Generalized Influence Functions for Better Model Change Estimates

Hyeonsu Lyu[1] **Jonggyu Jang**[3] **Sehyun Ryu**[4] **Hyun Jong Yang**[1,2]*

[1]Institute of New Media and Communications, Seoul National University, Seoul, Korea
[2]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[3]Chungnam National University, Daejeon, Korea
[4]Pohang University of Science and Technology, Pohang, Korea

hs.lyu@snu.ac.kr jgjang@cnu.ac.kr sh.ryu@postech.ac.kr hjyang@snu.ac.kr

Abstract

Influence functions (IFs) approximate the effect of removing or reweighting training data on a trained model, but their linear approximation often lacks the accuracy required for reliable post hoc model editing. We identify two culprits and propose generalized influence functions (GIFs) and pseudo-LiSSA (p-LiSSA) to address each in turn. First, classical IFs update all parameters indiscriminately, leading to unnecessary updates to data-irrelevant parameters. GIFs instead restrict edits to a *data-relevant subspace* and leave the remaining parameters fixed. We theoretically show that GIFs achieve a tighter error bound than classical IFs when the parameter subset is properly selected, and that effective subspace selection requires avoiding low-curvature directions while retaining a sufficiently large component of the target gradient. Second, LiSSA iterations are destabilized by negative Hessian eigenvalues, which existing methods handle via ℓ_2 damping—at the cost of a biased inverse-Hessian-vector product (iHVP) estimate. p-LiSSA eliminates this bias by solving the *restricted inverse-Hessian problem* without such damping. We further show that the presence of low-curvature directions slows p-LiSSA convergence, and that the data-relevant subspace selected by GIFs mitigates this. Across five image and text unlearning benchmarks, GIFs update only 10% of the parameters yet preserve accuracy up to 2.12 percentage points better than IF-based baselines, while improving unlearning performance by up to 2.73 percentage points. Moreover, p-LiSSA improves restricted iHVP estimation by reducing the residual by 9.6× over existing iHVP baselines, while using 19.8% less memory with comparable computation time. Code is available at `https://github.com/hslyu/GIF`.

1 Introduction

Influence functions (IFs) estimate how a trained model would change if selected training examples were removed, reweighted, or relabeled, by linearizing the effect of a data edit around the current optimum as an inverse-Hessian-vector product (iHVP), $-\mathbf{H}^{-1}\mathbf{g}$, where $\mathbf{H}$ is the Hessian of the training loss and $\mathbf{g}$ is the gradient induced by the selected training examples [25, 51]. This has motivated their use in post-hoc model editing [11, 42], including data removal [51, 52] and relabeling [16, 26]. In practice, however, as shown in previous studies [4, 14], classical IFs often

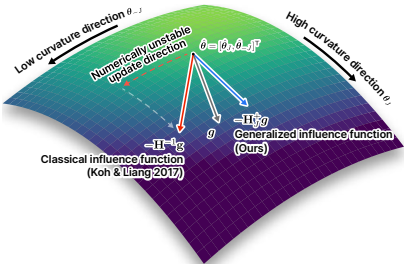


Figure 1: Conceptual illustration of classical and generalized IF directions.

*Corresponding author.

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

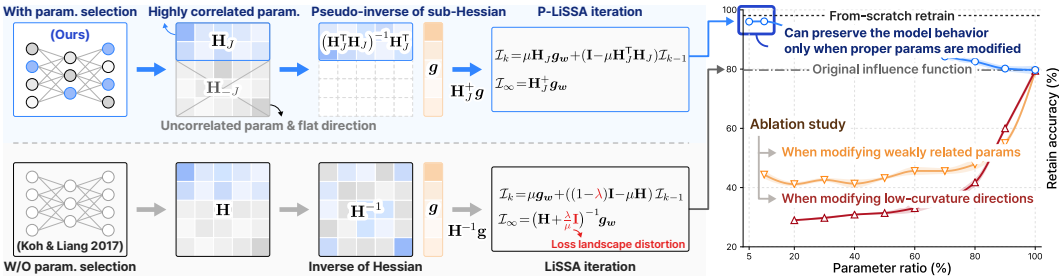


Figure 2: Comparison between GIFs and IFs; and between P-LiSSA and LiSSA.

fail to produce reliable model-change estimates, and have consequently seen more use in data attribution rather than in direct editing [25, 28, 38–40, 53].

This unreliability stems from the iHVP over the full parameter space, leading to two compounding problems. The first problem is that the classical IF update $\mathbf{H}^{-1}\mathbf{g}$ can deviate significantly along low-curvature directions, where the inverse Hessian amplifies even small gradient components into disproportionately large parameter changes, as depicted in Figure 1. This full-parameter view is also inconsistent with the observation that neural networks organize behavior through localized units and parameter subsets [6, 37]—the iHVP can therefore spread the model update into parameters weakly related to the edited data, causing unintended changes that should have been preserved. The second problem is the instability of LiSSA—the de facto standard iterative method to compute iHVP for IFs [1, 25]. For the convergence of LiSSA iteration, $\mathbf{H}$ needs to be positive definite; however, this condition can be easily violated in modern neural networks [4, 14]. A standard workaround is to use ℓ_2 damping, which computes $(\mathbf{H}+\lambda\mathbf{I})^{-1}\mathbf{g}$ instead of $\mathbf{H}^{-1}\mathbf{g}$ [3, 5, 24]. The ℓ_2 damping stabilizes the iteration but distorts the loss landscape that IFs are meant to approximate. Moreover, recent work has emphasized that iHVP and curvature approximations can strongly affect the quality of influence estimates [20, 23]. To summarize, classical IFs face two coupled obstacles: **(i)** IFs change parameters that should be preserved, and **(ii)** the numerically unstable LiSSA iteration.

To address the first problem, we introduce *generalized influence functions (GIFs)*, which decompose the model into target parameters and fixed parameters, then restrict the iHVP computation to the target parameters. As shown in the top row of Figure 2, GIFs restrict the edit to parameters that are most relevant to the edit data, in contrast to classical IFs (bottom row) that spread the update across all parameters. We theoretically show that GIFs offer a tighter approximation error bound than classical IFs when the parameter subset is properly selected, and that effective subspace selection requires avoiding low-curvature directions while preserving a sufficiently large component of the target gradient $\mathbf{g}$ (in **Theorems 1 and 2**). Based on these results, we examine a range of parameter-selection criteria and find, as shown in the right panel of Figure 2, a criterion favoring data-relevant parameters closely matches the behavior of the from-scratch retrained model even with a small number of parameter updates, while selecting weakly related parameters or low-curvature directions degrades accuracy.

To address the second problem, we propose *pseudo-LiSSA (p-LiSSA)*, a damping-free solver for the restricted inverse sub-Hessian problem induced by GIFs, as shown in Figure 2. By restricting the solve to the target parameter subspace, p-LiSSA operates on a positive-definite Gram matrix instead of the possibly indefinite full Hessian, thereby mitigating instability. To support the claim, we theoretically show that p-LiSSA converges to the exact GIF direction under a simple scaling condition without extra damping (in **Theorem 3**), where the convergence speed can be accelerated by avoiding low curvature. We further show that p-LiSSA with two independent mini-batches is an unbiased estimator of the truncated Neumann series, and its expectation converges to the GIF direction (in **Theorem 4**). We note that p-LiSSA also serves as a damping-free iHVP solver for classical full-parameter IFs, which arise as the special case of GIFs when all parameters are editable.

In experiments, we numerically evaluate whether GIFs improve the model-editing performance in class-level unlearning, where the edited model should forget target data while preserving performance on retained data. We then examine various parameter-selection criteria to confirm that data-relevance and curvature-awareness are both essential for reliable model editing. Finally, we benchmark p-LiSSA against existing iHVP solvers and show that it reduces the iHVP residual error by 9.6× over the best competing solver at lower memory cost.

2 Preliminaries on Classical Influence Functions

Notations and definitions. We define a model $\boldsymbol{\theta} = [\theta_1, ..., \theta_n]^\mathsf{T} \in \mathbb{R}^n$ and m training examples $z_1, ..., z_m \in \mathcal{Z}$. Let $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^m \ell(z_i, \boldsymbol{\theta})$ be the empirical risk, where $\ell(\cdot)$ is twice continuously differentiable in a neighborhood of the reference minimizer. The empirical risk minimizer is defined as $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}(\boldsymbol{\theta})$. We define the ϵ-weighted empirical risk minimizer as

$$\hat{\boldsymbol{\theta}}(\mathbf{1} + \epsilon\boldsymbol{w}) = \arg\min_{\boldsymbol{\theta}\in\Theta} \frac{1}{m}\sum_{i=1}^m (1 + \epsilon w_i)\ell(z_i, \boldsymbol{\theta}), \tag{1}$$

where vector $\boldsymbol{w} = [w_1, ..., w_m]^\mathsf{T} \in \{0, 1\}^m$ indicates which training examples are edited.

To approximate the parameter change induced by a finite perturbation of the empirical risk, we define the following finite influence update.

Definition 1 (Finite influence function)**.** Let $\mathcal{L}(\boldsymbol{\theta})$ be twice continuously differentiable in a neighborhood of $\hat{\boldsymbol{\theta}}$, and assume $\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}$ is invertible. For $\Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} = \hat{\boldsymbol{\theta}}(\mathbf{1} + \epsilon\boldsymbol{w}) - \hat{\boldsymbol{\theta}}$, the finite influence function and its parameter displacement are defined as

$$\mathcal{I}(\boldsymbol{w}, \hat{\boldsymbol{\theta}} \mid \epsilon) := -\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}} \in \mathbb{R}^n, \quad \Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} := \epsilon\mathcal{I}(\boldsymbol{w}, \hat{\boldsymbol{\theta}}|\epsilon) = -\epsilon\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}}, \tag{2}$$

where $\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}} = \frac{1}{m}\sum_{i=1}^m (1 + \epsilon w_i)\nabla^2\ell\big(z_i, \hat{\boldsymbol{\theta}}\big)$ and $\boldsymbol{g}_{\boldsymbol{w}} = \frac{1}{m}\sum_{i=1}^m w_i\nabla\ell\big(z_i, \hat{\boldsymbol{\theta}}\big)$.

