps also depend on residual alignment and the evaluation map. On the forget set, $\boldsymbol{K}_{ff}$ additionally appears in the prefactor of (15).
- **Retain cross-leverage governs transfer to retain outputs.** For fixed $\boldsymbol{K}_{ff}$ and forget residual, the retain-gap bound scales linearly with $\|\boldsymbol{K}_{rf}\|_2$, while the realized gap depends on alignment with the effective residual. The forget gap does not involve $\boldsymbol{K}_{rf}$, and gaps on other evaluation sets depend on their respective $\boldsymbol{K}_{af}$.

4.3 Local nonlinear approximation

In this section, we move away from the linear setting and show that the same structure persists for nonlinear models under a local approximation around $\boldsymbol{\theta}_r^\star$. Define for each forget sample j the per-sample Jacobian, output-gradient, and output-space Hessian

$$\boldsymbol{J}_j^- = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_r^\star, \boldsymbol{x}_j), \quad \boldsymbol{\delta}_j^- = \nabla_z \ell(z, y_j)\big|_{z=f(\boldsymbol{\theta}_r^\star, \boldsymbol{x}_j)}, \quad \boldsymbol{B}_j^- = \nabla_z^2 \ell(z, y_j)\big|_{z=f(\boldsymbol{\theta}_r^\star, \boldsymbol{x}_j)}, \tag{18}$$

with stacked counterparts $\boldsymbol{J}_f^- \in \mathbb{R}^{mc\times d}$, $\boldsymbol{\delta}_f^- \in \mathbb{R}^{mc}$, $\boldsymbol{B}_f^- \in \mathbb{R}^{mc\times mc}$ (block-diagonal), where c is the output dimension. Let $\boldsymbol{H}_r^-$ be the positive-definite local retain curvature used by the approximation (the exact Hessian in an exact Taylor model, or the retain GGN curvature in the experiments), and define the nonlinear leverage matrices

$$\boldsymbol{K}_{af}^{\mathrm{NL}} = \boldsymbol{J}_a^- \, (\boldsymbol{H}_r^-)^{-1} \, (\boldsymbol{J}_f^-)^\top, \qquad \boldsymbol{K}_{ff}^{\mathrm{NL}} = \boldsymbol{J}_f^- \, (\boldsymbol{H}_r^-)^{-1} \, (\boldsymbol{J}_f^-)^\top. \tag{19}$$

Also define the forget GGN curvature and the local stability matrix

$$\boldsymbol{H}_{f,\mathrm{G}}^- := \tfrac{1}{n}(\boldsymbol{J}_f^-)^\top \boldsymbol{B}_f^- \boldsymbol{J}_f^-, \qquad \boldsymbol{M}_\rho^- := \boldsymbol{H}_r^- - \rho \boldsymbol{H}_{f,\mathrm{G}}^-. \tag{20}$$

Proposition 2 (Local nonlinear GradDiff gap)**.** *Suppose $\mathcal{R}$ and $\mathcal{F}$ are twice continuously differentiable near $\boldsymbol{\theta}_r^\star$, and that $\boldsymbol{M}_\rho^-$ is positive definite. The local GGN/Taylor predictor for the GradDiff parameter gap is*

$$\widehat{\boldsymbol{\Delta}}_\rho := \rho(\boldsymbol{M}_\rho^-)^{-1}\nabla\mathcal{F}(\boldsymbol{\theta}_r^\star) = \tfrac{\rho}{n}(\boldsymbol{H}_r^-)^{-1}(\boldsymbol{J}_f^-)^\top \Big(\boldsymbol{I}_{mc} - \tfrac{\rho}{n}\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}\Big)^{-1}\boldsymbol{\delta}_f^-. \tag{21}$$

Consequently, for any evaluation set A with stacked Jacobian $\boldsymbol{J}_a^-$, the first-order predicted output gap is

$$\widehat{\Delta f}_A := \boldsymbol{J}_a^- \widehat{\boldsymbol{\Delta}}_\rho = \frac{\rho}{n}\, \boldsymbol{K}_{af}^{\mathrm{NL}} \Big(\boldsymbol{I}_{mc} - \tfrac{\rho}{n}\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}\Big)^{-1}\boldsymbol{\delta}_f^-, \tag{22}$$

with the nonlinear evaluation remainder controlled by the local-error discussion below.

The proof (Appendix A.7) is a Taylor expansion of the stationarity condition combined with the same push-through identity used in Theorem 2. Equation (22) preserves every interpretive feature of the linear case: the residual term $\boldsymbol{\delta}_f^-$ measures forget mispredictions at $\boldsymbol{\theta}_r^\star$, the leverage $\boldsymbol{K}_{af}^{\mathrm{NL}}$ transfers them to evaluation outputs, and the effective-residual factor becomes ill-conditioned as the smallest singular value of $\boldsymbol{I}_{mc} - (\rho/n)\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}$ approaches zero. Analogous local approximations for the other methods in Table 1 are obtained by the same expansion and are stated in Appendix A.9.

Approximation error. Equation (22) is a linear-response approximation to how the GradDiff stationary point moves under a forget-set reweighting. This is the same perturbative object studied by influence functions and the infinitesimal jackknife; finite-sample error bounds for such linearized reweighting approximations require local Hessian invertibility and Lipschitz/Taylor control of the gradients [10, 18]. In our notation, the error has the same form: it is amplified by the local conditioning of $\boldsymbol{M}_\rho^-$, is second order in the displacement when exact Hessians are used, and gains an additional first-order term when the exact Hessian is replaced by a GGN curvature. A representative bound and

Forget-set geometry under $\boldsymbol{\theta}_r^\star$		$\lambda_{\max}(\boldsymbol{K}_{ff})$	$\|\boldsymbol{K}_{rf}\|_F$	$\|\boldsymbol{e}_f^-\|$
Random (300 runs)		1.45×10^6	4.74×10^5	5.28
Class (30 runs)		2.27×10^6	6.26×10^5	24.25
Per-method output-space deviation from $\boldsymbol{\theta}_r^\star$ *(mean* $\pm$ *std)*				
Method	Regime	RMS_f	RMS_r	$\text{RMS}_f/\text{RMS}_r$
GradDiff	random	0.010 ± 0.016	0.009 ± 0.016	1.12
GradDiff	class	0.038 ± 0.037	0.031 ± 0.029	1.23
NPO	random	0.010 ± 0.015	0.009 ± 0.015	1.12
NPO	class	0.032 ± 0.032	0.025 ± 0.022	1.25
SimNPO	random	0.005 ± 0.009	0.004 ± 0.008	1.05
SimNPO	class	0.009 ± 0.011	0.009 ± 0.011	1.01
ProjGD	random	0.449 ± 0.036	0.438 ± 0.031	1.03
ProjGD	class	0.509 ± 0.065	0.440 ± 0.034	1.16
Newton Step	random	0.455 ± 0.037	0.416 ± 0.030	1.10
Newton Step	class	0.728 ± 0.074	0.470 ± 0.040	1.55

Table 2: **Class vs. random forgetting on MNIST.** Top: mean leverage estimates using retain-normalized GGN curvature at $\boldsymbol{\theta}_r^\star$. Bottom: per-coordinate logit RMS gaps from $\boldsymbol{\theta}_r^\star$ on $\mathcal{D}_f$ and $\mathcal{D}_r$ and their ratio. Class-regime forget accuracy is zero for GradDiff, NPO, SimNPO, and ProjGD, and 0.005 ± 0.020 for Newton Step. Loss ranges are summarized in the text; setup details in Appendix B.

derivation are given in Appendix A.8; we use it only to justify the locality of the diagnostic, not as a new tight guarantee.

Our formulas give exact stationary equations in linear regression, but introduce local approximations for nonlinear models. Moreover, practical implementations use stochastic finite-step updates, retain subsets, approximate Hessians, and sometimes dynamically updated weights or projectors. The approximation-error discussion above shows how these local errors enter the output gap; Section 5 shows that the leverage matrices remain informative in practice despite such implementation noise.

5 Experiments

In this section, we demonstrate how one may apply the leverage geometry analysis to understand and predict the behavior of unlearning methods in practice. We experiment on MNIST with a small MLP to (i) measure the leverage geometry $(\boldsymbol{K}_{ff}, \boldsymbol{K}_{rf}, \boldsymbol{e}_f^-)$ that Section 4.1 predicts to govern the output-space gap, in two qualitatively different forgetting regimes, and (ii) compare the five unlearners' output-space distance to the gold-standard retrained model in both regimes. The motivating evidence that loss and weight-space distance are not faithful evaluation surfaces is summarized in Figure 1; this section quantifies the geometry that underlies that dissociation. A scalable $\boldsymbol{\theta}^\star$-only leverage score at LLM scale on TOFU is reported in Appendix C, and additional validation of the exact linear formula, the pointwise nonlinear predictor, and the stability bound appears in Appendix D. Full reproducibility details are in Appendix B. Output deviations are reported as per-coordinate logit root-mean-square (RMS), $\sqrt{\mathbb{E}\|\Delta z\|_2^2/c}$, where Δz is the output logits difference measured against the reference $\boldsymbol{\theta}_r^\star$.

5.1 Class vs. random forgetting: leverage geometry and method behavior

We construct two regimes of forget set on MNIST. In the *class* regime, $\mathcal{D}_f$ contains all instances of one class in the $4{,}000$-sample training subset (so $|\mathcal{D}_f| \approx 350$–$490$, reflecting natural class-size variation across the seed's subsample rather than multiple set sizes per class); in the *random* regime, $\mathcal{D}_f$ is a uniformly random training subsample of size $|\mathcal{D}_f| = 200$. For each forget set we form $\boldsymbol{\theta}_r^\star$ by fixed-step retraining and estimate the leverage geometry using a retain-normalized GGN surrogate (Appendix B). The four iterative unlearners take 500 full-batch Adam steps; Newton Step takes one update. Aggregates are over 10 classes $\times$ 3 seeds (class regime) and 100 random sets $\times$ 3 seeds (random regime).

The bottom block of Table 2 reports the central observable, RMS_f and RMS_r from $\boldsymbol{\theta}_r^\star$. Three takeaways follow from this comparison.

Data	Method	Random forget set				Class forget set			
		P_f^K	P_f^G	P_r^K	P_r^G	P_f^K	P_f^G	P_r^K	P_r^G
MNIST	GradDiff	0.98±0.02	0.83±0.06	0.72±0.10	0.40±0.06	0.98±0.00	0.81±0.04	0.95±0.02	0.59±0.09
	NPO	0.98±0.02	0.83±0.06	0.72±0.10	0.40±0.05	0.98±0.00	0.81±0.04	0.95±0.02	0.58±0.09
	SimNPO	0.94±0.12	0.52±0.22	0.74±0.09	0.24±0.05	0.98±0.01	-0.03±0.13	0.97±0.01	0.00±0.18
	ProjGD	0.96±0.02	0.94±0.02	0.75±0.01	0.83±0.03	0.95±0.01	0.97±0.01	0.87±0.03	0.87±0.05
	Pure GA	0.01±0.01	0.02±0.02	0.05±0.03	0.04±0.04	-0.36±0.06	-0.39±0.05	-0.11±0.03	-0.23±0.07
CIFAR-10	GradDiff	0.89±0.03	0.74±0.04	0.82±0.05	0.63±0.06	0.97±0.01	0.94±0.02	0.87±0.09	0.74±0.07
	NPO	0.89±0.03	0.74±0.04	0.82±0.05	0.63±0.06	0.97±0.01	0.94±0.02	0.87±0.09	0.74±0.07
	SimNPO	0.91±0.01	0.60±0.11	0.86±0.01	0.51±0.09	0.97±0.01	-0.14±0.17	0.86±0.10	0.04±0.17
	ProjGD	0.79±0.03	0.74±0.05	0.71±0.04	0.67±0.06	0.97±0.01	0.98±0.01	0.87±0.02	0.80±0.04
	Pure GA	0.22±0.07	0.26±0.07	0.29±0.08	0.27±0.08	-0.51±0.07	-0.45±0.08	-0.04±0.01	-0.23±0.06

Table 3: **Local gap validation and controls.** Per-logit Pearson correlation between predicted and actual gaps on $\mathcal{D}_f$ (f) and 256 retain anchors (r), using retain-GGN (P^K) or Euclidean-Gram (P^G) geometry; mean±sd over 10 seeds at $s = 0.1$. GradDiff/NPO/SimNPO test the local predictors of Table 4; NPO/SimNPO predictors freeze reference weights, and SimNPO uses the weighted ρ rule. ProjGD compares its projected-gradient endpoint with the unprojected predictor; Pure GA is a nonlocal control.

(i) The forget channel is consistently larger than the retain channel. For every method and every regime, $\text{RMS}_f \geq \text{RMS}_r$, with ratios 1.0–1.55. Equations (15)–(16) distinguish these channels: $\boldsymbol{K}_{ff}$ enters the forget gap both as a prefactor and inside the inverse stability factor, while $\boldsymbol{K}_{rf}$ transfers the effective residual to retain outputs. The observed ordering also depends on residual alignment.

(ii) Class forgetting concentrates the residual; the leverage matrices grow much less. Class-regime forget CE is similar for the four iterative methods (4.368–4.404), all with zero forget accuracy, yet RMS_f differs by about 58× between SimNPO (0.009) and ProjGD (0.509). Newton has lower CE (2.972), so its larger gap illustrates accuracy mismatch only. We summarize these metrics here and in Appendix B rather than add columns to Table 2. Going from random to class forgetting, $\|\boldsymbol{e}_f^-\|$ grows by 4.59×, whereas the surrogate estimates of $\lambda_{\max}(\boldsymbol{K}_{ff})$ and $\|\boldsymbol{K}_{rf}\|_F$ grow by 1.57× and 1.32×: the residual changes much more than the leverage norms. The iterative methods (GradDiff, NPO, SimNPO) scale RMS_f by 1.8–3.8×, consistent with the residual ratio modulated by geometry and method weights. SimNPO separates from NPO mainly in the class regime. There the forget CE near $\boldsymbol{\theta}_r^\star$ is large (≈ 4.4 vs. 0.33 for random sets), so the default $\beta = 2.5$ saturates its weights ($\bar{w} \approx 10^{-4}$), switching the forget term off and leaving the iterate near $\boldsymbol{\theta}_r^\star$, whereas NPO at $\beta = 0.1$ keeps nearly uniform $O(1)$ weights and so behaves like GradDiff with a rescaled ρ. Swapping the two β values swaps the behavior (Table 11). ProjGD and the Newton step have larger output gaps; Appendix D discusses their behavior. Because Eq. (14) bounds the gap from $\boldsymbol{\theta}^\star$ quantities alone, the leverage geometry can still serve as an a-priori predictive diagnostic when a retrained reference is unavailable.

(iii) Geometry motivates a per-set ρ rule. The stability condition of Theorem 2 requires $s := (\rho/n)\lambda_{\max}(\boldsymbol{K}_{ff}) < 1$, motivating $\rho = s\,n/\lambda_{\max}(\boldsymbol{K}_{ff})$. In Table 2, we use $s = 0.3$ with the surrogate kernels described above. This is an empirical heuristic: the per-set ρ varies by an order of magnitude across random forget sets, yet optimization is stable across all 330 forget sets without per-set grid search. The positive-definiteness guarantee applies to the consistently normalized, unprojected frozen GGN models in Appendix B; locality up to $s = 2$ remains empirical. Saturated SimNPO weights motivate the weighted kernel in Section 5.2.

LLM-scale evidence. On TOFU with Llama-3.2-1B-Instruct, K-channel scores computed once at $\boldsymbol{\theta}^\star$ in a LoRA chart rank, before any unlearning run, the retain examples that GradDiff, NPO, and SimNPO will disturb most (Spearman 0.26–0.49 over three seeds, far above the permutation null) and, less sharply, the forget examples they will treat least like retraining (0.14–0.29, null for NPO on forget01; Appendix C). They are a cheap per-example signal of unlearning difficulty and spillover, on par with per-example loss and gradient-norm baselines. With curvature restricted to a LoRA chart and early-stopped endpoints they do not yet predict the size of the gap. Whether richer curvature estimates or endpoints nearer stationarity improve these diagnostics remains open.

5.2 Local gap validation and controls

We validate the local GradDiff, NPO, and SimNPO predictors on MNIST and a head over frozen CIFAR-10 features (Appendix B). Their stationary endpoints are reached by Newton continuation from $\boldsymbol{\theta}_r^\star$ with $s = 0.1$, and the corresponding predictors are compared with actual logit gaps on $\mathcal{D}_f$ and 256 retain anchors (Table 3). These endpoints lie within 5×10^{-3} of $\boldsymbol{\theta}_r^\star$ in parameter norm. Predictions track both channels and datasets ($P_f^K = 0.89$–0.98, $P_r^K = 0.72$–0.97, relative ℓ_2 error 0.25–0.84) and beat the Euclidean control throughout. As a control, ProjGD uses projected-gradient dynamics with a recomputed projector and is compared with the unprojected predictor (22): their directions correlate ($P_f^K = 0.79$–0.97), but the actual gap is about 7–$15\times$ smaller. Pure GA and IMU provide controls outside the local regime: their endpoints have output KL 0.7–3 nats from $\boldsymbol{\theta}_r^\star$, versus at most 10^{-5} for the retain-anchored methods, and their predictors are uninformative (Section 6).

The local validation concerns the stationary branch through the specified retrained reference; initialization at the original model can reach a different branch where the predictor fails, as demonstrated in Appendix B.

6 Limitations and Discussion

Our analysis is exact for linear regression, but nonlinear approximations and practical implementations introduce additional error. Although Appendix A.8 analyzes this error and Section 5.2 and Appendix D validate the approximation empirically, the leverage matrices should be read as output-space risk diagnostics rather than exact predictors, and they are local to the specified retrained reference and to the stationary branch through it (Section 3, Appendix B). Relatedly, Pure GA and IMU have no term anchoring their iterates near $\boldsymbol{\theta}^\star$. Their rows in Table 1 remain exact but contain $-\boldsymbol{X}_a\boldsymbol{\theta}_r^\star$, so their gap is of the order of the reference outputs and the local predictor is uninformative at their converged endpoints under either geometry (Section 5.2), while their early-stopped operating points (Figure 1) are not stationary points and lie outside our analysis. Computing the full gap is also infeasible for large models: Appendix C.2 shows that LoRA-space diagonal-GGN approximations of $\boldsymbol{K}_{ff}$ and $\boldsymbol{K}_{rf}$ are informative on TOFU, but their modest correlations leave better estimators as an open problem.

These limitations suggest future methods that account for both forget self-leverage and retain cross-leverage when choosing ρ, reweighting forget samples, or projecting updates, rather than optimizing losses alone. The precomputable $\boldsymbol{e}_f^+$ form of Theorem 1 also suggests LLM-scale audits that combine released-model forget residuals with sketched or low-rank $\boldsymbol{K}_{ff}$ estimates.

The broader impact is primarily diagnostic: output-space auditing can improve privacy, data-removal, copyright, and safety evaluations, but approximate diagnostics should complement behavioral evaluation, privacy auditing, and transparent assumptions rather than serve as compliance certificates.

7 Conclusion

We studied gradient-based machine unlearning through the output-space gap to gold-standard retraining. In linear regression, this gap admits a closed form; under a local GGN approximation, the same structure extends to nonlinear models. The resulting framework unifies many common gradient-based unlearners and identifies two leverage matrices that govern how forget-set residuals propagate: forget self-leverage, which controls behavior on the forget set, and retain cross-leverage, which controls transfer into retain outputs. Together, these matrices provide a sharper language for comparing existing methods, auditing forget sets, and designing better unlearning algorithms.

Acknowledgments and Disclosure of Funding

This work received no third-party funding, and we declare no competing interests.

References

[1] Sadia Asif and Mohammad Mohammadi Amiri. Ofmu: Optimization-driven framework for machine unlearning. *arXiv preprint arXiv:2509.22483*, 2025.

[2] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[3] Wei-Kai Chang and Rajiv Khanna. Why some models resist unlearning: A linear stability perspective. *arXiv preprint arXiv:2602.02986*, 2026.

[4] Jingpu Cheng, Ping Liu, Qianxiao Li, and Chi Zhang. Machine unlearning under retain-forget entanglement. *arXiv preprint arXiv:2603.26569*, 2026.

[5] Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

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

[7] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). Official Journal of the European Union, OJ L 119, 4.5.2016, pp. 1–88, 2016. URL `https://eur-lex.europa.eu/eli/reg/2016/679/oj`. Accessed via EUR-Lex.

[8] Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*, 2024.

[9] Xindi Fan, Jing Wu, Mingyi Zhou, Pengwei Liang, and Dinh Phung. IMU: Influence-guided machine unlearning. *arXiv preprint arXiv:2508.01620v1*, 2025. URL `https://arxiv.org/abs/2508.01620v1`.

[10] Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147. PMLR, 2019.

[11] 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 (CVPR)*, pages 9304–9312, 2020.

[12] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision (ECCV)*, 2020.

[13] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

[15] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[16] Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Jogging the memory of unlearned models through targeted relearning attacks. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024.

[17] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, 2023.

[18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[20] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. Dataset, n.d. URL `https://huggingface.co/datasets/ylecun/mnist`. Accessed October 28, 2026.

[21] Bruce W Lee, Addie Foote, Alex Infanger, Leni Shor, Harish Kamath, Jacob Goldman-Wetzler, Bryce Woodworth, Alex Cloud, and Alexander Matt Turner. Distillation robustifies unlearning. *arXiv preprint arXiv:2506.06278*, 2025.

[22] Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. Fast-NTK: Parameter-efficient unlearning for large-scale models. *arXiv preprint arXiv:2312.14923*, 2023.

[23] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[24] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[25] Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*, 2024.

[26] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

[27] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024.

[28] Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *International conference on machine learning*, pages 6765–6774. PMLR, 2020.

[29] Ioannis Mavrothalassitis, Pol Puigdemont, Noam Itzhak Levi, and Volkan Cevher. Ascent fails to forget. *arXiv preprint arXiv:2509.26427*, 2025.

[30] Mohamad Amin Mohamadi, Wonho Bae, and Danica J Sutherland. A fast, well-founded approximation to the empirical neural tangent kernel. In *International Conference on Machine Learning*. PMLR, 2023.

[31] Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14(7):1723–1738, 2002.

[32] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[33] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022.

[34] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX security symposium (USENIX Security 22)*, pages 4007–4022, 2022.

[35] Alexander Wei, Wei Hu, and Jacob Steinhardt. More than a toy: Random matrix models predict how real-world neural representations generalize. In *International Conference on Machine Learning*. PMLR, 2022.

[36] Zeguan Xiao, Siqing Li, Yong Wang, Xuetao Wei, Jian Yang, Yun Chen, and Guanhua Chen. Modeling llm unlearning as an asymmetric two-task learning problem. *arXiv preprint arXiv:2604.14808*, 2026.

[37] Jiatong Yu, Yinghui He, Anirudh Goyal, and Sanjeev Arora. On the impossibility of retrain equivalence in machine unlearning. *arXiv preprint arXiv:2510.16629*, 2025.

[38] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

[39] Xingjian Zhao, Mohammad Mohammadi Amiri, and Malik Magdon-Ismail. Win-u: Woodbury-informed newton-unlearning as a retain-free machine unlearning framework. *arXiv preprint arXiv:2604.13438*, 2026.

[40] Duo Zhou, Yuji Zhang, Tianxin Wei, Ruizhong Qiu, Ke Yang, Xiao Lin, Cheng Qian, Jingrui He, Hanghang Tong, Chengxiang Zhai, et al. Geometric-disentangelment unlearning. *arXiv preprint arXiv:2511.17100*, 2025.

A Proofs and Detailed Derivations

A.1 Auxiliary push-through identity

Lemma 2 (Push-through identity)**.** *Let $\boldsymbol{A}$, $\boldsymbol{U}$, $\boldsymbol{C}$, and $\boldsymbol{V}$ be conformable matrices. If $\boldsymbol{A}$ and $\boldsymbol{I} - \boldsymbol{C}\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}$ are invertible, then $\boldsymbol{A} - \boldsymbol{U}\boldsymbol{C}\boldsymbol{V}$ is invertible and*

$$(\boldsymbol{A} - \boldsymbol{U}\boldsymbol{C}\boldsymbol{V})^{-1}\boldsymbol{U} = \boldsymbol{A}^{-1}\boldsymbol{U}\big(\boldsymbol{I} - \boldsymbol{C}\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\big)^{-1}.$$

Proof. By Sylvester's determinant identity, $\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{U}\boldsymbol{C}\boldsymbol{V}$ is invertible whenever $\boldsymbol{I} - \boldsymbol{C}\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}$ is invertible; hence $\boldsymbol{A} - \boldsymbol{U}\boldsymbol{C}\boldsymbol{V} = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{A}^{-1}\boldsymbol{U}\boldsymbol{C}\boldsymbol{V})$ is invertible. Start from

$$(\boldsymbol{A} - \boldsymbol{U}\boldsymbol{C}\boldsymbol{V})\boldsymbol{A}^{-1}\boldsymbol{U} = \boldsymbol{U} - \boldsymbol{U}\boldsymbol{C}\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U} = \boldsymbol{U}\big(\boldsymbol{I} - \boldsymbol{C}\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\big).$$

Multiplying on the left by $(\boldsymbol{A} - \boldsymbol{U}\boldsymbol{C}\boldsymbol{V})^{-1}$ and on the right by $\big(\boldsymbol{I} - \boldsymbol{C}\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\big)^{-1}$ gives the result. □

A.2 Proof of Lemma 1

In the linear squared-loss setting the full-data objective is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2n}\|\boldsymbol{y}_r - \boldsymbol{X}_r\boldsymbol{\theta}\|_2^2 + \frac{1}{2n}\|\boldsymbol{y}_f - \boldsymbol{X}_f\boldsymbol{\theta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2,$$

whose stationarity condition $\nabla\mathcal{L}(\boldsymbol{\theta}^\star) = \boldsymbol{0}$ reads

$$\Big(\boldsymbol{H}_r + \tfrac{1}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f\Big)\boldsymbol{\theta}^\star \;=\; \boldsymbol{b}_r + \tfrac{1}{n}\boldsymbol{X}_f^\top\boldsymbol{y}_f, \qquad \boldsymbol{b}_r := \tfrac{1}{n}\boldsymbol{X}_r^\top\boldsymbol{y}_r. \tag{23}$$

The retrained optimum satisfies $\nabla\mathcal{R}(\boldsymbol{\theta}_r^\star) = \boldsymbol{0}$, i.e. $\boldsymbol{H}_r\boldsymbol{\theta}_r^\star = \boldsymbol{b}_r$. Rearranging (23),

$$\boldsymbol{H}_r\boldsymbol{\theta}^\star \;=\; \boldsymbol{b}_r - \tfrac{1}{n}\boldsymbol{X}_f^\top\big(\boldsymbol{X}_f\boldsymbol{\theta}^\star - \boldsymbol{y}_f\big) \;=\; \boldsymbol{b}_r - \tfrac{1}{n}\boldsymbol{X}_f^\top\boldsymbol{e}_f^+.$$

Subtracting from $\boldsymbol{H}_r\boldsymbol{\theta}_r^\star = \boldsymbol{b}_r$ and using positive-definiteness of $\boldsymbol{H}_r$ gives the *parameter shift*

$$\boldsymbol{H}_r(\boldsymbol{\theta}_r^\star - \boldsymbol{\theta}^\star) \;=\; \tfrac{1}{n}\boldsymbol{X}_f^\top\boldsymbol{e}_f^+ \quad\Longrightarrow\quad \boldsymbol{\theta}_r^\star - \boldsymbol{\theta}^\star \;=\; \tfrac{1}{n}\,\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top\boldsymbol{e}_f^+. \tag{24}$$

Evaluating (24) at $\boldsymbol{X}_f$,

$$\boldsymbol{e}_f^- - \boldsymbol{e}_f^+ \;=\; \boldsymbol{X}_f(\boldsymbol{\theta}_r^\star - \boldsymbol{\theta}^\star) \;=\; \tfrac{1}{n}\boldsymbol{X}_f\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top\boldsymbol{e}_f^+ \;=\; \tfrac{1}{n}\boldsymbol{K}_{ff}\,\boldsymbol{e}_f^+.$$