The quantity $\mathcal{I}(\boldsymbol{w}, \hat{\boldsymbol{\theta}}|\epsilon)$ is a parameter-sized update direction induced by the perturbation from $\ell(\cdot)$ to $(1+\epsilon)\ell(\cdot)$ on the selected data points. This update is obtained by linearizing the first-order optimality condition of (1), which can be mathematically expressed as

$$0 = \nabla\mathcal{L}\big(\hat{\boldsymbol{\theta}}(\mathbf{1} + \epsilon\boldsymbol{w})\big) + \frac{\epsilon}{m}\sum_{i=1}^m w_i\nabla\ell\big(z_i, \hat{\boldsymbol{\theta}}(\mathbf{1} + \epsilon\boldsymbol{w})\big). \tag{3}$$

Let $\Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} = \hat{\boldsymbol{\theta}}(\mathbf{1} + \epsilon\boldsymbol{w}) - \hat{\boldsymbol{\theta}}$. Applying a Taylor expansion to both terms of (3) around $\hat{\boldsymbol{\theta}}$ gives

$$0 = \frac{1}{m}\sum_{i=1}^m (1 + \epsilon w_i)\nabla\ell\big(z_i, \hat{\boldsymbol{\theta}}\big) + \frac{1}{m}\sum_{i=1}^m (1 + \epsilon w_i)\nabla^2\ell\big(z_i, \hat{\boldsymbol{\theta}}\big)\Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} + \boldsymbol{r}(\epsilon) \tag{4}$$

$$= \epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}\Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} + \boldsymbol{r}(\epsilon). \tag{5}$$

where $\boldsymbol{r}(\epsilon)$ collects higher-order terms. Neglecting the higher-order remainder gives the linearized approximation $\Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} \approx -\epsilon\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}} = \epsilon\mathcal{I}(\boldsymbol{w}, \hat{\boldsymbol{\theta}}|\epsilon)$, which is exactly the linearized parameter displacement induced by the finite influence update in **Definition 1**.

We study the finite-step linearized update rather than the infinitesimal derivative at $\epsilon = 0$ as in [24]. This choice is motivated by model editing and unlearning, where the goal is to approximate the actual parameter displacement produced by removing or modifying a subset of training points. Using $\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}$ allows the local curvature surrogate to reflect the edited weighting pattern itself, yielding a Newton-style update tailored to the perturbed objective. This finite formulation is also aligned with the Newton-step approximation used in prior work on group effects and data removal [24].

Numerical vulnerability of LiSSA. The authors of [1, 25] propose an iterative method, LiSSA, to numerically compute iHVPs using HVPs. LiSSA is based on the Neumann expansion [43], $\mathbf{H}^{-1} = \sum_{i=0}^\infty (\mathbf{I} - \mathbf{H})^i$, which converges if $\rho(\mathbf{H}) < 1$.[2] In practice, however, LiSSA introduces scaling and damping into the recursion [3, 5, 24] as modern neural networks are unlikely to meet the spectral condition as explained in Appendix B. With a scaling parameter $\mu > 0$ and a damping parameter $\lambda > 0$, the LiSSA iteration is implemented as

$$\mathcal{I}_k = \mathcal{I}_0 + \left((1 - \lambda)\mathbf{I} - \mu\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}\right)\mathcal{I}_{k-1}, \quad \mathcal{I}_0 = \mu\boldsymbol{g}_{\boldsymbol{w}}, \tag{6}$$

where $\mathcal{I}_k$ converges to $\left(\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}} + \frac{\lambda}{\mu}\mathbf{I}\right)^{-1}\boldsymbol{g}_{\boldsymbol{w}} \approx -\mathcal{I}(\boldsymbol{w}, \hat{\boldsymbol{\theta}}|\epsilon)$ only if $\rho\left((1 - \lambda)\mathbf{I} - \mu\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}\right) < 1$. Then, the resulting estimate no longer converges to $\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}} = -\mathcal{I}(\boldsymbol{w}, \hat{\boldsymbol{\theta}}|\epsilon)$, but instead targets

[2] $\rho(\cdot)$: The spectral radius of a matrix is the largest absolute value among its eigenvalues.

a damped surrogate of this direction. This facilitates numerical convergence, but also distorts the loss landscape that IFs are intended to approximate. Such distortion is especially undesirable for finite model edits, where accurately approximating the parameter change is more critical than merely estimating data attribution.

3 Generalized Influence Functions

We introduce GIFs, which compute influence over selected parameters by separating the model parameters into **(i) target parameters**, which are editable, and **(ii) fixed parameters**, which remain unchanged. We then analyze when GIFs can better approximate retraining behavior than IFs, and study how to select suitable target parameters. All proofs are provided in the appendices.

Definition and derivation. Let $J = \{j_1, ..., j_k\}$ denote an index set of k target parameters. We define the target parameters $\boldsymbol{\theta}_J = [\theta_j]_{j\in J}^\mathsf{T} \in \mathbb{R}^k$ and the fixed parameters as $\boldsymbol{\theta}_{-J} = [\theta_j]_{j\notin J} \in \mathbb{R}^{n-k}$. For brevity, we rearrange the parameters as $\hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\theta}}_J, \hat{\boldsymbol{\theta}}_{-J}]^\mathsf{T}$. Under the same reordering, we write the Hessian as $\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}} = [\mathbf{H}_J \mid \mathbf{H}_{-J}]$, where $\mathbf{H}_J \in \mathbb{R}^{n\times k}$ collects the columns indexed by J and $\mathbf{H}_{-J} \in \mathbb{R}^{n\times(n-k)}$ collects the remaining columns. Then, the GIF is formally defined as follows.

Definition 2 (Generalized influence function)**.** Assume that $\mathbf{H}_J$ has full column rank so that $(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J)^{-1}$ exists. Then, for $\mathbf{H}_J^+ = \left(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J\right)^{-1}\mathbf{H}_J^\mathsf{T}$ and $\Delta\hat{\boldsymbol{\theta}}_{J,\mathbf{1}+\epsilon\boldsymbol{w}} = \hat{\boldsymbol{\theta}}_J(\mathbf{1}+\epsilon\boldsymbol{w}) - \hat{\boldsymbol{\theta}}_J$, the *generalized influence function* for $\boldsymbol{\theta}_J$ and its parameter displacement are defined as

$$\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J | \hat{\boldsymbol{\theta}}, \epsilon\big) := -\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}, \quad \Delta\hat{\boldsymbol{\theta}}_{J,\mathbf{1}+\epsilon\boldsymbol{w}} := -\epsilon\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}} = \epsilon\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J \mid \hat{\boldsymbol{\theta}}, \epsilon\big). \tag{7}$$

The derivation of GIFs also begins with the first-order optimality condition (3) as in the classical IFs. As $\hat{\boldsymbol{\theta}}_{-J}$ are assumed to be stationary, the model changes from $\hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\theta}}_J, \hat{\boldsymbol{\theta}}_{-J}]^\mathsf{T}$ to $\hat{\boldsymbol{\theta}}(\epsilon, \boldsymbol{w}) = [\hat{\boldsymbol{\theta}}_J(\epsilon, \boldsymbol{w}), \hat{\boldsymbol{\theta}}_{-J}]^\mathsf{T}$. Then, a Taylor expansion of the first-order optimality condition in (5) gives the linearized full-model condition that the restricted update seeks to satisfy

$$0 = \epsilon\boldsymbol{g}_{\boldsymbol{w}} + [\mathbf{H}_J|\mathbf{H}_{-J}]\left(\begin{bmatrix}\hat{\boldsymbol{\theta}}_J(\mathbf{1}+\epsilon\boldsymbol{w})\\ \hat{\boldsymbol{\theta}}_{-J}\end{bmatrix} - \begin{bmatrix}\hat{\boldsymbol{\theta}}_J\\ \hat{\boldsymbol{\theta}}_{-J}\end{bmatrix}\right) = \epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_J\Delta\hat{\boldsymbol{\theta}}_{J,\mathbf{1}+\epsilon\boldsymbol{w}} \tag{8}$$

as the $\hat{\boldsymbol{\theta}}_{-J}$ terms cancel out. Since $\mathbf{H}_J \in \mathbb{R}^{n\times k}$ and $\hat{\boldsymbol{\theta}}_J(\mathbf{1}+\epsilon\boldsymbol{w}) \in \mathbb{R}^k$ for $k < n$, (8) is an overdetermined linear problem and need not admit an exact solution. We therefore define the restricted update as the minimum-residual problem [7]

$$\hat{\boldsymbol{\theta}}_J(\mathbf{1}+\epsilon\boldsymbol{w}) := \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^k} \left\|\epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_J(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_J)\right\|_2^2. \tag{9}$$

Then, the minimum-residual solution is given by

$$\Delta\hat{\boldsymbol{\theta}}_{J,\mathbf{1}+\epsilon\boldsymbol{w}} := \hat{\boldsymbol{\theta}}_J(\mathbf{1}+\epsilon\boldsymbol{w}) - \hat{\boldsymbol{\theta}}_J = -\epsilon\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}} = \epsilon\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J \mid \hat{\boldsymbol{\theta}}, \epsilon\big). \tag{10}$$

The classical IFs as a special case. GIFs reduce to the classical influence update (2) when all parameters are editable, i.e., $\hat{\boldsymbol{\theta}}_J = \hat{\boldsymbol{\theta}}$. Then, as $\mathbf{H}^+ = \mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}^{-1}$, we have

$$\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}} | \hat{\boldsymbol{\theta}}, \epsilon\big) = -\mathbf{H}_{\mathbf{1}+\epsilon\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}} = \mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}} | \epsilon\big). \tag{11}$$

Thus, GIFs are an extension of the classical IF formulation by choosing the parameter set J.