Rearranging gives $\boldsymbol{e}_f^- = \big(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\big)\boldsymbol{e}_f^+$. Since $\boldsymbol{K}_{ff}$ is positive semidefinite, $\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff} \succeq \boldsymbol{I}_m$ is invertible, yielding the inverse direction $\boldsymbol{e}_f^+ = \big(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\big)^{-1}\boldsymbol{e}_f^-$ and the equivalence $\boldsymbol{e}_f^+ = \boldsymbol{0} \iff \boldsymbol{e}_f^- = \boldsymbol{0}$. □

A.3 Proof of Theorem 1

Left-multiplying the parameter shift (24) (established in the proof of Lemma 1) by $\boldsymbol{X}_a$ and using the definition $\boldsymbol{K}_{af} = \boldsymbol{X}_a \boldsymbol{H}_r^{-1} \boldsymbol{X}_f^\top$ yields (9):

$$\boldsymbol{X}_a \boldsymbol{\theta}_r^\star - \boldsymbol{X}_a \boldsymbol{\theta}^\star \;=\; \tfrac{1}{n}\, \boldsymbol{K}_{af}\, \boldsymbol{e}_f^+.$$

The equivalent retrained-residual form (10) follows by substituting Lemma 1 into (9). □

A.4 Proof of Theorem 2

In the linear squared-loss setting,

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{2n}\|\boldsymbol{y}_r - \boldsymbol{X}_r\boldsymbol{\theta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \qquad \mathcal{F}(\boldsymbol{\theta}) = \frac{1}{2n}\|\boldsymbol{y}_f - \boldsymbol{X}_f\boldsymbol{\theta}\|_2^2.$$

Hence

$$\nabla\mathcal{R}(\boldsymbol{\theta}) = \Big(\tfrac{1}{n}\boldsymbol{X}_r^\top\boldsymbol{X}_r + \lambda\boldsymbol{I}\Big)\boldsymbol{\theta} - \tfrac{1}{n}\boldsymbol{X}_r^\top\boldsymbol{y}_r = \boldsymbol{H}_r\boldsymbol{\theta} - \boldsymbol{b}_r, \qquad \nabla\mathcal{F}(\boldsymbol{\theta}) = \tfrac{1}{n}\boldsymbol{X}_f^\top(\boldsymbol{X}_f\boldsymbol{\theta} - \boldsymbol{y}_f),$$

where $\boldsymbol{b}_r := \frac{1}{n}\boldsymbol{X}_r^\top\boldsymbol{y}_r$. The first-order condition for $\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}$ is

$$\nabla\mathcal{R}(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}) - \rho\nabla\mathcal{F}(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}) = \mathbf{0}.$$

Substituting the gradients and collecting the terms involving $\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}$ gives

$$\big(\boldsymbol{H}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f\big)\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} \;=\; \boldsymbol{b}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{y}_f,$$

or, writing

$$\boldsymbol{M}_\rho := \boldsymbol{H}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f,$$
$$\boldsymbol{M}_\rho\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} = \boldsymbol{b}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{y}_f.$$

The retrained optimum satisfies $\nabla\mathcal{R}(\boldsymbol{\theta}_r^\star) = 0$, so $\boldsymbol{H}_r\boldsymbol{\theta}_r^\star = \boldsymbol{b}_r$. Therefore

$$\boldsymbol{M}_\rho\boldsymbol{\theta}_r^\star = \Big(\boldsymbol{H}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f\Big)\boldsymbol{\theta}_r^\star = \boldsymbol{b}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f\boldsymbol{\theta}_r^\star.$$

Subtracting this identity from the equation for $\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}$ yields

$$\boldsymbol{M}_\rho(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star) = \Big(\boldsymbol{b}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{y}_f\Big) - \Big(\boldsymbol{b}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{X}_f\boldsymbol{\theta}_r^\star\Big) = \tfrac{\rho}{n}\boldsymbol{X}_f^\top(\boldsymbol{X}_f\boldsymbol{\theta}_r^\star - \boldsymbol{y}_f).$$

Using $\boldsymbol{e}_f^- = \boldsymbol{X}_f\boldsymbol{\theta}_r^\star - \boldsymbol{y}_f$, this becomes

$$\boldsymbol{M}_\rho(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star) = \tfrac{\rho}{n}\boldsymbol{X}_f^\top\boldsymbol{e}_f^-.$$

The assumption $\boldsymbol{M}_\rho \succ 0$ ensures that we can invert this system, so

$$\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star = \tfrac{\rho}{n}\boldsymbol{M}_\rho^{-1}\boldsymbol{X}_f^\top\boldsymbol{e}_f^-.$$

Applying Lemma 2 with $\boldsymbol{A} = \boldsymbol{H}_r$, $\boldsymbol{U} = \boldsymbol{X}_f^\top$, $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{I}_m$, and $\boldsymbol{V} = \boldsymbol{X}_f$,

$$\boldsymbol{M}_\rho^{-1}\boldsymbol{X}_f^\top = \boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top\Big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{X}_f\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top\Big)^{-1} = \boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top\Big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\Big)^{-1}.$$

Substituting this expression into the parameter gap gives

$$\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star \;=\; \tfrac{\rho}{n}\,\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top\big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\big)^{-1}\boldsymbol{e}_f^-.$$

Finally, left-multiplying by $\boldsymbol{X}_a$ and using $\boldsymbol{K}_{af} = \boldsymbol{X}_a\boldsymbol{H}_r^{-1}\boldsymbol{X}_f^\top$ gives

$$\boldsymbol{X}_a(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star) = \tfrac{\rho}{n}\boldsymbol{K}_{af}\Big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\Big)^{-1}\boldsymbol{e}_f^-,$$

which is (12). The equivalent full-data-residual form (11) follows by substituting Lemma 1:

$$\boldsymbol{X}_a(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star) = \tfrac{\rho}{n}\boldsymbol{K}_{af}\Big(\boldsymbol{I}_m - \tfrac{\rho}{n}\boldsymbol{K}_{ff}\Big)^{-1}\Big(\boldsymbol{I}_m + \tfrac{1}{n}\boldsymbol{K}_{ff}\Big)\boldsymbol{e}_f^+.$$

□

A.5 Proof of Corollary 1

The norm bound (13) follows from (12) and submultiplicativity of $\|\cdot\|_2$, using $\|(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{K}_{ff})^{-1}\|_2 = (1 - \frac{\rho}{n}\lambda_{\max}(\boldsymbol{K}_{ff}))^{-1}$ when $\boldsymbol{K}_{ff} \succeq 0$ and $\frac{\rho}{n}\lambda_{\max}(\boldsymbol{K}_{ff}) < 1$. The equivalent bound (14) follows analogously from (11) together with $\|\boldsymbol{I}_m + \frac{1}{n}\boldsymbol{K}_{ff}\|_2 = 1 + \frac{1}{n}\lambda_{\max}(\boldsymbol{K}_{ff})$ since $\boldsymbol{K}_{ff} \succeq 0$. The stability equivalence is obtained by conjugating $\boldsymbol{H}_r - \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{X}_f \succ 0$ by $\boldsymbol{H}_r^{-1/2}$ and noting that the nonzero eigenvalues of $\boldsymbol{H}_r^{-1/2}\boldsymbol{X}_f^\top \boldsymbol{X}_f \boldsymbol{H}_r^{-1/2}$ coincide with those of $\boldsymbol{K}_{ff}$. □

A.6 Proof of Proposition 1 (per-method)

All five non-trivial rows are obtained by the same recipe used for GradDiff in Appendix A.4: (i) write the method's regularized linear objective and its first-order stationarity condition, (ii) subtract the corresponding stationarity equation for $\boldsymbol{\theta}_r^\star$ to obtain a linear system in $\boldsymbol{\theta} - \boldsymbol{\theta}_r^\star$, and (iii) apply the push-through identity (Lemma 2) to express the result in the leverage-matrix form of Table 1. We give each row in turn.

GradDiff (recap). Objective $\mathcal{R}(\boldsymbol{\theta}) - \rho\mathcal{F}(\boldsymbol{\theta})$, stationarity $(\boldsymbol{H}_r - \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{X}_f)\boldsymbol{\theta} = \boldsymbol{b}_r - \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{y}_f$ where $\boldsymbol{b}_r := \frac{1}{n}\boldsymbol{X}_r^\top \boldsymbol{y}_r$ and $\boldsymbol{H}_r\boldsymbol{\theta}_r^\star = \boldsymbol{b}_r$. Subtracting and applying Lemma 2 with $\boldsymbol{A} = \boldsymbol{H}_r$, $\boldsymbol{U} = \boldsymbol{X}_f^\top$, $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{I}_m$, $\boldsymbol{V} = \boldsymbol{X}_f$ gives the GradDiff row (full derivation in Appendix A.4).

NPO and SimNPO. The (Sim)NPO objectives replace the unweighted forget term $-\rho\mathcal{F}$ by a per-sample preference-style loss

$$\text{NPO}: \ \mathcal{R}(\boldsymbol{\theta}) - \tfrac{\rho}{\beta} \cdot \tfrac{1}{n} \sum_{j \in \mathcal{D}_f} \log \sigma\big(\beta\,(\ell_j(\boldsymbol{\theta}) - \ell_j(\boldsymbol{\theta}_{\text{ref}}))\big),$$

$$\text{SimNPO}: \ \mathcal{R}(\boldsymbol{\theta}) - \tfrac{\rho}{\beta} \cdot \tfrac{1}{n} \sum_{j \in \mathcal{D}_f} \log \sigma\big(\beta\,\tilde{\ell}_j(\boldsymbol{\theta})\big),$$

where $\ell_j(\boldsymbol{\theta}) = \frac{1}{2}(y_j - \boldsymbol{x}_j^\top \boldsymbol{\theta})^2$ is the per-sample squared loss and $\tilde{\ell}_j$ is its length-normalized variant; $\boldsymbol{\theta}_{\text{ref}}$ is the NPO reference parameter. Differentiating,

$$\nabla_{\boldsymbol{\theta}}\big[-\tfrac{\rho}{\beta}\log\sigma(\beta\, t_j(\boldsymbol{\theta}))\big] = -\rho\,\sigma(-\beta t_j(\boldsymbol{\theta}))\,\nabla_{\boldsymbol{\theta}} t_j(\boldsymbol{\theta}),$$

which produces a per-sample weight $w_j(\boldsymbol{\theta}) = \sigma(z_j(\boldsymbol{\theta}))$ on the standard squared-loss gradient (the inverse-temperature factor β from differentiating the inner $\log\sigma(\beta\,\cdot)$ exactly cancels with the ρ/β prefactor of the objective, so the effective forget coefficient in the linear stationarity equation below is ρ and the diagonal weights are β-free, matching Section 4.2; under any alternative prefactor convention β would simply be absorbed into ρ). Freezing the weights at $\boldsymbol{\theta}_r^\star$ yields the linear stationarity condition

$$(\boldsymbol{H}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{W}^- \boldsymbol{X}_f)\boldsymbol{\theta} = \boldsymbol{b}_r - \tfrac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{W}^- \boldsymbol{y}_f,$$

with $\boldsymbol{W}^- = \boldsymbol{W}_{\text{NPO}}^-$ or $\boldsymbol{W}_{\text{SimNPO}}^-$. Subtracting the $\boldsymbol{\theta}_r^\star$ equation $\boldsymbol{H}_r\boldsymbol{\theta}_r^\star = \boldsymbol{b}_r$ gives $(\boldsymbol{H}_r - \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{W}^- \boldsymbol{X}_f)(\boldsymbol{\theta} - \boldsymbol{\theta}_r^\star) = \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{W}^- \boldsymbol{e}_f^-$. Applying Lemma 2 with $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{W}^-$ and left-multiplying by $\boldsymbol{X}_a$ yields the (Sim)NPO row.

Pure GA. The objective drops $\mathcal{R}$ and keeps only ridge regularization and forget ascent: $-\rho\mathcal{F}(\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$. The first-order condition is $(\lambda\boldsymbol{I}_d - \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{X}_f)\boldsymbol{\theta} = -\frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{y}_f$. Lemma 2 with $\boldsymbol{A} = \lambda\boldsymbol{I}_d$, $\boldsymbol{U} = \boldsymbol{X}_f^\top$, $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{I}_m$, $\boldsymbol{V} = \boldsymbol{X}_f$ gives $\boldsymbol{\theta}_\rho^{\text{GA}} = -\frac{\rho}{n\lambda}\boldsymbol{X}_f^\top (\boldsymbol{I}_m - \frac{\rho}{n\lambda}\boldsymbol{G}_{ff})^{-1}\boldsymbol{y}_f$. Subtracting $\boldsymbol{\theta}_r^\star$ and left-multiplying by $\boldsymbol{X}_a$ yields the Pure-GA row.

IMU. IMU rescales each forget term by a per-sample influence weight α_j, giving the objective $-\rho \cdot \frac{1}{n}\sum_{j \in \mathcal{D}_f} \alpha_j\, \ell_j(\boldsymbol{\theta}) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$ (without $\mathcal{R}$, matching the Pure-GA setting). The stationarity condition becomes the Pure-GA system with the influence diagonal $\boldsymbol{A}_{\text{IMU}}$ inserted: $(\lambda\boldsymbol{I}_d - \frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{A}_{\text{IMU}}\boldsymbol{X}_f)\boldsymbol{\theta} = -\frac{\rho}{n}\boldsymbol{X}_f^\top \boldsymbol{A}_{\text{IMU}}\boldsymbol{y}_f$. The same push-through application with $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{A}_{\text{IMU}}$ yields the IMU row. This is a fixed-weight local approximation; in the experiments the diagonal is instantiated once from $\boldsymbol{\theta}^\star$ using the final-layer proxy in Appendix B.

ProjGD. The fixed-projector method applies $\boldsymbol{P}_r$ to the forget gradient at every step, so the stationarity condition is $\nabla\mathcal{R}(\boldsymbol{\theta}) - \rho\boldsymbol{P}_r\nabla\mathcal{F}(\boldsymbol{\theta}) = \mathbf{0}$. In linear regression this gives $(\boldsymbol{H}_r - \frac{\rho}{n}\boldsymbol{P}_r\boldsymbol{X}_f^\top\boldsymbol{X}_f)\boldsymbol{\theta} = \boldsymbol{b}_r - \frac{\rho}{n}\boldsymbol{P}_r\boldsymbol{X}_f^\top\boldsymbol{y}_f$. Subtracting the $\boldsymbol{\theta}_r^\star$ equation and applying Lemma 2 with $\boldsymbol{A} = \boldsymbol{H}_r$, $\boldsymbol{U} = \boldsymbol{P}_r\boldsymbol{X}_f^\top$, $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{I}_m$, $\boldsymbol{V} = \boldsymbol{X}_f$ yields the ProjGD row, with $\boldsymbol{K}_{af}^{\mathrm{P}}$ and $\boldsymbol{K}_{ff}^{\mathrm{P}}$ in place of $\boldsymbol{K}_{af}$ and $\boldsymbol{K}_{ff}$.

Newton Step. Newton step solves $\boldsymbol{\theta}^\star + \boldsymbol{H}_r^{-1}\nabla\mathcal{F}(\boldsymbol{\theta}^\star) = \boldsymbol{\theta}_r^\star$ exactly in linear regression with squared loss, using the same push-through identity as Theorem 1 [39]. Hence the gap is zero. □

A.7 Proof of Proposition 2

Let $\boldsymbol{\Delta}_\rho = \boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star$. The first-order condition for $\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}$ is $\nabla\mathcal{R}(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}) - \rho\nabla\mathcal{F}(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}) = \mathbf{0}$. Using $\nabla\mathcal{R}(\boldsymbol{\theta}_r^\star) = \mathbf{0}$ and replacing the local gradients by their first-order GGN/Taylor model gives

$$\boldsymbol{H}_r^-\,\boldsymbol{\Delta}_\rho - \rho\,\boldsymbol{g}_f^- - \rho\,\boldsymbol{H}_{f,\mathrm{G}}^-\,\boldsymbol{\Delta}_\rho \;\approx\; \mathbf{0},$$

where $\boldsymbol{g}_f^- = \nabla\mathcal{F}(\boldsymbol{\theta}_r^\star) = \frac{1}{n}(\boldsymbol{J}_f^-)^\top\boldsymbol{\delta}_f^-$ and $\boldsymbol{H}_{f,\mathrm{G}}^- = \frac{1}{n}(\boldsymbol{J}_f^-)^\top\boldsymbol{B}_f^-\boldsymbol{J}_f^-$. Solving this local linear system gives

$$\widehat{\boldsymbol{\Delta}}_\rho = \rho(\boldsymbol{H}_r^- - \rho\boldsymbol{H}_{f,\mathrm{G}}^-)^{-1}\boldsymbol{g}_f^-.$$

Applying the same push-through identity with $\boldsymbol{A} = \boldsymbol{H}_r^-$, $\boldsymbol{U} = (\boldsymbol{J}_f^-)^\top$, $\boldsymbol{C} = \frac{\rho}{n}\boldsymbol{B}_f^-$, and $\boldsymbol{V} = \boldsymbol{J}_f^-$ yields

$$\widehat{\boldsymbol{\Delta}}_\rho = \tfrac{\rho}{n}\,(\boldsymbol{H}_r^-)^{-1}(\boldsymbol{J}_f^-)^\top\big(\boldsymbol{I}_{mc} - \tfrac{\rho}{n}\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}\big)^{-1}\boldsymbol{\delta}_f^-.$$

Left-multiplying by $\boldsymbol{J}_a^-$ gives the displayed output predictor in (22). □

A.8 A local nonlinear approximation-error bound

This appendix records a representative local bound for the approximation-error discussion in Section 4.3. Let $\boldsymbol{\Delta}_\rho = \boldsymbol{\theta}_\rho^{\mathrm{GDiff}} - \boldsymbol{\theta}_r^\star$ be any stationary GradDiff displacement in a neighborhood of $\boldsymbol{\theta}_r^\star$, and assume $\|(\boldsymbol{M}_\rho^-)^{-1}\|_2 \le \gamma$. Suppose that for all displacements $\boldsymbol{\Delta}$ in this neighborhood,

$$\|\nabla\mathcal{R}(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}) - \boldsymbol{H}_r^-\boldsymbol{\Delta}\|_2 \le \varepsilon_r\|\boldsymbol{\Delta}\|_2 + \tfrac{L_r}{2}\|\boldsymbol{\Delta}\|_2^2,$$
$$\|\nabla\mathcal{F}(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}) - \nabla\mathcal{F}(\boldsymbol{\theta}_r^\star) - \boldsymbol{H}_{f,\mathrm{G}}^-\boldsymbol{\Delta}\|_2 \le \varepsilon_f\|\boldsymbol{\Delta}\|_2 + \tfrac{L_f}{2}\|\boldsymbol{\Delta}\|_2^2,$$

and that the evaluation map satisfies

$$\|f(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}, A) - f(\boldsymbol{\theta}_r^\star, A) - \boldsymbol{J}_a^-\boldsymbol{\Delta}\|_2 \le \tfrac{L_a}{2}\|\boldsymbol{\Delta}\|_2^2.$$

Then the output-space predictor $\widehat{\Delta f}_A := \boldsymbol{J}_a^-\widehat{\boldsymbol{\Delta}}_\rho$ obeys

$$\big\|f(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}, A) - f(\boldsymbol{\theta}_r^\star, A) - \widehat{\Delta f}_A\big\|_2 \le \|\boldsymbol{J}_a^-\|_2\gamma\Big((\varepsilon_r + \rho\varepsilon_f)\|\boldsymbol{\Delta}_\rho\|_2 + \tfrac{L_r + \rho L_f}{2}\|\boldsymbol{\Delta}_\rho\|_2^2\Big) + \tfrac{L_a}{2}\|\boldsymbol{\Delta}_\rho\|_2^2. \tag{25}$$

For exact Hessians, $\varepsilon_r = \varepsilon_f = 0$ under ordinary Lipschitz-Hessian smoothness, by which we mean that $\nabla^2\mathcal{R}$ and $\nabla^2\mathcal{F}$ are L_r- and L_f-Lipschitz on the neighborhood, so a standard Taylor argument removes the linear remainder and leaves only the L_r, L_f second-order terms. With GGN curvatures, ε_r and ε_f record the omitted model-Hessian terms.

Let $\boldsymbol{g}_f^- = \nabla\mathcal{F}(\boldsymbol{\theta}_r^\star)$ and define the local-model remainders

$$\boldsymbol{r}_r(\boldsymbol{\Delta}) := \nabla\mathcal{R}(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}) - \boldsymbol{H}_r^-\boldsymbol{\Delta},$$
$$\boldsymbol{r}_f(\boldsymbol{\Delta}) := \nabla\mathcal{F}(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}) - \boldsymbol{g}_f^- - \boldsymbol{H}_{f,\mathrm{G}}^-\boldsymbol{\Delta}.$$

Since $\boldsymbol{\Delta}_\rho$ is stationary for GradDiff,

$$\begin{aligned}\mathbf{0} &= \nabla\mathcal{R}(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}_\rho) - \rho\nabla\mathcal{F}(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}_\rho)\\ &= \boldsymbol{M}_\rho^-\boldsymbol{\Delta}_\rho - \rho\boldsymbol{g}_f^- + \boldsymbol{r}_r(\boldsymbol{\Delta}_\rho) - \rho\boldsymbol{r}_f(\boldsymbol{\Delta}_\rho).\end{aligned}$$

The predictor satisfies $\boldsymbol{M}_\rho^-\widehat{\boldsymbol{\Delta}}_\rho = \rho\boldsymbol{g}_f^-$. Subtracting the two equations gives

$$\boldsymbol{\Delta}_\rho - \widehat{\boldsymbol{\Delta}}_\rho = (\boldsymbol{M}_\rho^-)^{-1}\big(-\boldsymbol{r}_r(\boldsymbol{\Delta}_\rho) + \rho\boldsymbol{r}_f(\boldsymbol{\Delta}_\rho)\big).$$

Using the assumed remainder bounds and $\|(\boldsymbol{M}_\rho^-)^{-1}\|_2 \le \gamma$,

$$\|\boldsymbol{\Delta}_\rho - \widehat{\boldsymbol{\Delta}}_\rho\|_2 \le \gamma\Big((\varepsilon_r + \rho\varepsilon_f)\|\boldsymbol{\Delta}_\rho\|_2 + \tfrac{L_r+\rho L_f}{2}\|\boldsymbol{\Delta}_\rho\|_2^2\Big).$$

Finally,

$$\begin{aligned}&\big\|f(\boldsymbol{\theta}_\rho^{\mathrm{GDiff}}, A) - f(\boldsymbol{\theta}_r^\star, A) - \boldsymbol{J}_a^-\widehat{\boldsymbol{\Delta}}_\rho\big\|_2 \\ &\quad\le \|\boldsymbol{J}_a^-\|_2\|\boldsymbol{\Delta}_\rho - \widehat{\boldsymbol{\Delta}}_\rho\|_2 + \big\|f(\boldsymbol{\theta}_r^\star + \boldsymbol{\Delta}_\rho, A) - f(\boldsymbol{\theta}_r^\star, A) - \boldsymbol{J}_a^-\boldsymbol{\Delta}_\rho\big\|_2,\end{aligned}$$

and the evaluation-map remainder bound gives (25). □

A.9 Local nonlinear forms for the other methods

The local nonlinear rows are obtained from Table 1 by replacing linear features with Jacobians and inserting the output-Hessian/GGN factor $\boldsymbol{B}_f^-$. For NPO and SimNPO, weights are frozen at $\boldsymbol{\theta}_r^\star$ and their derivatives are omitted. Scalar per-sample weights are lifted to output blocks as $\widetilde{\boldsymbol{W}} = \boldsymbol{W} \otimes \boldsymbol{I}_c$ and $\widetilde{\boldsymbol{A}} = \boldsymbol{A} \otimes \boldsymbol{I}_c$. For fixed-projector ProjGD and the Euclidean methods, define

$$\boldsymbol{K}_{af}^{\mathrm{NL,P}} := \boldsymbol{J}_a^-(\boldsymbol{H}_r^-)^{-1}\boldsymbol{P}_r(\boldsymbol{J}_f^-)^\top, \qquad \boldsymbol{K}_{ff}^{\mathrm{NL,P}} := \boldsymbol{J}_f^-(\boldsymbol{H}_r^-)^{-1}\boldsymbol{P}_r(\boldsymbol{J}_f^-)^\top,$$
$$\boldsymbol{G}_{af}^{\mathrm{NL}} := \boldsymbol{J}_a^-(\boldsymbol{J}_f^-)^\top, \qquad \boldsymbol{G}_{ff}^{\mathrm{NL}} := \boldsymbol{J}_f^-(\boldsymbol{J}_f^-)^\top.$$

For the Euclidean rows, let

$$\boldsymbol{\eta}_f^- := \boldsymbol{\delta}_f^- - \boldsymbol{B}_f^-\boldsymbol{J}_f^-\boldsymbol{\theta}_r^\star.$$

The shifted residual $\boldsymbol{\eta}_f^-$ appears in the Pure GA and IMU rows because their local linearization replaces the regression target $\boldsymbol{y}_f$ of Table 1 by its linearized analog $\boldsymbol{\delta}_f^- - \boldsymbol{B}_f^-\boldsymbol{J}_f^-\boldsymbol{\theta}_r^\star$; this is the per-sample object that plays the role of $-\boldsymbol{y}_f$ when the loss is taken to second order in the model output rather than in the parameter directly. Whenever the displayed inverse exists, the leading-order local output gaps are summarized in Table 4. All quantities are evaluated at $\boldsymbol{\theta}_r^\star$. The Pure GA and IMU rows are written in Euclidean-Gram form; because they do not use the retrained-Hessian K geometry, the K-channel pointwise diagnostic in Appendix D is not the predictor for those rows.

Method	Local output gap $f(\boldsymbol{\theta}_\rho^{\text{method}}, A) - f(\boldsymbol{\theta}_r^\star, A)$	Geometry
GradDiff	$\frac{\rho}{n}\boldsymbol{K}_{af}^{\mathrm{NL}}\big(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}\big)^{-1}\boldsymbol{\delta}_f^-$	retrained Hessian
NPO	$\frac{\rho}{n}\boldsymbol{K}_{af}^{\mathrm{NL}}\big(\boldsymbol{I} - \frac{\rho}{n}\widetilde{\boldsymbol{W}}_{\mathrm{NPO}}^-\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}\big)^{-1}\widetilde{\boldsymbol{W}}_{\mathrm{NPO}}^-\boldsymbol{\delta}_f^-$	retrained Hessian
SimNPO	$\frac{\rho}{n}\boldsymbol{K}_{af}^{\mathrm{NL}}\big(\boldsymbol{I} - \frac{\rho}{n}\widetilde{\boldsymbol{W}}_{\mathrm{SimNPO}}^-\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL}}\big)^{-1}\widetilde{\boldsymbol{W}}_{\mathrm{SimNPO}}^-\boldsymbol{\delta}_f^-$	retrained Hessian
Pure GA	$-\boldsymbol{J}_a^-\boldsymbol{\theta}_r^\star + \frac{\rho}{n\lambda}\boldsymbol{G}_{af}^{\mathrm{NL}}\big(\boldsymbol{I} - \frac{\rho}{n\lambda}\boldsymbol{B}_f^-\boldsymbol{G}_{ff}^{\mathrm{NL}}\big)^{-1}\boldsymbol{\eta}_f^-$	Euclidean Gram
IMU	$-\boldsymbol{J}_a^-\boldsymbol{\theta}_r^\star + \frac{\rho}{n\lambda}\boldsymbol{G}_{af}^{\mathrm{NL}}\big(\boldsymbol{I} - \frac{\rho}{n\lambda}\widetilde{\boldsymbol{A}}_{\mathrm{IMU}}\boldsymbol{B}_f^-\boldsymbol{G}_{ff}^{\mathrm{NL}}\big)^{-1}\widetilde{\boldsymbol{A}}_{\mathrm{IMU}}\boldsymbol{\eta}_f^-$	Euclidean Gram
ProjGD (fixed)	$\frac{\rho}{n}\boldsymbol{K}_{af}^{\mathrm{NL,P}}\big(\boldsymbol{I} - \frac{\rho}{n}\boldsymbol{B}_f^-\boldsymbol{K}_{ff}^{\mathrm{NL,P}}\big)^{-1}\boldsymbol{\delta}_f^-$	retrained Hessian (projected)
Newton Step	$\mathbf{0}$ in the local quadratic retraining model	retrained Hessian (Newton)

Table 4: Local nonlinear counterparts of Table 1; notation as defined in Section 4.3. NPO/SimNPO use weights frozen at $\boldsymbol{\theta}_r^\star$, omitting their derivatives. The Newton Step row omits higher-order and GGN-mismatch residuals.