3.1 The Model Edit Fidelity of Generalized Influence Functions

This subsection studies the approximation quality of GIFs under finite-step update. Without loss of generality, we set $\epsilon = -1$ (i.e., unlearning $\boldsymbol{w}$), while the same argument extends to any value of ϵ. We define the model edited by the GIF as

$$\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w}) := \hat{\boldsymbol{\theta}} + \Delta \text{ for } \Delta = \left[\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}} \mid \mathbf{0}\right]^\mathsf{T} \tag{12}$$

which corresponds to the restricted linearized displacement $-\epsilon\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}}$ evaluated at $\epsilon = -1$.

We first analyze GIFs from the perspective of retraining fidelity. In model editing, the finite influence update is useful only insofar as it approximates the model that would have been obtained by retraining on the edited empirical risk. Thus, we measure how close the edited model $\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})$ is to the from-scratch retrained model $\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})$ under an arbitrary evaluation functional. Let f be an evaluation functional on $\Theta \subset \mathbb{R}^n$. We define the approximation error as

$$\mathrm{Err}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) := f\big(\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})\big) - f\big(\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})\big). \tag{13}$$

Typical choices of $f(\cdot)$ include the test loss, $f(\theta) = \ell(x_{\mathrm{test}}, y_{\mathrm{test}}; \theta)$, and the self-loss on the removed samples, $f(\theta) = \sum_i w_i \ell(x_i, y_i; \theta)$. The former measures the effect of the edit on generalization, while the latter measures its effect on the data points selected by $\boldsymbol{w}$.

The following theorem shows that the GIF update admits a tighter approximation error bound than the full-parameter IF update when the selected parameters are better conditioned and preserve a sufficiently large component of the target gradient.

Theorem 1 (Error bound of GIFs)**.** *Let the functional $f(\cdot)$ and the Hessian map $\nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\cdot)$ be C_f- and $C_{\mathbf{H}}$-Lipschitz continuous on a convex neighborhood $\mathcal{U}$ containing $\hat{\boldsymbol{\theta}}$, $\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})$, and the retrained optimum $\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})$. Assume $\nabla \mathcal{L}_{\mathbf{1}}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$. If $\mathcal{L}_{\mathbf{1}-\boldsymbol{w}}$ is $\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$-strongly convex on $\mathcal{U}$, with $\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}) > 0$, then, for $\boldsymbol{\Pi}_J = \mathbf{H}_J \mathbf{H}_J^{+}$ and $\rho(J) = \|(\mathbf{I} - \boldsymbol{\Pi}_J)\boldsymbol{g}_{\boldsymbol{w}}\|_2$, the evaluation error is upper-bounded as*

$$\big|\mathrm{Err}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}})\big| \le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})}\left[\frac{C_{\mathbf{H}}}{2}\|\mathbf{H}_J^{+}\boldsymbol{g}_{\boldsymbol{w}}\|_2^2 + \rho(J)\right] \le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})}\left[\frac{C_{\mathbf{H}}}{2}\frac{\|\boldsymbol{g}_{\boldsymbol{w}}\|_2^2}{\sigma_{\min}(\mathbf{H}_J)^2} + \rho(J)\right]. \tag{14}$$

The bound in **Theorem 1** is governed by $\sigma_{\min}(\mathbf{H}_J)$, the smallest singular value of the Hessian block restricted to the selected parameter subset J, and the projection residual $\rho(J)$. The residual measures the component of the target gradient that cannot be represented by the selected Hessian columns. Here the reference-point quantity $\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$ is assumed to lower-bound the Hessian eigenvalues throughout $\mathcal{U}$, so the guarantee applies locally to the reference, edited, and retrained models in this neighborhood. $\mathbf{H}_J^{\top}\mathbf{H}_J$ is a principal submatrix of the full Gram matrix $\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}^{\top}\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}$, up to a coordinate reordering. By the Cauchy interlacing theorem [21],

$$\sigma_{\min}^2(\mathbf{H}_J) = \lambda_{\min}(\mathbf{H}_J^{\top}\mathbf{H}_J) \ge \lambda_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}^{\top}\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}) = \sigma_{\min}^2(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}), \tag{15}$$

and hence $\sigma_{\min}(\mathbf{H}_J) \ge \sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$. This inequality controls the curvature-dependent term in (14), while a small $\rho(J)$ preserves the target-gradient component needed for the edit.

When all parameters are editable, $\boldsymbol{\Pi}_J = \mathbf{I}$ and $\mathbf{H}_J^{+} = \mathbf{H}_{\mathbf{1}-\boldsymbol{w}}^{-1}$. Provided that the full-parameter edited model also lies in $\mathcal{U}$, the corresponding bound is

$$\big|\mathrm{Err}_{\mathrm{IF}}(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}})\big| \le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})}\frac{C_{\mathbf{H}}}{2}\|\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}}\|_2^2. \tag{16}$$

Here, $\mathrm{Err}_{\mathrm{IF}}$ denotes the evaluation error when all parameters are editable. Since $C_f/\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$ is common to both bounds, the GIF bound is no larger whenever

$$\frac{C_{\mathbf{H}}}{2}\|\mathbf{H}_J^{+}\boldsymbol{g}_{\boldsymbol{w}}\|_2^2 + \rho(J) \le \frac{C_{\mathbf{H}}}{2}\|\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}^{-1}\boldsymbol{g}_{\boldsymbol{w}}\|_2^2. \tag{17}$$

Consequently, GIF updates admit a tighter error bound when the reduction in the curvature-dependent term exceeds the projection residual.

We next provide a complementary analysis focused on the effect of the GIF update on the edited data themselves. While Theorem 1 bounds the discrepancy between the retrained model and the edited model under an arbitrary evaluation functional f, here we ask whether the GIF update moves the edited data in the desired direction.

Define the masked empirical risk $\mathcal{L}_{\boldsymbol{w}}(\boldsymbol{\theta}) := \frac{1}{m}\sum_{i=1}^{m} w_i \ell(z_i, \boldsymbol{\theta})$. We define the target-risk response induced by the GIF update as

$$\Delta\mathcal{L}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) := \mathcal{L}_{\boldsymbol{w}}\big(\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})\big) - \mathcal{L}_{\boldsymbol{w}}(\hat{\boldsymbol{\theta}}). \tag{18}$$

$\Delta\mathcal{L}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}})$ compares the loss on the selected data before and after applying the GIF-based unlearning update. A positive value means that the selected data become harder for the edited model to fit, which is the desired direction in unlearning.

Theorem 2 (Response of the self-loss)**.** *Suppose that $\mathcal{L}_{\boldsymbol{w}}$ is locally convex and let $\boldsymbol{\Pi}_J = \mathbf{H}_J\mathbf{H}_J^+$. If $\boldsymbol{g}_{\boldsymbol{w}}$ and Δ are aligned, i.e., $\boldsymbol{g}_{\boldsymbol{w}}^\top\Delta \geq 0$, then, for some $c \in [0,1]$, we have*

$$\Delta\mathcal{L}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) \geq \left(1 - c + \frac{\sigma_{\min}(\mathbf{H}_{\boldsymbol{w}})}{2\sigma_{\max}(\mathbf{H}_J)^2}\right) \|\boldsymbol{\Pi}_J\boldsymbol{g}_{\boldsymbol{w}}\|_2^2. \tag{19}$$

Theorem 2 shows that effective forgetting requires a large projection response $\|\boldsymbol{\Pi}_J\boldsymbol{g}_{\boldsymbol{w}}\|_2^2$. Since $\boldsymbol{\Pi}_J = \mathbf{H}_J\mathbf{H}_J^+$ projects the target gradient onto the Hessian-column space induced by the selected parameters, this term measures how much of the target-gradient direction can be realized by updating only J. Thus, parameter selection should choose a subset that both avoids unstable curvature directions and captures a large target-gradient response.

Why full-parameter updates fail? **Theorem 1** shows that the curvature-dependent term in the error bound grows as the minimum singular value decreases. In deep neural networks, the full parameter space typically contains many flat directions with near-zero singular values. These directions can amplify small gradient perturbations into unstable updates, enlarging the full-parameter bound in (16). Restricting the update to a well-conditioned parameter subset J can avoid these flat directions and yield a tighter finite-step error bound when the reduction in the curvature-dependent term exceeds the projection residual, as in (17).

Moreover, **Theorem 2** provides another explanation for why full-parameter updates can be suboptimal. When GIFs are extended to the full-parameter IF case, Eq. (19) reduces to

$$\Delta\mathcal{L}_{\mathrm{IF}}(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) \geq \left(1 + \frac{\sigma_{\min}(\mathbf{H}_{\boldsymbol{w}})}{2\sigma_{\max}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})^2}\right) \|\boldsymbol{g}_{\boldsymbol{w}}\|_2^2. \tag{20}$$

However, in well-trained models, $\boldsymbol{g}_{\boldsymbol{w}}$ could be sparse, concentrated on a relatively small subset of parameter directions. In such cases, matching the full-parameter response is sufficient for J to cover the high-magnitude directions that carry most of the gradient [33, 44]. Consequently, one may have $\|\boldsymbol{\Pi}_J\boldsymbol{g}_{\boldsymbol{w}}\|_2^2 \approx \|\boldsymbol{g}_{\boldsymbol{w}}\|_2^2$ even when J contains only a small fraction of the parameters. When J preserves most of the target-gradient response, the comparison is mainly governed by the curvature denominator. Since Cauchy interlacing gives $\sigma_{\max}(\mathbf{H}_J) \leq \sigma_{\max}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$, the full-parameter update can increase this denominator without a commensurate gain in self-loss response. Thus, a well-chosen GIF update can yield a stronger effective edit than the classical IF update.