B Experimental Setup Details

This appendix gives the full reproducibility detail for the experiments of Section 5: data, model, hardware, software, hyperparameters, and per-experiment protocols.

Hardware and software. All MNIST/MLP experiments run in float64 on a single NVIDIA H100 NVL 96GB GPU within a Linux + CUDA $\ge$ 11.8 environment (PyTorch 2). The figures below were measured by re-running one seed of every experiment in its final configuration on an otherwise idle GPU with our implementation and scaling by the number of seeds. A single iterative MNIST unlearning run (full-batch Adam, 500 steps) takes a few seconds, and the Newton step is a single closed-form update. One (forget set, seed) cell of Table 2, comprising the $\boldsymbol{\theta}_r^\star$ solve,

the leverage geometry, and all five unlearners, takes about one minute, so the regime comparison (330 cells, 100×3 random and 10×3 class) totals $\sim$5 GPU-hours. The matched-CE experiments of Figure 1 and Table 5 (10 seeds, ρ bisection for six methods) take $\sim$0.2 GPU-hours. The per-example validations of Appendix D take about 4 (MNIST) and 2 (CIFAR-10) minutes per seed. The fixed-point validation of Table 3 takes $\sim$2.9 GPU-hours on MNIST and $\sim$1.9 on CIFAR-10 for 10 seeds, dominated by the polishing of $\boldsymbol{\theta}_r^\star$ and the dense fp64 curvature solves. The TOFU experiments train no base model. The oracle-proxy analysis of Appendix C.1 consumes the saved per-checkpoint per-example evaluations of one trajectory of the public `open-unlearning` recipe. The seed study of Appendix C.2 uses fifteen ten-epoch unlearning trajectories of that recipe with per-epoch evaluation (6–49 minutes each, 6.9 GPU-hours in total), a one-time pass on Llama-3.2-1B-Instruct that caches the LoRA-space diagonal GGN over the 4,000 TOFU training sequences ($\sim$7 minutes), and K_{ff}/K_{rf} scoring that takes under two minutes per split. We do not separately enumerate compute used by preliminary or discarded experiments. We plan to open-source our code at `https://github.com/xingjianz/loss-is-not-behavior`; the code is not yet publicly available. *Asset versions and licenses.* MNIST [20] (CC BY-SA 3.0), CIFAR-10 [19] together with the pretrained ResNet-18 released with the NeurIPS 2023 Machine Unlearning Challenge starting kit (`https://github.com/unlearning-challenge/starting-kit`), the TOFU dataset [27] (MIT), the `open-unlearning` library [6] (MIT), Llama-3.2-1B-Instruct [14] (Llama 3.2 Community License), PyTorch (BSD-3-Clause), and Hugging Face Transformers/PEFT (Apache-2.0); all are used within their licensed terms. Random seeds are explicit per run; all stochastic data subsampling, weight initialization, and SGD/Adam mini-batching is reproducible from the seed.

Data: MNIST. We use the standard MNIST dataset of 28×28 grayscale digits (60 000 training, 10 000 test), accessed locally as a NumPy archive. Per seed, we draw a random training subsample of $n_{\mathrm{tr}} = 4000$ images and a random test subsample of $n_{\mathrm{te}} = 2000$ images, normalize pixels to $[0, 1]$, and standardize features per-coordinate using the training-subset mean and standard deviation (an additive 10^{-6} floor on the std). Labels are one-hot encoded for squared/CE losses.

Model and base training. The MLP has architecture $784 \to 24 \to 10$ with $\tanh$ hidden activation, a softmax output head, and biases in both layers ($d = 19{,}090$). We initialize weights from a zero-mean Gaussian with variance equal to inverse fan-in and biases to zero, using the run seed. Base models are trained for 40 epochs with SGD (learning rate 0.1, batch size 256, momentum 0.9), without L-BFGS polishing. The matched-CE motivation experiments use $\lambda = 10^{-3}$; the regime comparison and fixed-point validation use $\lambda = 10^{-2}$. The initial warm retrain runs for 20 epochs from $\boldsymbol{\theta}^\star$ on the retain set, with learning rate 0.05, batch size 64, momentum 0.9, and mini-batch seed $\mathrm{seed} + 1$. The regime comparison replaces this reference and the fixed-point validation polishes it as specified below.

Unlearners and optimization. The motivation and regime experiments start unlearning from $\boldsymbol{\theta}^\star$; the fixed-point experiments start from the polished $\boldsymbol{\theta}_r^\star$. The iterative methods optimize full-batch objectives. GradDiff minimizes retain CE minus ρ times forget CE plus ℓ_2 regularization. NPO uses reference log-probabilities from $\boldsymbol{\theta}^\star$ with $\beta = 0.1$; SimNPO uses the reference-free loss with $\beta = 2.5$ and margin $\delta = 0$. The β values follow the open-unlearning library defaults. Pure GA drops the retain term and ascends forget CE with only the ℓ_2 anchor. IMU uses the same Pure-GA objective but multiplies each forget CE by a fixed weight equal to the norm of the final-layer CE gradient at $\boldsymbol{\theta}^\star$, normalized to mean one; this is the experimental proxy for the ideal influence norm in Table 1. ProjGD applies the GradDiff direction after projecting out the span of 64 retain-anchor CE gradients in the Euclidean metric, with the surrogate subspace recomputed along the trajectory. These choices differ from the closed-form theory rows, which solve stationary fixed-weight or frozen-local surrogate objectives.

The matched-CE protocols use deterministic full-batch SGD-style runners during ρ bisection: 120 steps per evaluation, lr 0.05 for GradDiff/NPO/SimNPO/ProjGD and lr 0.005 for Pure GA/IMU. The four iterative methods in Section 5.1 instead use full-batch Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, lr 5×10^{-3}) for 500 steps with $\lambda = 10^{-2}$; no convergence-based stopping is used. Per-method ρ-bracket centers for matched-CE tuning were calibrated once at the start of the project as $\rho_0 \in \{$GradDiff: 5, NPO: 5, SimNPO: 30, ProjGD: 5, Pure GA: 2, IMU: 0.5$\}$. The Newton step is a single closed-form second-order update from $\boldsymbol{\theta}^\star$ using the local GGN curvature; it has no ρ knob.

Method	Forget CE	Top-1 ↓	Top-2 ↓	KL($p_{r^\star}\|p$) ↓	Margin	MIA → 0.5	Test acc ↑
GradDiff	0.172±0.037	0.961±0.009	0.988±0.009	0.057±0.014	0.930±0.007	0.497±0.027	0.904±0.006
NPO	0.174±0.036	0.959±0.008	0.988±0.008	0.054±0.012	0.928±0.008	0.499±0.026	0.904±0.006
SimNPO	0.176±0.041	0.904±0.024	0.995±0.006	0.053±0.014	0.872±0.011	0.490±0.029	0.904±0.007
PureGA	0.171±0.036	0.958±0.007	0.988±0.009	0.075±0.019	0.924±0.008	0.495±0.028	0.902±0.007
IMU	0.178±0.039	0.972±0.006	0.986±0.005	0.106±0.028	0.948±0.008	0.507±0.027	0.901±0.007
ProjGD	0.171±0.040	0.961±0.009	0.986±0.010	0.054±0.013	0.933±0.007	0.496±0.025	0.905±0.007
Newton Step	0.409±0.057	0.880±0.019	0.952±0.013	0.077±0.022	0.869±0.015	0.476±0.024	0.905±0.007
Retraining	0.173±0.034	0.940±0.012	0.985±0.009	0	0.901±0.012	0.489±0.026	0.907±0.007

Table 5: Random-forget aggregate results on MNIST/MLP ($|\mathcal{D}_f| = 200$, 10 seeds). The tunable gradient methods are matched to the retrained forget cross-entropy; Newton Step has no ρ knob and is reported as its single second-order step. Setup in Appendix B; class-forget version in Figure 1.

The MNIST per-coordinate logit RMS is $\sqrt{(1/|A|)\sum_{i\in A}\|f(\boldsymbol{\theta},x_i)-f(\boldsymbol{\theta}_r^\star,x_i)\|_2^2/c}$ with $c=10$; TOFU instead uses per-example sequence-NLL deviation from the retain95 oracle.

Class-forget protocol (Figure 1). The forget set is the entire instance of a single class in the seed's training subsample (rotated across seeds: seed $i \to$ class $i \bmod 10$, so $|\mathcal{D}_f|$ varies between ~ 350 and ~ 490 depending on which class lands in that seed's 4000-sample subset). For each (seed, method) pair we tune ρ by geometric bisection on the bracket $[\rho_0/16,\ 16\rho_0]$ until the achieved forget CE on $\mathcal{D}_f$ falls within $\pm 10\%$ of $\mathrm{CE}(\boldsymbol{\theta}_r^\star, \mathcal{D}_f)$, capped at 12 bisection steps. Because Newton step has no ρ knob, we report it without the matching constraint and flag it with hatched bars in Figure 1. Figure 1 displays four observables: forget CE, weight distance $\|\theta-\boldsymbol{\theta}_r^\star\|_2$, $\mathrm{KL}(p_{\boldsymbol{\theta}_r^\star}\|p_\theta)$ on $\mathcal{D}_f$, and top-1 forget accuracy. The full observable set — additionally including top-2 forget accuracy, confidence margin, held-out test accuracy, and a balanced-accuracy MIA based on a logistic attack on per-example loss (anchor batch from the retain set, equal forget/retain balance) — appears in the random-forget aggregate Table 5. Means and standard deviations are over 10 seeds. The random-forget protocol uses the same recipe but with $\mathcal{D}_f$ a uniformly random subsample of size 200 from the seed's training subsample.

Stability-fraction protocol (Section 5.1). For each forget set, we form the reference $\boldsymbol{\theta}_r^\star$ by 500 full-batch Adam steps on the full-data-normalized $\mathcal{R}$ from $\boldsymbol{\theta}^\star$, at learning rate 5×10^{-3} and $\lambda = 10^{-2}$. This fixed-step reference is not polished to a prescribed gradient tolerance. Curvature is estimated at this reference as

$$\widetilde{\boldsymbol{H}}_r = \frac{1}{n-m}\sum_{i\in\mathcal{D}_r} \boldsymbol{J}_i^\top \boldsymbol{B}_i \boldsymbol{J}_i + \lambda \boldsymbol{I}.$$

This retain-normalized GGN is a surrogate for the full-n curvature in the theory. The $\boldsymbol{K}$ quantities in Table 2 denote estimates $\boldsymbol{J}_a \widetilde{\boldsymbol{H}}_r^{-1} \boldsymbol{J}_f^\top$: the self-leverage block uses all m forget examples, and the cross-leverage block uses 200 retain anchors and all m forget examples. We set $\rho = 0.3\, n/\lambda_{\max}(\boldsymbol{K}_{ff})$ for GradDiff, NPO, SimNPO, and ProjGD as a heuristic motivated by the linear stability rule. Each takes 500 full-batch Adam steps from $\boldsymbol{\theta}^\star$; Newton Step takes one update. The class regime uses $m = 350$–490 (mean $\approx$400); the random regime uses $m = 200$. Aggregates are over $10\times3 = 30$ class runs and $100\times3 = 300$ random runs. Output deviations are per-coordinate logit RMS on $\mathcal{D}_f$ and $\mathcal{D}_r$. Mean ρ is 1.14×10^{-3} (random) and 6.69×10^{-4} (class), varying by an order of magnitude across random forget sets. Across all five methods, mean forget CE ranges from 0.329 to 0.359 in the random regime. Class-regime CE is 4.368–4.404 for the four iterative methods and 2.972 for Newton Step.

GGN stability rule. For unprojected GradDiff and frozen-weight (Sim)NPO GGN models formed with the same full-n normalization as the retain objective, the softmax-curvature bound $\|\boldsymbol{B}_j^-\|_2 \le \frac{1}{2}$ bounds the effective kernel eigenvalues by $(w_{\max}/2)\lambda_{\max}(\boldsymbol{K}_{ff}^{\mathrm{NL}})$, where $w_{\max}$ is the largest sample weight. It equals 1 for GradDiff and is at most 1 under the table's σ convention or 2 under the deployed 2σ convention, equivalent after rescaling ρ. Thus $s < 1$ suffices for positive definiteness of these frozen GGN models under the unweighted rule; locality up to $s = 2$ remains an empirical observation. Table 2 uses the differently normalized surrogate above, so its s budget is not a direct certificate for the implemented objectives. Saturated SimNPO weights barely move the iterate under this rule, motivating the weighted kernel in Section 5.2.

CIFAR-10 over frozen features (Table 3 and Table 10). For CIFAR-10 [19] we use the ResNet-18 pretrained on CIFAR-10 that was released with the NeurIPS 2023 Machine Unlearning Challenge starting kit (test accuracy 88.6%) and extract its frozen 512-dimensional penultimate features. The backbone is held fixed and is not the target of unlearning. The unlearning analysis runs on a 2-layer MLP head of architecture $512 \to 24 \to 10$ ($\tanh$ hidden activation, softmax output, $d = 12{,}562$ parameters), so the feature vector plays the role of the input $\boldsymbol{x}$ and the MNIST pipeline applies unchanged. Per seed we draw 4000 training and 2000 test examples, standardize the features per coordinate, train the head to obtain $\boldsymbol{\theta}^\star$, and form $\boldsymbol{\theta}_r^\star$ exactly as for MNIST, with $|\mathcal{D}_f| = 200$ in the random regime and one full class ($|\mathcal{D}_f| \approx 400$) in the class regime.