3.2 Identifying Suitable Target Parameters

Accurate GIF updates require an editable subset $\boldsymbol{\theta}_J$ that captures the desirable direction while avoiding unnecessary changes to parameters that support retained behavior. **Theorems 1** and **2** show that the parameter subspace should avoid flat curvature directions, while preserving the target-gradient response. Therefore, we design scoring rules based on these two requirements.

Since the curvature often has strong interactions among nearby parameters in the computational graph, we use a blockwise-sparse curvature approximation and evaluate parameters blockwise [17]. Let $B \subset \{1, \ldots, n\}$ be a candidate parameter block, and score how suitable each block is for the editable subset. The most direct criterion is curvature-aware parameter selection (CAPS), defined by

$$S(B) := \boldsymbol{g}_{\boldsymbol{w},B}^\top(\mathbf{C}_B + \lambda\mathbf{I})^{-1}\boldsymbol{g}_{\boldsymbol{w},B}, \tag{21}$$

where $\boldsymbol{g}_{\boldsymbol{w},B}$ is the gradient restricted to B and $\mathbf{C}_B$ is a local curvature proxy. Motivated by the min-residual structure of GIFs (9), CAPS favors blocks with large target-gradient while penalizing low local curvature. Appendix I derives why CAPS suggests a well-conditioned parameter subspace.

We also consider two lighter selection criteria, Top-$\rho\%$ gradients and Top-$\rho\%$ outputs. For these criteria, each block consists of the parameters that produce one coordinate or channel of a layer output tensor. Top-$\rho\%$ gradients assigns a score to each block using the magnitude of the backward-pass gradient at the corresponding output unit on the target data. It then selects the highest-scoring blocks until they cover $\rho\%$ of the layer's trainable parameters. Top-$\rho\%$ outputs uses the same block construction, but scores each block using the magnitude of the forward-pass activation at the corresponding output unit on the target data. Thus, Top-$\rho\%$ gradients follows the target loss more directly, whereas Top-$\rho\%$ outputs provides a cheaper forward-pass proxy. These criteria are related to pruning saliency methods for selecting relevant parameters [9, 10, 45], but here they define the editable subspace rather than compressing the network.

4 Pseudo-LiSSA for Inverse Sub-Hessian-Vector Product

We introduce p-LiSSA, which converges to the inverse sub-Hessian vector product, i.e., $\mathbf{H}_J^+\mathbf{g}$. For any $\mu > 0$, we define p-LiSSA by the following iteration,

$$\mathcal{I}_k = \mathcal{I}_0 + (\mathbf{I} - \mu\mathbf{H}_J^\top\mathbf{H}_J)\mathcal{I}_{k-1}, \quad \mathcal{I}_0 = \mu\mathbf{H}_J^\top\boldsymbol{g}_{\boldsymbol{w}}. \tag{22}$$

If $\rho(\mathbf{I} - \mu\mathbf{H}_J^\top\mathbf{H}_J) < 1$, $\mathcal{I}_k$ converges to $\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}} := \mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J|\hat{\boldsymbol{\theta}}, \epsilon\big)$. The theorem below shows that p-LiSSA converges to iHVP without auxiliary regularization. This contributes to reducing the error of IFs by facilitating more accurate iHVP computation.

Theorem 3 (Convergence of p-LiSSA)**.** *Under the above definition,*
i) *p-LiSSA is scale-invariant, i.e., scalar multiplication on $\ell(\cdot)$ does not change $\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J|\hat{\boldsymbol{\theta}}, \epsilon\big)$.*
ii) *If $0 < \mu < 2/\lambda_{\max}(\mathbf{H}_J^\top\mathbf{H}_J)$, then p-LiSSA converges to $\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}}$.*
iii) *If $\mu = \frac{2}{\lambda_{\max}(\mathbf{H}_J^\top\mathbf{H}_J)+\lambda_{\min}(\mathbf{H}_J^\top\mathbf{H}_J)}$, the convergence rate of p-LiSSA is given by $\frac{\kappa-1}{\kappa+1}$, where $\kappa = \frac{\lambda_{\max}(\mathbf{H}_J^\top\mathbf{H}_J)}{\lambda_{\min}(\mathbf{H}_J^\top\mathbf{H}_J)}$.*

Theorem 3 holds as $\mathbf{H}_J^\top\mathbf{H}_J$ is positive definite under the definition of p-LiSSA, and the eigenvalues of $\mathbf{I} - \mu\mathbf{H}_J^\top\mathbf{H}_J$ are $1 - \mu\lambda_i$; hence $\rho(\mathbf{I} - \mu\mathbf{H}_J^\top\mathbf{H}_J) < 1$ for $0 < \mu < 2/\lambda_{\max}(\mathbf{H}_J^\top\mathbf{H}_J)$. In practice, we can adaptively decrease μ whenever p-LiSSA diverges rather than explicitly computing $\sigma_{\max}(\mathbf{H}_J^\top\mathbf{H}_J)$, which is a standard damping strategy in numerical optimization, such as the Levenberg-Marquardt algorithm [49] and classical backtracking line search [8]. Furthermore, we show that the convergence rate is governed by the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ of $\mathbf{H}_J^\top\mathbf{H}_J$. Since low-curvature directions correspond to small eigenvalues of $\mathbf{H}_J^\top\mathbf{H}_J$, including them drives $\lambda_{\min} \to 0$, causing $\big((\kappa-1)/(\kappa+1)\big)^k \to 1$ for any fixed k.

Numerical implementation. Note that $\mathbf{H}_J^\top\mathbf{H}_J\mathcal{I}_{k-1}$ in p-LiSSA cannot be directly computed by a single HVP. Instead, we can obtain $\mathbf{H}_J^\top\mathbf{H}_J\mathcal{I}_{k-1}$ by applying HVP twice and selecting the J entries from the result vector, where

$$\underbrace{\mathbf{H}\,\mathbf{H}\begin{bmatrix}\mathcal{I}_{k-1}\\ \mathbf{0}\end{bmatrix}}_{\text{First HVP}} = \underbrace{\begin{bmatrix}\mathbf{H}_J^\top\\ \mathbf{H}_{-J}^\top\end{bmatrix}\mathbf{H}_J\mathcal{I}_{k-1}}_{\text{Second HVP}} = \underbrace{\begin{bmatrix}\mathbf{H}_J^\top\mathbf{H}_J\mathcal{I}_{k-1}\\ \cancel{\mathbf{H}_{-J}^\top\mathbf{H}_J\mathcal{I}_{k-1}}\end{bmatrix}}_{\text{Dropping } \boldsymbol{\theta}_{-J} \text{ entries}}. \tag{23}$$

While LiSSA provides an unbiased estimate using single mini-batches per iteration [1], applying the same batch to both the first and the second HVP in (23) leads to a biased estimator, as $\mathbb{E}[\mathbf{H}_{J,B}^\top\mathbf{H}_{J,B}] \neq \mathbb{E}[\mathbf{H}_{J,B}]^\top\mathbb{E}[\mathbf{H}_{J,B}]$. Thus, we adopt two independent mini-batches per iteration, which the following theorem shows is sufficient to guarantee that p-LiSSA is an unbiased estimator of GIFs.

Theorem 4 (Unbiasedness of p-LiSSA)**.** *Assume that the batches $B_{k,1}, B_{k,2}$ are drawn independently to compute $\mathbf{H}_{J,k,1}$ and $\mathbf{H}_{J,k,2}$. Define the stochastic recurrence as*

$$\tilde{\mathcal{I}}_k = \tilde{\mathcal{I}}_0 + \big(\mathbf{I} - \mathbf{H}_{J,k,1}^\top\mathbf{H}_{J,k,2}\big)\tilde{\mathcal{I}}_{k-1},\ \tilde{\mathcal{I}}_0 = \mathbf{H}_{J,0,1}^\top\boldsymbol{g}_{\boldsymbol{w}}.$$

Then, for any $k \geq 0$, the expected value of $\tilde{\mathcal{I}}_k$ satisfies $\mathbb{E}[\tilde{\mathcal{I}}_k] = \sum_{j=0}^{k}\big(\mathbf{I} - \mathbf{H}_J^\top\mathbf{H}_J\big)^j\mathbf{H}_J^\top\boldsymbol{g}_{\boldsymbol{w}}$. Consequently, as $k \to \infty$, $\mathbb{E}[\tilde{\mathcal{I}}_k]$ converges to $\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}}$.

Taken together, Alg. 1 provides the complete computational procedure for GIFs.

5 Numerical Experiments

We organize our experiments around the three main claims of this paper. First, we evaluate whether GIFs improve the model-editing trade-off in class-level unlearning, where the edited model should forget the target data while preserving performance on the retained data. Second, we study the role of parameter selection by varying the editable parameter subset while keeping the influence update rule fixed. Third, we test whether p-LiSSA provides a more stable numerical approximation to the iHVP than existing iHVP solvers. Together, these experiments test our claim that classical IFs become inaccurate as full-parameter updates move irrelevant or low-curvature directions, and as the required iHVP computation is numerically unstable. Implementation and reproducibility details are provided in Appendix J.

Algorithm 1. Generalized Influence Functions

Input: $\hat{\boldsymbol{\theta}}$, $\boldsymbol{w} \in \{0,1\}^m$, μ.
Determine J via the selection methods.
$\mathcal{I}_0 \leftarrow \mu\,\mathbf{H}_J^\top \boldsymbol{g}_{\boldsymbol{w}}$, $k \leftarrow 0$.
repeat
  # Using two mini-batches
  $\mathcal{I}_{k+1} \leftarrow \mu\,\mathbf{H}_J^\top \boldsymbol{g}_{\boldsymbol{w}} + (\mathbf{I} - \mu\,\mathbf{H}_J^\top \mathbf{H}_J)\mathcal{I}_k$.
  if $\mathcal{I}_{k+1}$ is diverging **then**
    $\mu \leftarrow \mu/2$, $\mathcal{I}_0 \leftarrow \mu\,\mathbf{H}_J^\top \boldsymbol{g}_{\boldsymbol{w}}$.
    Restart the loop with $k \leftarrow 0$.
  end if
  $k \leftarrow k + 1$.
until convergence

5.1 GIFs Improve Model Edits

Table 1: Data-removal results for full-parameter influence-function, restricted, and trajectory-aware baselines. *Retain* is the accuracy on non-target examples and should be high; *Unlearn* is the accuracy on target examples and should be low.

Method	MNIST FCN		CIFAR-10 ResNet-18		SVHN VGG-11		Newsgroups BERT-L2-H128		PubMed 20k RCT BERT-L2-H128	
	Retain (↑)	Unlearn (↓)	Retain (↑)	Unlearn (↓)	Retain (↑)	Unlearn (↓)	Retain (↑)	Unlearn (↓)	Retain (↑)	Unlearn (↓)
Retrain	98.09 ± 0.00	0.00 ± 0.00	85.86 ± 0.00	0.00 ± 0.00	95.51 ± 0.00	0.00 ± 0.00	65.52 ± 0.00	0.00 ± 0.00	87.64 ± 0.00	0.00 ± 0.00
Classic IF [25]	93.38 ± 2.02	1.33 ± 1.47	84.50 ± 0.11	0.58 ± 0.14	93.94 ± 0.17	0.11 ± 0.12	60.06 ± 0.50	5.83 ± 0.61	86.57 ± 0.09	2.51 ± 0.04
Freezing	86.79 ± 7.64	4.04 ± 4.84	79.99 ± 11.90	1.35 ± 2.64	93.76 ± 0.00	0.00 ± 0.00	62.50 ± 0.02	14.73 ± 0.00	83.65 ± 0.30	3.67 ± 0.23
Second-order IF [5]	93.66 ± 2.55	10.90 ± 30.97	84.54 ± 0.09	0.60 ± 0.13	93.92 ± 0.16	0.15 ± 0.16	60.04 ± 0.58	5.92 ± 0.79	86.70 ± 0.20	2.53 ± 0.05
TracIn [38]	94.48 ± 0.65	0.68 ± 0.61	83.08 ± 0.00	1.00 ± 0.00	94.36 ± 0.00	0.00 ± 0.00	60.65 ± 0.00	11.60 ± 0.00	83.81 ± 0.46	2.19 ± 0.12
Diff-In [47]	81.03 ± 6.90	4.84 ± 2.88	77.86 ± 3.18	4.00 ± 2.96	89.79 ± 1.15	1.08 ± 0.80	55.53 ± 0.38	9.89 ± 0.51	76.73 ± 0.99	1.41 ± 0.16
HyperInf [56]	92.53 ± 2.92	1.81 ± 1.44	83.44 ± 0.19	0.57 ± 0.50	94.29 ± 0.12	0.06 ± 0.11	59.52 ± 0.36	8.06 ± 0.47	86.34 ± 0.27	3.43 ± 0.31
DataInf [28]	91.12 ± 2.17	2.62 ± 1.17	85.31 ± 0.01	0.30 ± 0.00	94.68 ± 0.00	0.00 ± 0.00	63.33 ± 0.00	3.76 ± 0.00	85.15 ± 0.39	1.69 ± 0.25
Schioppa [40]	94.82 ± 2.04	0.56 ± 0.62	60.27 ± 0.56	11.01 ± 0.70	93.13 ± 0.00	0.00 ± 0.00	63.05 ± 1.30	2.73 ± 1.65	85.40 ± 1.24	2.98 ± 0.71
EKFAC [17, 34]	91.45 ± 2.53	1.30 ± 0.83	83.62 ± 0.00	0.50 ± 0.00	94.62 ± 0.00	0.00 ± 0.00	61.69 ± 0.00	4.39 ± 0.00	85.55 ± 0.26	1.614 ± 0.06
GIF (Ours)	**94.95** ± 2.33	**0.42** ± 0.38	**85.87** ± 0.04	**0.10** ± 0.00	**95.14** ± 0.03	**0.00** ± 0.00	**65.45** ± 0.05	**0.00** ± 0.00	**87.20** ± 0.21	**1.610** ± 0.31

We first show that GIFs can produce better model edits in class-level data removal tasks than the IF baselines across multiple datasets and model architectures. In the class-removal task, each method edits a trained model to remove the behavior associated with a target class. The edited model is evaluated on two disjoint metrics. *Retain* is the accuracy on non-target examples and measures whether unrelated behavior is preserved. *Unlearn* is the accuracy on target examples and measures how much target behavior remains after editing. We adopt the model trained from scratch without the target class as the reference, since IF and GIF are derived to approximate the model change induced by retraining under the corresponding edited objective.

We compare GIFs using a 10% target-relevant parameter subset against full-parameter IF baselines, namely classical IF [25] and second-order IF [5], and with recent influence-estimation schemes including TracIn [38], HyperInf [56], DataInf [28], Schioppa [40], and EKFAC [17, 34]. We also include Freezing[3], which solves the restricted ERM update in (25), and the trajectory-aware Diff-In estimator [47].

Table 1 shows that GIF achieves the highest retain accuracy among the editing methods on all five benchmarks, while reducing unlearn accuracy to 0.00–1.61%. Its retain accuracy is within 0.44 percentage points of Retrain on four benchmarks, with a larger 3.14-point gap on MNIST. Compared with the newly added baselines, GIF improves retain accuracy by 1.38–8.16 points over Freezing and

[3]Appendix C explains how the Freezing update differs from GIF's full-gradient residual update.

5.35–13.92 points over Diff-In. GIF achieves lower unlearn accuracy than Diff-In on four datasets, while its unlearn accuracy is 0.20 percentage points higher on PubMed.

The error-bound diagnostics in Appendix E further show that selectors with positive measured bound gap $G(J)$ retain accuracy better than deliberately poor selectors, and that low target-gradient capture coincides with the largest gap from retraining on MNIST.

5.2 Selecting Relevant Parameters Preserves Retained Behavior

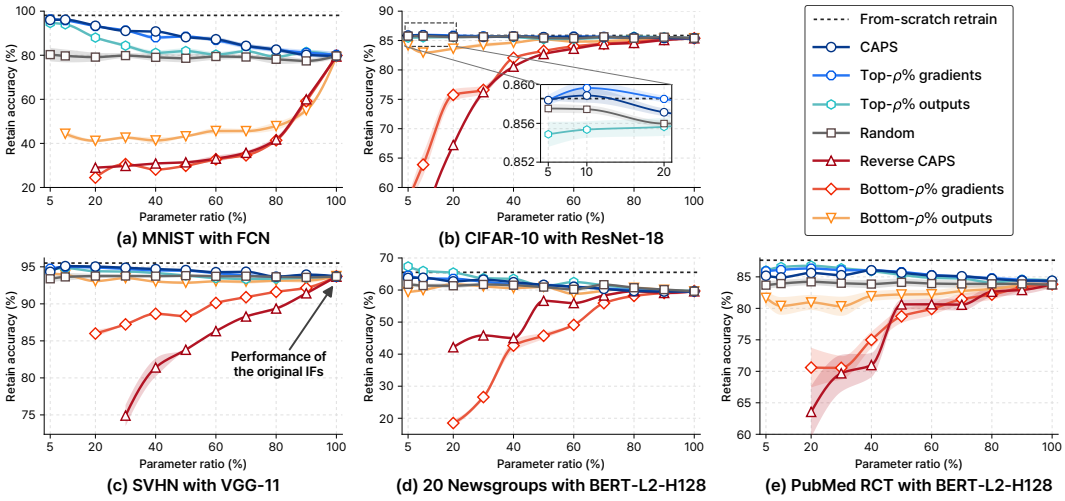


Figure 3: Retained accuracy of parameter selection schemes.

We next analyze how parameter selection affects model editing. In this experiment, we keep the influence update rule fixed and vary only the selection method. We compare the selectors against their counterparts, including Reverse CAPS, Bottom-$\rho\%$ gradients, and Bottom-$\rho\%$ outputs, as well as Random, which selects $\rho\%$ of parameters uniformly at random. We measure the retained accuracy after the edit on the same target-class removal task.

Figure 3 shows that CAPS and Top-ρ gradients closely track the from-scratch retrained reference at the 5% and 10% parameter ratios, achieving nearly identical and stable performance. Top-ρ outputs performs best on the Newsgroups dataset, but its performance is less stable across architectures and datasets. In contrast, Reverse CAPS and Bottom-ρ gradients tend to select low-relevance or poorly conditioned directions, and their performance degrades more severely as the parameter budget decreases. At the 5% and 10% ratios, these selectors can even fail to produce class-removal edits. Interestingly, Bottom-ρ outputs does not show the same severe drop in retained accuracy. This is because a block with low activation on the target data can still have parameters with sufficient gradient signal for the target loss [22, 35].

Compared with the full-parameter setting, which corresponds to classical IFs, selecting a relevant parameter subset improves retained accuracy by 0.58 percentage points on CIFAR-10, 16.53 points on MNIST, 7.72 points on Newsgroups, 2.27 points on PubMed 20k RCT, and 1.40 points on SVHN. These gains show that GIFs can outperform IFs not by increasing the update dimension, but by restricting the edit to parameters that better align with the target behavior while avoiding nuisance changes in retained behavior. Overall, these results empirically support **Theorems 1 and 2** and show that selective model editing is effective when the editable subset is properly chosen.