Fixed-point validation protocol (Section 5.2). Per dataset, regime, and seed, the reference $\boldsymbol{\theta}_r^\star$ is the 20-epoch warm-start retrain above, polished to a stationary point of $\mathcal{R}$ under the normalization of (2) with $\lambda = 10^{-2}$. The SGD retrain alone is not stationary, so we polish it with multi-restart L-BFGS, which reduces $\|\nabla\mathcal{R}\|_2$ below 10^{-7} and brings it to the stationary point in the basin of $\boldsymbol{\theta}^\star$. The retain curvature is the exact dense fp64 GGN $\boldsymbol{H}_r = \frac{1}{n}\sum_{i\in\mathcal{D}_r} \boldsymbol{J}_i^\top \boldsymbol{B}_i \boldsymbol{J}_i + \lambda \boldsymbol{I}$ at $\boldsymbol{\theta}_r^\star$. The stability rule sets $\rho = s\, n/\lambda_{\max}(\boldsymbol{K}_{ff})$ with $s = 0.1$. For SimNPO the method-weighted kernel $\boldsymbol{W}^{1/2}\boldsymbol{K}_{ff}\boldsymbol{W}^{1/2}$ replaces $\boldsymbol{K}_{ff}$, because under the uniform rule its saturated class-regime weights ($\bar{w} \approx 10^{-4}$) leave the endpoint at $\boldsymbol{\theta}_r^\star$. For GradDiff, NPO, SimNPO, Pure GA, and IMU, stationary endpoints are located by damped Newton continuation from $\boldsymbol{\theta}_r^\star$; NPO and SimNPO optimize their deployed objectives with parameter-dependent weights. ProjGD is run as projected-gradient dynamics from $\boldsymbol{\theta}_r^\star$ (1000 steps, learning rate 0.005, 64 retain anchors). The branch through $\boldsymbol{\theta}_r^\star$ is not the only stationary point near $\boldsymbol{\theta}^\star$. For MNIST seed 0, a GradDiff run started at $\boldsymbol{\theta}^\star$ and converged with L-BFGS at the same ρ ends 1.8 (random) and 2.5 (class) from the continuation endpoint in parameter norm, at a stationary point whose forget-logit RMS from $\boldsymbol{\theta}_r^\star$ (0.26 and 0.39) the local predictor does not describe ($P_f^K \approx 0$), while the same run started at $\boldsymbol{\theta}_r^\star$ returns to the continuation endpoint within 5×10^{-5}. The formulas describe the branch through the specified reference, and a deployed run inherits them only insofar as it converges to that branch. GradDiff, NPO, and SimNPO predictions use their rows of Table 4, with NPO/SimNPO weights frozen at $\boldsymbol{\theta}_r^\star$ and their derivatives omitted. ProjGD is compared with the unprojected predictor (22); Pure GA/IMU are evaluated with simplified K- and Euclidean-Gram controls that omit the shifted-residual and reference-offset terms of their theoretical rows. Predictions are compared with the actual logit gap on $\mathcal{D}_f$ and on 256 random retain anchors by per-logit Pearson correlation and relative ℓ_2 error. The Euclidean control replaces $\boldsymbol{H}_r^{-1}$ by $\lambda^{-1}\boldsymbol{I}$ in the prediction; for NPO/SimNPO it uses uniform forget weights, so these comparisons change both geometry and weighting. The shuffled-anchor control permutes the anchor rows of the prediction (it gives $|P| \le 0.20$ on every row). The output KL quoted in Section 5.2 is the mean per-example $\mathrm{KL}(p_{\boldsymbol{\theta}_r^\star} \| p_{\boldsymbol{\theta}})$ at the endpoint. Aggregates are over 10 seeds per dataset and regime.

TOFU/Llama setup (Appendix C). All TOFU runs use the `open-unlearning` full-finetune recipe [6] on TOFU [27] with Llama-3.2-1B-Instruct [14] as the base model (learning rate 10^{-5}, 10 epochs, per-epoch checkpoints, library-default β for NPO and SimNPO). The released retain95 (retain99) model is the oracle $\boldsymbol{\theta}_r^\star$ for forget05 (forget01). Two analyses are reported.

The *oracle-NLL forget-channel proxy* (Appendix C.1) uses one GradDiff forget05 trajectory and the score $s_{\mathrm{NLL}}(i) = \mathrm{avg_loss}_{\boldsymbol{\theta}_r^\star}(x_i)$, the per-token NLL of the gold answer under the retain95 oracle. This score is not an explicit Hessian leverage estimate; it is a scalable oracle-evaluable surrogate for the forget residual term that appears together with K_{ff} in Eq. (22). We sort all 200 forget05 examples by s_{NLL}, split them into quartiles, and compute the per-bin mean $|\Delta_i|$ at every saved checkpoint, where Δ_i is the per-example sequence-NLL deviation of the unlearned model from the oracle.

The *seed study with $\boldsymbol{\theta}^\star$-only scores* (Appendix C.2) covers $\{\text{forget01, forget05}\} \times \{\text{GradDiff, NPO}\}$ and forget05 SimNPO with three optimizer seeds each. The three seeds are the trainer seeds 0, 1, 2 of the recipe with no other change, and the seed-0 GradDiff forget05 trajectory is the one used in Appendix C.1. The scores are computed once at $\boldsymbol{\theta}^\star$ and never see $\boldsymbol{\theta}_r^\star$ or the trajectories. We restrict the parameter space to a LoRA adapter (r=8, target modules `all-linear`, 5.6×10^6 trainable parameters) and approximate the curvature by $\widehat{\boldsymbol{H}} = \mathrm{diag}\big(\frac{1}{N}\widehat{\mathrm{GGN}}\big) + \lambda\boldsymbol{I}$ with $\lambda = 0.01$, where the diagonal uses one Monte-Carlo label sample per training sequence on $\mathcal{D}_f \cup \mathcal{D}_r$, followed by a Woodbury low-rank correction built from S=4 MC pseudo-gradients on $\mathcal{D}_f$. With $\boldsymbol{g}_i$ the LoRA-chart gradient of the answer NLL of example i at $\boldsymbol{\theta}^\star$, the forget score is $\boldsymbol{g}_j^\top \widehat{\boldsymbol{H}}^{-1} \boldsymbol{g}_j$ on the 200 (forget05)

GradDiff step	Forget $\lvert\Delta_i\rvert$ by oracle-NLL proxy quartile				Q4/Q1	Spearman$(s_{\mathrm{NLL}}, \lvert\Delta\rvert)$
	Q1 (low)	Q2	Q3	Q4 (high)		
0	1.219	2.037	2.740	3.518	2.89×	+0.993
12	1.006	1.757	2.386	3.129	3.11×	+0.909
31	0.903	1.099	1.154	1.202	1.33×	+0.251
50	1.228	1.589	1.578	1.309	1.07×	+0.104
60	1.250	1.622	1.613	1.316	1.05×	+0.091

Table 6: Per-example forget deviation $|\Delta_i|$ grouped by oracle-NLL proxy quartile along the GradDiff trajectory on TOFU/Llama-3.2-1B-Instruct (forget05, $|\mathcal{D}_f| = 200$, 50 examples per quartile). Q1/Q4 contain the lowest/highest proxy scores. Setup in Appendix B. The rows are successive checkpoints of one trajectory, not independent runs. The step-0 Spearman is close to a tautology, because at step 0 $\Delta_i = \mathrm{NLL}_{\boldsymbol{\theta}_r^\star}(x_i) - \mathrm{NLL}_{\boldsymbol{\theta}^\star}(x_i)$ while the proxy is $\mathrm{NLL}_{\boldsymbol{\theta}_r^\star}(x_i)$ and $\mathrm{NLL}_{\boldsymbol{\theta}^\star}(x_i)$ is small (mean 0.13 vs. 2.5). Independent seeds and $\boldsymbol{\theta}^\star$-only scores are in Tables 7 and 8.

or 40 (forget01) forget examples, and the retain score is $\sum_{j\in\mathcal{D}_f} |\boldsymbol{g}_i^\top \widehat{\boldsymbol{H}}^{-1} \boldsymbol{g}_j|$ on the 400 probes of the standard TOFU `retain_perturbed` split. Each score is compared with the per-example $|\mathrm{NLL}_u - \mathrm{NLL}_{\mathrm{ret}}|$ at every per-epoch checkpoint except checkpoint 0, by Spearman correlation, AUC, and the Q4/Q1 ratio of mean gaps. Checkpoints are aggregated by the median within a trajectory and then averaged across seeds, with hierarchical-bootstrap confidence intervals (B=1,000, outer level trajectories, inner level examples) and within-checkpoint permutation nulls (1,000 permutations). Cheap baselines are the NLL at $\boldsymbol{\theta}^\star$ (forget) and the gradient norm $\|\boldsymbol{g}_i\|_2$ or the NLL at $\boldsymbol{\theta}^\star$ (retain). Table 8 re-analyzes the same trajectories against the NLL movement from $\boldsymbol{\theta}^\star$ with cosine, Pearson, and a fitted scale, and repeats the scoring on a half-frozen LoRA chart ($\boldsymbol{A}$ frozen, exact GGN over the $\boldsymbol{B}$ coordinates).

C LLM-scale diagnostics on TOFU/Llama

We ask whether the K-channel quantities of Section 5 remain informative in the LLM regime, where explicit GGN leverage on the full parameter space is intractable. All runs use the `open-unlearning` recipe on TOFU with Llama-3.2-1B-Instruct [6, 14, 27] and the released retain models as $\boldsymbol{\theta}_r^\star$. Appendix C.1 reports an oracle-NLL proxy for the forget-side residual channel along one GradDiff trajectory. Appendix C.2 reports $\boldsymbol{\theta}^\star$-only LoRA-space scores for the forget and retain channels against five method/split combinations with three optimizer seeds each. Both protocols are described in Appendix B.

C.1 Oracle-NLL forget-channel proxy

Equation (22) couples the forget self-leverage with the residual term $\boldsymbol{\delta}_f^-$; for next-token CE, the per-token NLL of the gold answer under $\boldsymbol{\theta}_r^\star$ is not a direct estimate of K_{ff}, but it is an oracle-evaluable proxy for this residual-weighted forget channel. We compute $s_{\mathrm{NLL}}(i) = \mathrm{avg_loss}_{\boldsymbol{\theta}_r^\star}(x_i)$ for each forget example, sort examples by it, and split them into four equal-sized quartile groups. Because this score uses only forget-example oracle NLL, it probes the forget-side channel rather than retain cross-leverage.

Observation. The qualitative prediction holds for this proxy: the high-score quartile (Q4) separates from the low-score one (Q1) throughout the trajectory, with a Q4/Q1 ratio of 1.3–3.1× at intermediate checkpoints and a per-example Spearman$(s_{\mathrm{NLL}}, |\Delta|)$ that starts at 0.99 at step 0 and decays to ∼0.1 as the forget channel saturates (Table 6). The diagnostic remains informative at LLM scale despite the absence of an explicit Hessian, optimization noise, and the move from squared loss to next-token CE. We note that this proxy still requires the retrained reference $\boldsymbol{\theta}_r^\star$; a fully deployable LLM-scale variant would build on the precomputable $\boldsymbol{e}_f^+$ form of Theorem 1 (computed only at $\boldsymbol{\theta}^\star$), which we discuss as future work in Section 6.

C.2 $\theta^\star$-only K_{ff} and K_{rf} scores over independent seeds

The Section 5.1 regime comparison hinges on K_{ff} and K_{rf} being computable. To check that this is not specific to small networks, and to remove the dependence on a single trajectory and on the oracle, we compute both channels once at $\boldsymbol{\theta}^\star$ in a LoRA chart (r=8, 5.6×10^6 coordinates) with a diagonal-plus-Woodbury GGN, and compare them with the realized per-example gaps $|\mathrm{NLL}_u - \mathrm{NLL}_{\mathrm{ret}}|$ of GradDiff, NPO, and SimNPO over three optimizer seeds on forget01 and forget05 (full protocol in Appendix B). The forget score is the residual-weighted self-leverage $\boldsymbol{g}_j^\top \widehat{\boldsymbol{H}}^{-1} \boldsymbol{g}_j$ and the retain score is the cross-leverage row sum $\sum_{j\in\mathcal{D}_f} |\boldsymbol{g}_i^\top \widehat{\boldsymbol{H}}^{-1} \boldsymbol{g}_j|$, both from gradients at $\boldsymbol{\theta}^\star$ alone, so this is the precomputable form anticipated in Section 6.

Forget channel						
Split	Method	Spearman$_f$ [95% CI] / AUC$_f$	Q4/Q1$_f$ [CI]	Δ_f vs. loss	null$_f$ (p)	token KL$_f$
forget01	GradDiff	0.14 [4.7e-3, 0.24] / 0.61	1.48 [1.05, 1.75]	0.04	[-0.07, 0.07] (0.001)	3.49±0.09
forget05	GradDiff	0.29 [0.25, 0.33] / 0.68	1.83 [1.61, 1.97]	-9.0e-3	[-0.03, 0.03] (0.001)	5.32±0.12
forget01	NPO	2.4e-3 [-0.07, 0.13] / 0.52	1.12 [0.95, 1.31]	0.03	[-0.07, 0.07] (0.988)	1.69±0.08
forget05	NPO	0.21 [0.16, 0.23] / 0.58	1.39 [1.29, 1.48]	0.06	[-0.03, 0.03] (0.001)	1.37±0.10
forget05	SimNPO	0.23 [0.20, 0.26] / 0.61	1.28 [1.25, 1.34]	0.04	[-0.03, 0.03] (0.001)	2.29±0.03
Retain channel						
Split	Method	Spearman$_r$ [95% CI] / AUC$_r$	Q4/Q1$_r$ [CI]	Δ_r vs. baseline	null$_r$ (p)	token KL$_r$
forget01	GradDiff	0.41 [0.40, 0.44] / 0.72	2.92 [2.23, 4.28]	-0.08	[-0.02, 0.02] (0.001)	0.694±0.042
forget05	GradDiff	0.28 [0.26, 0.31] / 0.63	1.60 [1.55, 1.76]	-0.07 (loss)	[-0.02, 0.02] (0.001)	1.183±0.049
forget01	NPO	0.49 [0.46, 0.53] / 0.73	3.69 [3.28, 4.22]	-0.08	[-0.02, 0.02] (0.001)	0.142±2.4e-4
forget05	NPO	0.26 [0.26, 0.31] / 0.62	1.32 [1.31, 1.41]	0.03 (loss)	[-0.02, 0.02] (0.001)	0.780±0.014
forget05	SimNPO	0.32 [0.30, 0.35] / 0.69	2.62 [2.43, 2.78]	-0.17 (loss)	[-0.02, 0.02] (0.001)	0.065±0.001

Table 7: **TOFU/Llama-3.2-1B: $\theta^\star$-only K-channel scores vs. realized per-example gaps over independent optimizer seeds.** Unlearning uses the open-unlearning full-finetune recipe (lr 10^{-5}, 10 epochs, 8–10 per-epoch checkpoints per trajectory) for {forget01, forget05} × {GradDiff, NPO} and forget05 SimNPO, with 3 optimizer seeds each. The reference is the released retain99/95 model. Scores are computed once at $\boldsymbol{\theta}^\star$ in the LoRA chart (r=8, diagonal GGN + MC-Woodbury $\widehat{\boldsymbol{H}}$, λ=0.01) without access to $\boldsymbol{\theta}_r^\star$ or to the trajectories. With $\boldsymbol{g}_i$ the LoRA-chart gradient of the answer NLL of example i at $\boldsymbol{\theta}^\star$, the forget score is $\boldsymbol{g}_j^\top \widehat{\boldsymbol{H}}^{-1} \boldsymbol{g}_j$ and the retain score is $\sum_{j\in\mathcal{D}_f} |\boldsymbol{g}_i^\top \widehat{\boldsymbol{H}}^{-1} \boldsymbol{g}_j|$. Actual gaps are the per-example $|\mathrm{NLL}_u - \mathrm{NLL}_{\mathrm{ret}}|$ on the forget set (top) and on 400 retain anchors (bottom). Per-epoch checkpoints (checkpoint 0 excluded) are aggregated by the median within a trajectory and then averaged across seeds, with hierarchical-bootstrap CIs (B=1,000). The Δ columns subtract the Spearman of a cheap per-example baseline, namely the NLL at $\boldsymbol{\theta}^\star$ on the forget channel and the gradient norm (or the NLL where marked "loss") on the retain channel. The null columns give the 95% interval of the within-checkpoint permutation null (1,000 permutations) and the two-sided p of the observed Spearman. Token KL is the mean answer-token $\mathrm{KL}(p_{\boldsymbol{\theta}_r^\star} \| p_{\boldsymbol{\theta}_u})$ at the final checkpoint. The scores identify examples with the largest NLL deviations from retraining on both channels, above the permutation null in every cell except the NPO forget01 forget channel, and are comparable to the cheap per-example baselines.