5.3 Efficiency and Stability of iHVP Solvers

We compare p-LiSSA with standard iHVP solvers, including LiSSA [25], conjugate gradient, Schulz iteration [56], Lanczos iteration [40], KFAC, and EKFAC [17], across models ranging from 5,000 to 20 million parameters. For a solver output $\hat{x}$, we measure the relative residual $r(\hat{x}) = \|\mathbf{H}\hat{x} - \boldsymbol{g}\|_2/\|\boldsymbol{g}\|_2$. Only for p-LiSSA with 10% parameters, we define the residual as $r(\hat{x}) = \|\mathbf{H}_J^{\mathsf{T}}\mathbf{H}_J\hat{x} -$

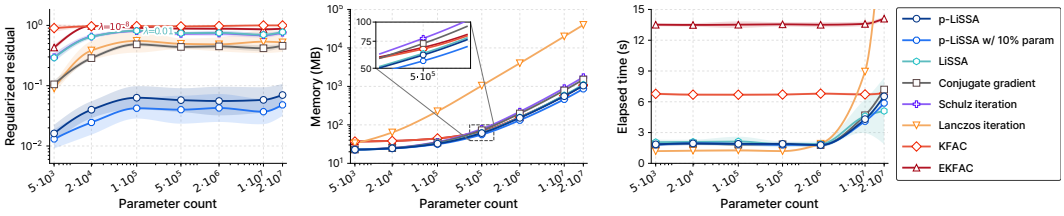


Figure 4: Comparison of iHVP solvers across model scales.

$\mathbf{H}_J^\top \boldsymbol{g}\|_2 / \|\boldsymbol{g}\|_2$. This measures how accurately the computed update satisfies the iHVP linear system, with smaller values indicating a direction closer to the exact inverse-Hessian solution.

Figure 4 shows that p-LiSSA gives the best overall solver trade-off. At all scales, p-LiSSA obtains an average residual of 0.039. The best baseline residual is 0.379 from conjugate gradient, so p-LiSSA reduces the residual by 9.6×. It also has the lowest average memory use, 227.5 MB compared with 283.0 MB for EKFAC, the most memory-efficient baseline. The average runtime of p-LiSSA is 2.75 seconds, slightly faster than LiSSA at 2.82 seconds and lower than that of the other solvers. The same pattern holds at the largest scale. p-LiSSA with 10% parameters attains the lowest residual, 0.052, using 864 MB and 5.87 seconds. LiSSA is slightly faster at 5.10 seconds but has a much larger residual of 0.777, while conjugate gradient reaches only 0.460 residual with 1.51 GB memory.

6 Discussion

Parameter selection introduces a trade-off between conditioning and target-gradient coverage. Excluding low-curvature directions can improve conditioning, but may also discard components needed to approximate the target response, increasing the projection residual. Effective selection therefore requires considering both quantities together. For large edits, curvature can also change substantially as the model moves away from the reference point, motivating further study of how parameter selection should adapt over the course of an edit.

A remaining computational limitation is that p-LiSSA, although it provides a damping-free restricted solve, still relies on second-order computation. In particular, computing $H_J^\top H_J I_k$ requires two HVPs, which can become a bottleneck at larger scales. Developing more efficient restricted curvature approximations and scalable solvers is therefore an important direction for future work.

Another promising direction is to extend GIFs to model editing in generative models. At foundation scale, repeated HVPs and restricted solves can themselves become a bottleneck. Moreover, increasing a training loss does not by itself guarantee semantic forgetting. Evaluation for diffusion models should examine changes in the generated distribution or score field. For LLMs, it should test whether the targeted information remains accessible through paraphrases or alternative reasoning paths. Addressing these computational and evaluation challenges would extend the scope of GIFs and help advance efficient and reliable model editing for generative models.

Acknowledgments

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25523626, Development of End-to-End Service Performance Guarantee Framework Technology for 6G), in part by the IITP grant funded by the Korea government (MSIT) under the Information Technology Research Center (ITRC) Support Program (No. IITP-2026-2021-0-02048), and in part by the IITP grant funded by the Korea government (MSIT) under the AI Star Fellowship Support Program (No. IITP-2026-RS-2026-25616664).

References

[1] Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18(116):1–40, 2017.

[2] Ahmed Alaa and Mihaela Van Der Schaar. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 165–174. PMLR, 13–18 Jul 2020.

[3] Juhan Bae, Nathan Hoyen Ng, Alston Lo, Marzyeh Ghassemi, and Roger Baker Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, 2022.

[4] S Basu, Phillip E Pope, and S Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021.

[5] Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences of the United States of America*, 117(48):30071–30078, 2020.

[7] Adi Ben-Israel and Thomas N. E. Greville. *Generalized Inverses: Theory and Applications*, volume 15. Springer, 2003.

[8] Dimitri Bertsekas. *Nonlinear Programming*, volume 4. Athena Scientific, 2016.

[9] Shi Chen and Qi Zhao. Shallowing deep networks: Layer-wise pruning based on feature representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3048–3056, 2019.

[10] Rocktim Jyoti Das, Mingjie Sun, Liqun Ma, and Zhiqiang Shen. Beyond size: How gradients shape pruning decisions in large language models, 2024.

[11] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[12] Franck Dernoncourt and Ji Young Lee. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In Greg Kondrak and Taro Watanabe, editors, *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, June 2019.

[14] Jacob R. Epifano, Ravi P. Ramachandran, Aaron J. Masino, and Ghulam Rasool. Revisiting the fragility of influence functions. *Neural Networks*, 162:581–588, 2023.

[15] Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A swiss army infinitesimal jackknife. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1139–1147. PMLR, 16–18 Apr 2019.

[16] Xiuwen Gong, Dong Yuan, and Wei Bao. Partial label learning via label influence function. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7665–7678. PMLR, 17–23 Jul 2022.

[17] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying large language model generalization with influence functions, 2023.

[18] Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

[19] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data Cleansing for Models Trained with SGD. In *Advances in Neural Information Processing Systems*, 2019.

[20] Dat Minh Hong, Bruno Kacper Mlodozeniec, Runa Eschenhagen, and Richard E. Turner. Better hessians matter: Studying the impact of curvature approximations in influence functions. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.

[21] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, USA, 2nd edition, 2012.

[22] Nannan Huang, Haytham M. Fayek, and Xiuzhen Zhang. Less is more? examining fairness in pruned large language models for summarising opinions. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17994–18018, Suzhou, China, November 2025. Association for Computational Linguistics.

[23] Yegor Klochkov and Yang Liu. Revisiting inverse hessian vector products for calculating influence functions, 2024.

[24] Pang Wei Koh, Kai-Siang Ang, Hubert H K Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, pages 5254–5264, December 2019.

[25] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.

[26] Shuming Kong, Yanyan Shen, and Linpeng Huang. Resolving training biases via influence-based data relabeling. In *International Conference on Learning Representations*, 2022.

[27] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[28] Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *International Conference on Learning Representations*, 2024.

[29] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

[30] Ken Lang. Newsweeder: learning to filter netnews. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ICML'95, pages 331–339, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Zhe Li, Wei Zhao, Yige Li, and Jun Sun. Do influence functions work on large language models? In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14367–14382, Suzhou, China, November 2025. Association for Computational Linguistics.

[33] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

[34] James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417, Lille, France, 07–09 Jul 2015. PMLR.

[35] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[37] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks.

[38] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, pages 19920–19930, 2020.

[39] Ittai Rubinstein and Samuel B. Hopkins. Rescaled influence functions: Accurate data attribution in high dimension. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[40] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 8179–8186, 2022.

[41] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE/CVF International Conference on Computer Vision*, pages 618–626, 2017.

[42] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. In *International Conference on Learning Representations*, 2020.

[43] G. W. Stewart. *Matrix Algorithms*. Society for Industrial and Applied Mathematics, 1998.

[44] Xu Sun, Xuancheng Ren, Shuming Ma, and Houfeng Wang. meProp: Sparsified back propagation for accelerated deep learning with reduced overfitting. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3299–3308. PMLR, 06–11 Aug 2017.

[45] Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. Ecoflap: Efficient coarse-to-fine layer-wise pruning for vision-language models. In *International Conference on Learning Representations*, 2024.

[46] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data Pruning via Moving-one-Sample-out. In *Advances in Neural Information Processing Systems*, 2023.

[47] Haoru Tan, Sitong Wu, Xiuzhe Wu, Wang Wang, Bo Zhao, Zeke Xie, Gui-Song Xia, and Xiaojuan Qi. Understanding Data Influence with Differential Approximation. *arXiv preprint arXiv:2508.14648*, 2025.

[48] Haoru Tan, Xiuzhe Wu, Sitong Wu, Shaofeng Zhang, Yanfeng Chen, Xingwu Sun, Jeanne Shen, and Xiaojuan Qi. Understanding Data Influence in Reinforcement Finetuning. In *Advances in Neural Information Processing Systems*, 2025.

[49] Mark K. Transtrum, Benjamin B. Machta, and James P. Sethna. Geometry of nonlinear least squares with applications to sloppy models and optimization. *Physical Review E*, 83:036701, Mar 2011.

[50] Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the Temporal Dependence of Training Data Influence. In *International Conference on Learning Representations*, 2025.

[51] Ga Wu, Masoud Hashemi, and Christopher Srinivasa. PUMA: Performance unchanged model augmentation for training data removal. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8675–8682, Jun. 2022.

[52] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. GIF: A general graph unlearning strategy via influence function. In *Proceedings of the ACM Web Conference*, page 651–661, New York, NY, USA, 2023. Association for Computing Machinery.

[53] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *International Conference on Learning Representations*, 2023.

[54] Xichen Ye, Yifan Wu, Weizhong Zhang, Cheng Jin, and Yifan Chen. Towards robust influence functions with flat validation minima. In *Forty-second International Conference on Machine Learning*, 2025.

[55] Youngsik Yoon, Jinhwan Nam, Hyojeong Yun, Jaeho Lee, Dongwoo Kim, and Jungseul Ok. Few-shot unlearning. In *IEEE Symposium on Security and Privacy*, pages 3276–3292, Los Alamitos, CA, USA, May 2024. IEEE Computer Society.

[56] Xinyu Zhou, Simin Fan, and Martin Jaggi. HyperINF: Unleashing the hyperpower of the schulz's method for data influence estimation, 2025.

A Related Works

Table 2: Comparison of representative influence-function methods. The methods are organized by problem setting, IF variant, assumptions, estimator, computational profile, empirical behavior, and limitations.

Method	Problem setting	IF variant	Main assumptions	Estimator / computation	Cost profile	Empirical behavior	Limitations
Classical and higher-order influence-function variants							
Classical IF [25]	Single-example attribution for test loss or prediction	First-order IF	Local twice differentiability; invertible or damped Hessian; infinitesimal reweighting	IHVP with HVP-based iterative solvers	Requires one or more IHVP solves per query	Effective for convex models and some small neural networks	Unstable in deep non-convex models; updates the full parameter space
Group IF [24]	Group removal and leave-k-out approximation	First-order group IF	Small-perturbation approximation; group structure preserves useful ranking	Sum of pointwise IFs or group-specific influence computation	Efficient when additive; self-loss requires additional computation	Can preserve ranking even when absolute error is large	Large coherent groups can induce substantial approximation error
Second-order group IF [5]	Selecting influential groups for black-box predictions	Second-order IF	Second-order Taylor approximation is informative; higher-order derivatives or approximations are tractable	First-order term plus second-order correction	More expensive than first-order IF	Improves correlation in linear models and helps in some neural-network settings	Still sensitive in deep non-convex regimes; higher computation cost
Higher-order IF / DJ [2]	Uncertainty quantification in deep regression	Higher-order IF	Von Mises expansion; repeated higher-order gradient computation is feasible	Higher-order IF approximation to leave-one-out behavior	High computational cost	Competitive discriminative intervals compared with Bayesian alternatives	Focuses on uncertainty quantification rather than direct data attribution or editing
Scalable influence estimators for modern models							
TracIn [38]	Large-scale data attribution	Checkpoint-based first-order surrogate	Training checkpoints provide useful trajectory information	Cumulative gradient inner products over checkpoints	Linear in the number of checkpoints; no Hessian inverse	Highly practical and scalable	Weak direct connection to retraining or curvature-corrected Newton dynamics
Scaling up IF [40]	Influence estimation for large transformer models	Arnoldi-based low-rank IHVP	Low-dimensional Krylov approximation is sufficient	Arnoldi basis with low-rank inverse approximation	High upfront basis construction cost; efficient reuse afterward	Scales to large language and vision transformers	Approximation quality depends on rank and Hessian spectrum
DataInf [28]	Data influence in LoRA-tuned LLMs and diffusion models	Closed-form approximate IF	Low effective dimensionality induced by PEFT / LoRA	Closed-form influence estimator	Low memory and computation	Fast and accurate in PEFT settings	Less directly applicable to full fine-tuning
Robust and structured influence estimators							
HyperINF [56]	Influence estimation for LoRA-tuned LLMs and VLMs	Schulz-type hyperpower IF	Low-rank GFIM or Hessian surrogate is accurate	Schulz iteration with structured curvature approximation	Nearly independent of LoRA rank in favorable settings	Improves accuracy, stability, and memory efficiency	Depends on surrogate quality and PEFT structure

This paper focuses on improving inaccurate IF estimates for model editing. We organize the review into five areas covering the statistical foundations and higher-order extensions of IFs, their fragility in deep non-convex models, scalable IHVP and curvature approximations, applications to editing, unlearning, and relabeling, and trajectory-aware data influence.

Statistical foundations and higher-order influence estimates. IFs originated in robust statistics as a local sensitivity measure of a statistical functional under infinitesimal contamination [18]. In modern machine learning, IFs were reintroduced for empirical risk minimization, where the effect of upweighting or removing a training example on a test loss can be approximated through an inverse-Hessian-vector product [25]. This formulation typically relies on local twice differentiability, invertibility or appropriate damping of the Hessian, and a small-perturbation approximation.

Subsequent work has extended the classical first-order formulation to larger perturbations and more structured data edits. Group effects were analyzed to show that, although absolute errors may grow for larger groups, influence estimates can still preserve useful ranking information [24]. Finite-sample guarantees were also developed for infinitesimal jackknife approximations to leave-k-out retraining [15]. Higher-order variants further attempt to reduce linearization error. Second-order group influence functions were proposed for black-box prediction, and higher-order influence-function estimators were developed for uncertainty quantification in deep learning [2, 5].

Fragility of influence functions in deep models. Despite their theoretical appeal, classical IFs often become unreliable in deep non-convex settings. Prior work empirically showed that IF accuracy is highly sensitive to model depth, width, regularization, test-point choice, and stochastic IHVP estimation [4]. Other work decomposed the mismatch between IFs and retraining into multiple error sources and argued that practical IF implementations often approximate a proximal Bregman response function rather than exact leave-one-out retraining [3]. Further studies revisited the fragility of IFs and showed that some reported failures can depend on the evaluation protocol [14].

These studies suggest that the difficulty is not only statistical but also geometric and numerical. Classical IFs compute an update in the full parameter space, where low-curvature or poorly conditioned directions can amplify small gradient components and induce parameter changes that are weakly

Table 3: Chronological development of influence-function research.

Year	Representative work	Main contribution
1974	Hampel [18]	Formalized the influence curve as a sensitivity measure under infinitesimal contamination.
2017	Agarwal et al. [1]; Koh and Liang [25]	Introduced LiSSA for stochastic second-order optimization and reintroduced IFs to modern machine learning through IHVP-based data attribution.
2019	Giordano et al. [15]; Koh et al. [24]	Developed finite-sample infinitesimal-jackknife guarantees and analyzed group effects for data removal.
2020	Basu et al. [5]; Alaa and van der Schaar [2]; Pruthi et al. [38]; Sinitsin et al. [42]	Extended IFs to second-order group effects and higher-order uncertainty estimation; proposed scalable gradient-tracing attribution and editable neural networks.
2021	Basu et al. [4]; De Cao et al. [11]	Empirically demonstrated IF fragility in deep learning and studied factual knowledge editing in language models.
2022	Bae et al. [3]; Schioppa et al. [40]; Wu et al. [51]; Kong et al. [26]	Analyzed the mismatch between IFs and retraining, scaled IFs with Arnoldi iteration, and applied influence-based ideas to data removal and relabeling.
2023	Epifano et al. [14]; Wu et al. [52]; Grosse et al. [17]	Revisited IF fragility, proposed influence-function-based graph unlearning, and used IFs to study large language model generalization.
2024	Kwon et al. [28]; Klochkov and Liu [23]	Proposed efficient influence estimation for LoRA-tuned models and revisited practical IHVP computation.
2025	Ye et al. [54]; Rubinstein and Hopkins [39]; Zhou et al. [56]; Li et al. [32]	Developed robust and rescaled IF variants, Schulz-type large-scale influence estimators, and analyzed the limitations of IFs in LLMs.

related to the edited data. This observation motivates our restricted formulation, which explicitly selects an editable parameter subspace based on data relevance and curvature stability.

Scalable IHVP and curvature approximations. A major computational bottleneck in IF estimation is the IHVP. LiSSA is a widely used iterative estimator based on a Neumann-series approximation, but it typically requires scaling and damping for numerical stability in modern neural networks [1, 25]. Several alternatives have been proposed to improve scalability or stability. Arnoldi iteration has been used to scale IFs to large transformer models [40]. EK-FAC-style curvature approximations have been employed for studying generalization in large language models [17]. Closed-form influence estimators have been proposed for LoRA-tuned large language models and diffusion models, and Schulz-type hyperpower iterations have been developed for efficient influence estimation in parameter-efficient fine-tuning settings [28, 56]. Recent work has also emphasized that the quality of IHVP and curvature approximations can dominate the final accuracy of influence estimates [20, 23].

Most of these methods aim to approximate the full-parameter IHVP more efficiently or more accurately. In particular, Schioppa et al. [40] approximate the conventional full-parameter IF in a dominant spectral/Krylov subspace, which can mitigate instability associated with poorly represented spectral directions. GIF similarly benefits from avoiding low-curvature directions, but defines a finite model-edit update on a target-dependent coordinate support. Its selection condition also requires target-gradient capture, quantified by the projection residual, rather than curvature alone. In contrast, our approach changes the object being solved. GIFs define a restricted inverse-sub-Hessian problem on an editable parameter subspace, and p-LiSSA solves this restricted problem without damping-induced bias. This distinction is important because the restricted problem can avoid irrelevant or ill-conditioned directions before the numerical IHVP computation is performed.

Influence functions for editing, unlearning, and relabeling. Beyond data attribution and debugging, IFs have been used in model editing and unlearning. TracIn traces gradient inner products along training checkpoints and provides a scalable first-order surrogate for data attribution, although it does not directly approximate Newton-style retraining dynamics [38]. Editable neural networks and KnowledgeEditor study local edits to model behavior [11, 42]. For data removal, influence-based updates have been proposed for performance-preserving training-data removal and graph unlearning [51, 52]. Influence estimates have also been used for relabeling and label correction [16, 26]. In large-scale

language-model settings, IFs have been used to analyze generalization and data influence [17, 28], but recent studies suggest that parameter-space changes, model-behavior changes, IHVP approximation errors, and fine-tuning convergence issues can cause standard IFs to fail in LLMs [32].

Trajectory-aware data influence. Several methods estimate sample attribution or utility by using intermediate checkpoints and the optimizer trajectory, including SGD retracing for data cleansing [19], moving-one-sample-out pruning [46], trajectory-specific influence embeddings [50], differential influence accumulation [47], and influence estimation for reinforcement fine-tuning [48]. These methods retain information about how individual examples interact with the evolving training state. GIF constructs one joint, class-level finite edit from the final checkpoint. Consequently, summing sample-wise trajectory influences is not generally equivalent to the class-level edit considered here; neither formulation removes the need for the other.