Observation. The $\boldsymbol{\theta}^\star$-only scores carry usable per-example signal at LLM scale. The retain score ranks the retain probes that the unlearners disturb most, with Spearman 0.26–0.49 and Q4/Q1 gap ratios of 1.3–3.7, far outside the permutation null (±0.02), on both splits and for all three methods (SimNPO on forget05 only), across independent seeds. The forget score ranks the forget examples whose treatment departs most from retraining, with Spearman 0.14–0.29 and Q4/Q1 ratios of 1.1–1.8, and is at the null only for NPO on forget01 ($p = 0.99$). Both scores are available before any unlearning run, so they can flag which examples will be hard to unlearn faithfully and where an evaluation of the unlearned model should look. Two limits are visible in the same table. The scores are comparable to, not clearly better than, the per-example loss or gradient norm at $\boldsymbol{\theta}^\star$ (margins between -0.17 and $+0.06$), and they are not calibrated predictions of retraining-gap magnitude. A separate diagnostic in Table 8 evaluates the profile and scale of NLL movement from $\boldsymbol{\theta}^\star$ on the same trajectories. The per-example profile of the NLL movement away from $\boldsymbol{\theta}^\star$ follows the scores (cosine above its permutation null on every row, centered Pearson 0.16–0.58 forget and 0.19–0.39 retain), but the relative error after fitting one scale stays at 0.6–0.8 and the fitted scale varies by 27–65%

across checkpoints and seeds. Several differences from the local validation may contribute. The curvature is a diagonal-plus-rank-4 GGN on a LoRA chart of 5.6×10^6 of the 1.2×10^9 parameters, the endpoints are early-stopped ten-epoch runs, and these scores target sequence NLL rather than the logit gaps of Section 5.2. These endpoints sit 0.07–1.2 nats of retain KL from a cold-start $\boldsymbol{\theta}_r^\star$, far beyond the $\leq 10^{-5}$-nat regime of the positive local validation in Section 5.2. Computing the scores on a half-frozen LoRA chart with an exact GGN on the active half changes little. These comparisons do not isolate the effects of chart restriction, curvature approximation, endpoint nonstationarity, or target choice; whether richer curvature or more local endpoints improve prediction remains open. The output-space KL of the final checkpoints (0.07–1.2 nats on the retain probes) confirms that these recipes do not leave the retain channel untouched.

Split	Method	P f/r	Cosine f/r	perm. null f/r	RelErr f/r	scale CV (within/pooled)
forget01	GradDiff	0.52±0.07 / 0.33±0.03	0.78±0.02 / 0.72±0.06	0.59±0.01 / 0.57±6.2e-3	0.63±0.03 / 0.69±0.06	0.51 / 0.65
forget05	GradDiff	0.58±0.03 / 0.23±5.6e-3	0.75±0.01 / 0.72±3.5e-3	0.52±6.3e-3 / 0.64±4.2e-3	0.67±0.02 / 0.69±3.6e-3	0.50 / 0.51
forget01	NPO	0.40±0.09 / 0.27±0.03	0.74±0.03 / 0.70±0.02	0.62±0.01 / 0.60±5.0e-3	0.67±0.03 / 0.71±0.02	0.39 / 0.49
forget05	NPO	0.16±0.03 / 0.19±0.01	0.61±6.6e-3 / 0.75±1.3e-3	0.57±4.3e-3 / 0.70±2.9e-3	0.79±5.1e-3 / 0.67±1.5e-3	0.43 / 0.43
forget05	SimNPO	0.43±0.01 / 0.25±0.02	0.62±0.02 / 0.58±0.03	0.33±0.01 / 0.44±6.0e-3	0.78±0.01 / 0.82±0.02	0.27 / 0.28
Dropout-LoRA score arm: A frozen, exact GGN on the active B half, vocab-dimension MC Woodbury						
forget01	GradDiff	0.55±0.08 / 0.39±0.02	0.78±0.03 / 0.76±0.05	0.57±0.01 / 0.60±5.9e-3	0.63±0.04 / 0.64±0.06	0.49 / 0.64
forget05	GradDiff	0.57±0.03 / 0.35±4.2e-3	0.74±0.01 / 0.83±1.3e-3	0.52±6.4e-3 / 0.73±3.3e-3	0.67±0.02 / 0.56±2.0e-3	0.50 / 0.51
forget01	NPO	0.33±0.17 / 0.31±0.04	0.71±0.05 / 0.74±0.02	0.60±0.01 / 0.62±4.7e-3	0.70±0.05 / 0.68±0.03	0.37 / 0.46
forget05	NPO	0.16±0.03 / 0.23±0.02	0.61±6.2e-3 / 0.84±1.4e-3	0.56±4.3e-3 / 0.80±2.3e-3	0.80±4.7e-3 / 0.54±2.2e-3	0.43 / 0.43
forget05	SimNPO	0.44±0.01 / 0.39±0.01	0.63±0.02 / 0.67±0.02	0.33±9.9e-3 / 0.51±4.9e-3	0.78±0.01 / 0.74±0.02	0.27 / 0.27

Table 8: **TOFU/Llama-3.2-1B: per-example profile of the NLL movement from $\boldsymbol{\theta}^\star$ against the $\boldsymbol{\theta}^\star$-only scores.** Predictor-only re-analysis of the trajectories of Table 7 (3 seeds, 7–10 checkpoints each) with the same forget and retain scores, which carry no method weights $\boldsymbol{W}^-$ and no ρ/n factor. The actual quantity is the per-example sequence-NLL movement $\mathrm{nll}(\mathrm{ckpt}) - \mathrm{nll}(\boldsymbol{\theta}^\star)$ (absolute value on the retain channel). This is the movement away from the original model and not the gap to retraining, so the test asks only whether the per-example profile of that movement is predicted up to one scale. P is the per-example Pearson correlation (its permutation null is 0). Cosine is the cosine between the two vectors (median over checkpoints per seed, mean±sd over seeds) and perm. null is its within-checkpoint permutation baseline (500 permutations), so the profile is predicted only where Cosine exceeds its null. RelErr is $\min_s \|s\,k - \Delta\|_2 / \|\Delta\|_2$ after fitting the single free scale, and scale CV is the coefficient of variation of that fitted scale (within-seed / pooled). The second block computes the scores on the half-frozen LoRA chart ($\boldsymbol{A}$ frozen, exact GGN over the active $\boldsymbol{B}$ coordinates, S=4 vocabulary-dimension MC pseudo-gradients). NPO/SimNPO rows use the unweighted scores and are therefore conservative checks, not calibrated $\boldsymbol{W}^-$ predictions.

D Additional Experimental Validations

These additional experiments test the framework along axes complementary to Section 5 and are referenced from there but not relied on for the main claims.

ProjGD and Newton-step behavior in the regime comparison. ProjGD has nearly unchanged retain RMS across regimes ($0.438 \rightarrow 0.440$) but larger output gaps than GradDiff, NPO, and SimNPO. Its projection removes components along sampled retain-loss gradients; this does not guarantee unchanged retain logits, and the regime comparison does not isolate the cause of its larger gaps. The Newton step, which recovers $\boldsymbol{\theta}_r^\star$ exactly in the linear case (Table 1), shows the largest nonlinear output gaps ($0.46 \rightarrow 0.73$ on the forget set). This illustrates the limits of the local quadratic approximation. The comparison does not separately identify GGN mismatch, higher-order error, or imperfect reference stationarity.

Linear closed-form sanity check (Theorem 2). On a synthetic ridge regression problem we compare the analytical gap of Eq. (12) to the numerical solution of the GradDiff stationarity equations. The relative error is $< 10^{-10}$ across all evaluation sets and all values of ρ inside the stability region, confirming that the closed form is exact in the linear setting.

Full local validation and controls (Eq. (22)). Tables 9 and 10 expand the local GradDiff/NPO/SimNPO validation and the ProjGD and Pure GA/IMU comparisons of Table 3, under the protocol of Section 5.2. Panel A adds the held-out test channel $\boldsymbol{K}_{af}$ ($P_a^K = 0.80$–0.96 on

MNIST and 0.71–0.95 on CIFAR-10 for GradDiff, NPO, and ProjGD), the relative ℓ_2 error of the predicted gap vector, the output-space KL to $\boldsymbol{\theta}_r^\star$, and the IMU rows, which behave like Pure GA. For Pure GA and IMU the correlations are small or negative (between -0.51 and 0.32) under both the K-geometry and the Euclidean-Gram predictor, and their endpoints sit 0.7–3 nats of KL away from $\boldsymbol{\theta}_r^\star$, so these rows are outside the local regime as discussed in Section 6. Panel B sweeps the removal scale $s \in \{0.05, 0.2, 0.5, 1, 2\}$. P_f^K stays at or above 0.96 (MNIST) and 0.88 (CIFAR-10) up to $s = 2$, while the endpoint displacement from $\boldsymbol{\theta}_r^\star$ grows by more than an order of magnitude and the class-regime forget cross-entropy rises. Table 11 is the β-swap control referenced in Section 5.1.

Panel A: agreement at the operating point $s = 0.1$

Regime	Method	P_f^K	P_f^G	P_r^K	P_r^G	P_a^K	RelErr f/r	KL($\boldsymbol{\theta}_r^\star \Vert \boldsymbol{\theta}_u$) f/r
Random	GradDiff	0.98±0.02	0.83±0.06	0.72±0.10	0.40±0.06	0.87±0.05	0.30±0.17 / 0.77±0.11	5.3e-7±4.7e-7 / 9.2e-10±1.1e-9
Random	NPO†	0.98±0.02	0.83±0.06	0.72±0.10	0.40±0.05	0.87±0.06‡	0.31±0.17 / 0.77±0.11	4.8e-7±4.4e-7 / 7.9e-10±9.6e-10
Random	SimNPO†	0.94±0.12	0.52±0.22	0.74±0.09	0.24±0.05	—	0.31±0.21 / 0.75±0.10	1.5e-7±2.2e-7 / 1.8e-10±2.1e-10
Random	ProjGD	0.96±0.02	0.94±0.02	0.75±0.01	0.83±0.03	0.80±0.04	13.76±2.63 / 6.00±0.53	3.5e-9±2.6e-9 / 6.5e-11±7.9e-11
Random	Pure GA	5.5e-3±0.01	0.02±0.02	0.05±0.03	0.04±0.04	0.04±0.02	1.00±4.9e-6 / 1.00±1.9e-6	1.82±0.03 / 1.90±0.02
Random	IMU	5.5e-3±0.01	-0.04±0.02	0.05±0.03	-0.05±0.03	0.04±0.02	1.00±4.9e-6 / 1.00±1.9e-6	1.82±0.03 / 1.90±0.02
Class	GradDiff	0.98±4.9e-3	0.81±0.04	0.95±0.02	0.59±0.09	0.96±9.7e-3	0.31±0.07 / 0.31±0.05	7.9e-6±1.0e-5 / 7.3e-8±1.3e-7
Class	NPO†	0.98±4.9e-3	0.81±0.04	0.95±0.02	0.58±0.09	0.96±9.4e-3‡	0.31±0.07 / 0.30±0.05	4.7e-6±6.1e-6 / 4.4e-8±7.5e-8
Class	SimNPO†	0.98±6.2e-3	-0.03±0.13	0.97±8.2e-3	2.5e-4±0.18	—	0.27±0.08 / 0.25±0.03	1.4e-6±1.9e-6 / 1.4e-8±1.7e-8
Class	ProjGD	0.95±0.01	0.97±0.01	0.87±0.03	0.87±0.05	0.92±0.01	8.43±1.37 / 8.28±0.75	2.7e-7±4.0e-7 / 2.8e-9±5.2e-9
Class	Pure GA	-0.36±0.06	-0.39±0.05	-0.11±0.03	-0.23±0.07	-0.15±0.01	1.00±7.3e-4 / 1.00±3.9e-5	1.24±0.16 / 1.93±0.02
Class	IMU	-0.36±0.06	-0.39±0.05	-0.11±0.03	-0.23±0.06	-0.15±0.01	1.00±7.3e-4 / 1.00±3.9e-5	1.24±0.16 / 1.93±0.02

Panel B: agreement across the normalized removal scale s (GradDiff / NPO), $\rho = s\,n/\lambda_{\max}(\boldsymbol{K}_{ff})$

		P_f^K at s					at $s = 2$		
Regime	Method	0.05	0.2	0.5	1.0	2.0	P_r^K	$\Vert\boldsymbol{\theta}_u - \boldsymbol{\theta}_r^\star\Vert_2$	forget CE s=0.05→2
Random	GradDiff	0.98±0.02	0.98±0.01	0.98±0.02	0.98±0.02	0.97±0.03	0.73±0.10	0.016±0.009	0.32±0.03 → 0.32±0.03
Random	NPO	0.98±0.02	0.98±0.02	0.98±0.02	0.98±0.02	0.97±0.03	0.72±0.11	0.015±0.008	0.32±0.03 → 0.32±0.03
Class	GradDiff	0.98±4.9e-3	0.98±5.0e-3	0.97±6.2e-3	0.96±0.02	0.97±0.01	0.92±0.07	0.093±0.063	4.41±0.25 → 4.65±0.36
Class	NPO	0.98±5.1e-3	0.97±5.2e-3	0.97±0.02	0.97±8.7e-3	0.96±0.01	0.91±0.08	0.070±0.043	4.41±0.25 → 4.59±0.31

Table 9: **MNIST: full local gap validation and controls.** The setting is a 2-layer $\tanh$ MLP ($784 \to 24 \to 10$, d=19,090), n=4000, λ=10^{-2}, exact fp64 throughout. Protocol (Appendix B). The warm retrain is polished to stationarity under the full-n normalization and $\boldsymbol{H}_r$ is the exact fp64 GGN. GradDiff, NPO, SimNPO, Pure GA, and IMU endpoints are tracked from $\boldsymbol{\theta}_r^\star$ by Newton continuation. ProjGD uses projected-gradient dynamics and the unprojected predictor. GradDiff/NPO/SimNPO test the local predictors of Table 4; Pure GA and IMU are controls outside the local regime. Mean±sd over 10 seeds. At $s = 0.1$ the GradDiff endpoint lies 0.0008 (random) / 0.0040 (class) from $\boldsymbol{\theta}_r^\star$ in parameter norm, versus $\Vert\boldsymbol{\theta}^\star - \boldsymbol{\theta}_r^\star\Vert_2 = 3.39$ / 4.06, illustrating the locality of the retain-anchored validation. Columns are as in Table 3. P_a^K is the per-logit Pearson correlation on the held-out test subsample ($\boldsymbol{K}_{af}$ channel), RelErr is the relative ℓ_2 error of the predicted gap vector, and KL is the output-space $\mathrm{KL}(p_{\boldsymbol{\theta}_r^\star} \Vert p_{\boldsymbol{\theta}_u})$ at the endpoint. The shuffled-anchor control gives $|P| \le 0.20$. †method-specific $(\boldsymbol{K}, \boldsymbol{W}^-)$ scoring with frozen reference weights for NPO/SimNPO. ‡test channel scored with uniform weights (for NPO the weights at $\boldsymbol{\theta}_r^\star$ are nearly uniform), not stored for SimNPO. Panel B repeats the construction at larger removal scales. P_f^K stays at or above 0.96 up to $s = 2$, while the endpoint moves farther from $\boldsymbol{\theta}_r^\star$ and the class-regime forget cross-entropy rises. Here s is normalized by $\lambda_{\max}(\boldsymbol{K}_{ff})$ alone, so $s = 1$ is the squared-loss boundary of Corollary 1. Under cross-entropy the softmax curvature $\boldsymbol{B}_f^-$ enters the stability factor of Eq. (22), and the tracked fixed point continues past $s = 1$.

Stability divergence as $s \to 1$ (Corollary 1). Sweeping the stability fraction $s = \frac{\rho}{n}\lambda_{\max}(\boldsymbol{K}_{ff}) \in \{0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95\}$, the predicted and actual logit deviations increase as the inverse factor approaches the stability boundary. The norm bound of Eq. (13) has the correct denominator but is extremely loose in absolute magnitude, as expected from a worst-case operator-norm argument.

Panel A: agreement at the operating point $s = 0.1$

Regime	Method	P_f^K	P_f^G	P_r^K	P_r^G	P_a^K	RelErr f/r	KL($\boldsymbol{\theta}_r^\star \| \boldsymbol{\theta}_u$) f/r
Random	GradDiff	0.89±0.03	0.74±0.04	0.82±0.05	0.63±0.06	0.83±0.04	0.78±0.03 / 0.84±0.06	7.1e-8±3.8e-8 / 4.9e-9±3.4e-9
Random	NPO†	0.89±0.03	0.74±0.04	0.82±0.05	0.63±0.06	0.83±0.04‡	0.78±0.03 / 0.84±0.06	6.8e-8±3.6e-8 / 4.7e-9±3.1e-9
Random	SimNPO†	0.91±8.6e-3	0.60±0.11	0.86±0.01	0.51±0.09	—	0.68±0.03 / 0.69±0.05	1.1e-8±4.0e-9 / 1.5e-9±3.0e-10
Random	ProjGD	0.79±0.03	0.74±0.05	0.71±0.04	0.67±0.06	0.71±0.03	8.29±1.20 / 6.82±0.90	1.6e-9±8.5e-10 / 2.3e-10±1.2e-10
Random	Pure GA	0.22±0.07	0.26±0.07	0.29±0.08	0.27±0.08	0.24±0.07	1.00±9.5e-6 / 1.00±7.8e-6	2.72±1.07 / 2.71±1.06
Random	IMU	0.24±0.06	0.07±0.04	0.32±0.08	0.07±0.05	0.26±0.06	1.00±6.0e-6 / 1.00±5.3e-6	2.94±0.25 / 2.91±0.26
Class	GradDiff	0.97±0.01	0.94±0.02	0.87±0.09	0.74±0.07	0.94±0.04	0.78±0.14 / 0.61±0.16	3.0e-6±1.4e-6 / 3.2e-9±2.0e-9
Class	NPO†	0.97±0.01	0.94±0.02	0.87±0.09	0.74±0.07	0.94±0.04‡	0.78±0.14 / 0.61±0.16	2.1e-6±9.6e-7 / 2.2e-9±1.4e-9
Class	SimNPO†	0.97±0.01	-0.14±0.17	0.86±0.10	0.04±0.17	—	0.79±0.15 / 0.63±0.17	1.4e-6±8.3e-7 / 1.6e-9±1.1e-9
Class	ProjGD	0.97±8.3e-3	0.98±8.8e-3	0.87±0.02	0.80±0.04	0.95±8.2e-3	13.82±1.19 / 11.48±0.62	4.5e-8±3.0e-8 / 5.2e-11±3.8e-11
Class	Pure GA	-0.51±0.07	-0.45±0.08	-0.04±0.01	-0.23±0.06	-0.12±8.3e-3	1.00±2.6e-4 / 1.00±4.6e-6	0.74±0.14 / 2.11±0.01
Class	IMU	-0.51±0.07	-0.44±0.08	-0.04±0.01	-0.23±0.06	-0.12±8.3e-3	1.00±2.6e-4 / 1.00±4.6e-6	0.74±0.14 / 2.11±0.01

Panel B: agreement across the normalized removal scale s (GradDiff / NPO), $\rho = s\,n/\lambda_{\max}(\boldsymbol{K}_{ff})$

		P_f^K at s					at $s = 2$		
Regime	Method	0.05	0.2	0.5	1.0	2.0	P_r^K	$\|\boldsymbol{\theta}_u - \boldsymbol{\theta}_r^\star\|_2$	forget CE s=0.05→2
Random	GradDiff	0.89±0.03	0.89±0.03	0.89±0.03	0.89±0.03	0.88±0.04	0.80±0.05	0.023±0.007	0.05±0.01 → 0.05±0.01
Random	NPO	0.89±0.03	0.89±0.03	0.89±0.03	0.89±0.03	0.88±0.05	0.80±0.05	0.023±0.007	0.05±0.01 → 0.05±0.01
Class	GradDiff	0.97±0.01	0.97±0.02	0.97±0.03	0.95±0.08	0.97±0.01	0.85±0.07	0.030±0.004	3.31±0.18 → 3.39±0.20
Class	NPO	0.97±0.01	0.97±0.02	0.97±0.02	0.96±0.05	0.97±0.01	0.86±0.07	0.025±0.003	3.31±0.18 → 3.38±0.19

Table 10: **CIFAR-10: full local gap validation and controls.** The setting is a 2-layer $\tanh$ MLP head ($512 \to 24 \to 10$, d=12,562) over frozen ResNet-18 penultimate features, n=4000, λ=10^{-2}, exact fp64 throughout. Protocol (Appendix B). The warm retrain is polished to stationarity under the full-n normalization and $\boldsymbol{H}_r$ is the exact fp64 GGN. GradDiff, NPO, SimNPO, Pure GA, and IMU endpoints are tracked from $\boldsymbol{\theta}_r^\star$ by Newton continuation. ProjGD uses projected-gradient dynamics and the unprojected predictor. GradDiff/NPO/SimNPO test the local predictors of Table 4; Pure GA and IMU are controls outside the local regime. Mean±sd over 10 seeds. At $s = 0.1$ the GradDiff endpoint lies 0.0011 (random) / 0.0015 (class) from $\boldsymbol{\theta}_r^\star$ in parameter norm, versus $\|\boldsymbol{\theta}^\star - \boldsymbol{\theta}_r^\star\|_2 = 0.91$ / 1.70, illustrating the locality of the retain-anchored validation. Columns are as in Table 3. P_a^K is the per-logit Pearson correlation on the held-out test subsample ($\boldsymbol{K}_{af}$ channel), RelErr is the relative ℓ_2 error of the predicted gap vector, and KL is the output-space $\mathrm{KL}(p_{\boldsymbol{\theta}_r^\star} \| p_{\boldsymbol{\theta}_u})$ at the endpoint. The shuffled-anchor control gives $|P| \le 0.18$. †method-specific $(\boldsymbol{K}, \boldsymbol{W}^-)$ scoring with frozen reference weights for NPO/SimNPO. ‡test channel scored with uniform weights (for NPO the weights at $\boldsymbol{\theta}_r^\star$ are nearly uniform), not stored for SimNPO. Panel B repeats the construction at larger removal scales. P_f^K stays at or above 0.88 up to $s = 2$, while the endpoint moves farther from $\boldsymbol{\theta}_r^\star$ and the class-regime forget cross-entropy rises. Here s is normalized by $\lambda_{\max}(\boldsymbol{K}_{ff})$ alone, so $s = 1$ is the squared-loss boundary of Corollary 1. Under cross-entropy the softmax curvature $\boldsymbol{B}_f^-$ enters the stability factor of Eq. (22), and the tracked fixed point continues past $s = 1$.

Regime	Method	β	$\bar{w}(\boldsymbol{\theta}_r^\star)$	forget CE	P_f^K	P_r^K	Cosine f/r	RelErr f/r
Random	NPO	0.1	0.99±1.2e-3	0.32±0.03	0.98±0.02	0.72±0.10	0.98±0.02 / 0.72±0.10	0.41±0.21 / 0.82±0.12
Random	NPO	2.5	0.87±0.02	0.32±0.03	0.60±0.26	0.43±0.10	0.60±0.26 / 0.43±0.10	4.93±2.08 / 2.98±0.62
Random	SimNPO	0.1	0.98±1.5e-3	0.32±0.03	0.98±0.02	0.72±0.10	0.98±0.02 / 0.72±0.10	0.42±0.22 / 0.84±0.12
Random	SimNPO	2.5	0.78±0.02	0.32±0.03	0.58±0.28	0.39±0.10	0.58±0.28 / 0.39±0.10	5.82±2.48 / 4.26±0.73
Class	NPO	0.1	0.79±0.01	4.41±0.25	0.98±5.2e-3	0.95±0.02	0.98±5.2e-3 / 0.95±0.02	0.60±0.11 / 0.45±0.06
Class	NPO	2.5	8.0e-4±1.6e-3	4.40±0.25	0.04±0.16	0.01±0.07	0.04±0.16 / 0.01±0.07	1444.73±1884.98 / 412.95±628.17
Class	SimNPO	0.1	0.78±0.01	4.41±0.25	0.98±5.2e-3	0.95±0.02	0.98±5.2e-3 / 0.95±0.02	0.61±0.11 / 0.45±0.06
Class	SimNPO	2.5	9.7e-5±4.8e-5	4.40±0.25	0.03±0.13	0.01±0.02	0.03±0.13 / 0.01±0.02	1500.59±1931.20 / 417.88±630.95

Table 11: **β-swap control on MNIST.** The four corners of the (method, β) square under the protocol of Table 9, scored with uniform weights (10 seeds). $\bar{w}(\boldsymbol{\theta}_r^\star)$ is the mean per-example forget weight at the reference in the deployed $2\sigma(\cdot)$ convention, $2\sigma(-\beta(\log p^{\boldsymbol{\theta}^\star} - \log p^{\boldsymbol{\theta}_r^\star}))$ for NPO and $2\sigma(\beta \log p^{\boldsymbol{\theta}_r^\star})$ for SimNPO. At $\beta = 0.1$ both methods agree with the predictor equally well. At $\beta = 2.5$ both collapse in the class regime. There the weights saturate ($\bar{w} \sim 10^{-4}$ when $\mathrm{CE}_f(\boldsymbol{\theta}_r^\star) \approx 4.4$), the effective forget term vanishes, and the endpoint stays at $\boldsymbol{\theta}_r^\star$ up to optimizer noise, so the forget CE equals the reference value and RelErr $\gg 1$. Swapping β swaps the behavior, so the NPO/SimNPO difference comes from β-induced weight saturation and not from the reference-based vs. reference-free form.

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 main claims made in the abstract and introduction are that we provide a unified view of machine unlearning methods, and that we provide a theoretical analysis of the output gaps of these methods. These claims are supported by the content of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes discussions of the limitations of the proposed methods, including assumptions made for theoretical analysis, practical approximation errors, and potential practicality issues.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: The paper provides a complete set of assumptions and proofs for all theoretical results, either in the main text or in the supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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 B describes the datasets, model architectures, training procedures, and evaluation metrics used in the experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [No]

   Justification: We plan to open-source our code; it is not yet publicly available. The datasets are public, and Appendix B describes the experimental protocols.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: Appendix B describes the datasets, model architectures, training procedures, and evaluation metrics used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: The paper reports results across multiple runs with error bars that capture variability across runs for the main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: Appendix B reports the detailed resource requirements for all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses potential societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: The paper properly credits all creators and original owners of assets used, and the licenses and terms of use are explicitly mentioned and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

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 assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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 nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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 crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.