Overall, prior work has made substantial progress in theoretical influence estimation, scalable curvature approximation, and downstream applications. However, relatively less attention has been paid to finite model edits performed in a restricted, data-relevant, and numerically stable parameter subspace. GIFs and p-LiSSA address this gap by jointly formulating a restricted edit direction and a damping-free solver for the corresponding inverse-sub-Hessian problem.

B The Existence of Negative Eigenvalues

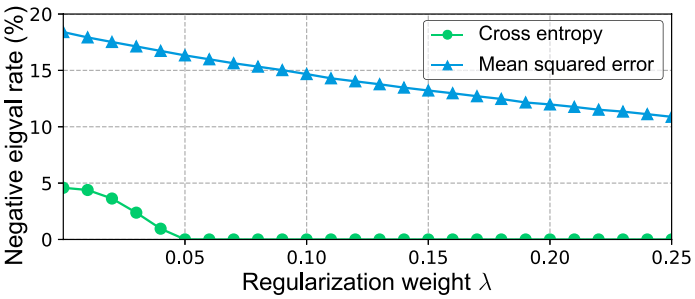


Figure 5: Negative eigenvalue ratio under various regularization weights λ for the MNIST classification on the fully-connected network.

Fig.5 shows the percentage of negative eigenvalues among the top 500 eigenvalues with the largest absolute values in the Hessian matrix, obtained using the Lanczos algorithm [29]. The negative eigenvalue rate for the cross-entropy loss implies that the Hessian matrix is positive semi-definite for large enough λ. However, the network might not be trained with a sufficiently large λ to ensure all eigenvalues are positive. Moreover, such λ may exist outside of the feasible range as in the case of the mean squared error.

C GIF and Freezing on Selected Parameters

GIF and parameter freezing answer different linearized problems when only the coordinates in J are editable. With fixed coordinates held constant, the linearized first-order condition of the restricted empirical-risk problem is

$$\mathbf{0} = \epsilon \boldsymbol{g}_{\boldsymbol{w},J} + \mathbf{H}_{JJ}\Delta_J^{\mathsf{Freeze}}, \tag{24}$$

and hence

$$\Delta_J^{\mathsf{Freeze}} = -\epsilon \mathbf{H}_{JJ}^{-1}\boldsymbol{g}_{\boldsymbol{w},J}. \tag{25}$$

Here, $\boldsymbol{g}_{\boldsymbol{w},J}$ and $\mathbf{H}_{JJ}$ are the target gradient and Hessian restricted to the editable coordinates. This is the usual freezing update, provided $\mathbf{H}_{JJ}$ is invertible. It approximates the solution of the restricted ERM problem to first order. An exact minimizer of the restricted ERM objective attains the smallest loss among all updates supported on the same J, including GIF.

GIF instead minimizes the residual of the full-model linearized condition after constraining the update to J. When $\mathbf{H}_J$ has full column rank, its update is

$$\Delta_J^{\mathsf{GIF}} = \operatorname*{argmin}_{\Delta_J} \|\epsilon \boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_J \Delta_J\|_2^2 = -\epsilon \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}. \tag{26}$$

Writing $\mathbf{H}_J = [\mathbf{H}_{JJ}^\mathsf{T} \mid \mathbf{H}_{-J,J}^\mathsf{T}]^\mathsf{T}$ makes the distinction explicit. GIF uses the full target gradient and includes the change induced in the fixed-coordinate gradients through $\mathbf{H}_{-J,J}$, whereas freezing uses only the editable-coordinate optimality condition. If $\mathbf{H}_{-J,J} = \mathbf{0}$, the two updates agree. These formulations target different criteria, and the experiments compare their resulting finite model edits.

D Proof of Theorem 1

Theorem 1 (Error bound of GIFs)**.** Let the functional $f(\cdot)$ and the Hessian map $\nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\cdot)$ be C_f- and $C_{\mathbf{H}}$-Lipschitz continuous on a convex neighborhood $\mathcal{U}$ containing $\hat{\boldsymbol{\theta}}$, $\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})$, and the retrained optimum $\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})$. Assume $\nabla \mathcal{L}_{\mathbf{1}}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$. If $\mathcal{L}_{\mathbf{1}-\boldsymbol{w}}$ is $\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$-strongly convex on $\mathcal{U}$, with $\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}) > 0$, then, for $\mathbf{\Pi}_J = \mathbf{H}_J \mathbf{H}_J^+$ and $\rho(J) = \|(\mathbf{I} - \mathbf{\Pi}_J)\boldsymbol{g}_{\boldsymbol{w}}\|_2$, the evaluation error is upper-bounded as

$$\left|\mathrm{Err}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}})\right| \le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \left[\frac{C_{\mathbf{H}}}{2} \|\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}\|_2^2 + \rho(J)\right] \le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \left[\frac{C_{\mathbf{H}}}{2} \frac{\|\boldsymbol{g}_{\boldsymbol{w}}\|_2^2}{\sigma_{\min}(\mathbf{H}_J)^2} + \rho(J)\right]. \tag{27}$$

This inequality for IFs has been earlier studied by [24], and we extend the error bound to GIFs. Define $\mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^m (1-w_i)\ell(z_i, \boldsymbol{\theta})$ and $\Delta = [\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}\ \mathbf{0}]^\mathsf{T}$ for brevity of notation. The local strong-convexity assumption means that $\nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\boldsymbol{\theta}) \succeq \sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})\mathbf{I}$ for every $\boldsymbol{\theta} \in \mathcal{U}$. Together with stationarity at the retrained optimum, it gives

$$\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}) \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})\|_2 \le \|\nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\boldsymbol{\theta})\|_2, \qquad \boldsymbol{\theta} \in \mathcal{U}. \tag{28}$$

This follows by integrating the Hessian along the segment from the retrained optimum to $\boldsymbol{\theta}$ and applying Cauchy–Schwarz to the resulting gradient inner product. The same neighborhood contains the segment $\hat{\boldsymbol{\theta}} + t\Delta$ for $t \in [0,1]$, on which the Hessian Lipschitz condition is used below. Because $\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}$ is evaluated at the reference model, we explicitly assume that its smallest singular value also lower-bounds curvature throughout $\mathcal{U}$. With these settings, the error is upper-bounded as

$$\left|\mathrm{Err}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}})\right| = \left|f\big(\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})\big) - f\big(\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})\big)\right| \le C_f \|\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w}) - \hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})\|_2 \tag{29}$$

$$\le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \Big\| \nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}\big(\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})\big) - \underbrace{\nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}\big(\hat{\boldsymbol{\theta}}(\mathbf{1}-\boldsymbol{w})\big)}_{=0 \text{ (from the first-order optimality)}} \Big\|_2 \tag{30}$$

$$= \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \Big\| \nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}\big(\underbrace{\hat{\boldsymbol{\theta}}^{\mathsf{GIF}}(\mathbf{1}-\boldsymbol{w})}_{=\hat{\boldsymbol{\theta}}+\Delta}\big) - \underbrace{\Big[\nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\hat{\boldsymbol{\theta}}) + \boldsymbol{g}_{\boldsymbol{w}}\Big]}_{=\nabla \mathcal{L}_{\mathbf{1}}(\hat{\boldsymbol{\theta}})=\mathbf{0}} \Big\|_2 \tag{31}$$

$$= \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \Big\| \underbrace{\nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}\big(\hat{\boldsymbol{\theta}} + \Delta\big) - \nabla \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}\big(\hat{\boldsymbol{\theta}}\big)}_{=\int_0^1 \nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\hat{\boldsymbol{\theta}}+t\Delta)\Delta \mathrm{d}t} - \underbrace{\mathbf{H}_J \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}}_{=\mathbf{H}_{\mathbf{1}-\boldsymbol{w}}\Delta} + \mathbf{\Pi}_J \boldsymbol{g}_{\boldsymbol{w}} - \boldsymbol{g}_{\boldsymbol{w}} \Big\|_2 \tag{32}$$

$$= \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \Big\| \int_0^1 \underbrace{\big[\nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\hat{\boldsymbol{\theta}} + t\Delta) - \nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\hat{\boldsymbol{\theta}})\big]}_{\text{operator-norm bounded by } C_{\mathbf{H}}\|\Delta\|_2 t} \Delta \mathrm{d}t + \mathbf{\Pi}_J \boldsymbol{g}_{\boldsymbol{w}} - \boldsymbol{g}_{\boldsymbol{w}} \Big\|_2 \tag{33}$$

$$\le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \left[\|\Delta\|_2 \int_0^1 \underbrace{\big\|\nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\hat{\boldsymbol{\theta}} + t\Delta) - \nabla^2 \mathcal{L}_{\mathbf{1}-\boldsymbol{w}}(\hat{\boldsymbol{\theta}})\big\|_2}_{\le C_{\mathbf{H}}\|\Delta\|_2 t \text{ (from the Lipschitz condition of } \mathbf{H})} \mathrm{d}t + \rho(J) \right] \tag{34}$$

$$\le \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \left[\frac{C_{\mathbf{H}}}{2}\|\Delta\|_2^2 + \rho(J)\right] = \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \left[\frac{C_{\mathbf{H}}}{2}\|\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}\|_2^2 + \rho(J)\right] \tag{35}$$

$$\leq \frac{C_f}{\sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})} \left[\frac{C_{\mathbf{H}}}{2} \frac{\|\boldsymbol{g}_{\boldsymbol{w}}\|_2^2}{\sigma_{\min}(\mathbf{H}_J)^2} + \rho(J) \right]. \tag{36}$$

The last inequality uses $\|\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}\|_2 \leq \|\boldsymbol{g}_{\boldsymbol{w}}\|_2 / \sigma_{\min}(\mathbf{H}_J)$. When all parameters are editable, $\mathbf{\Pi}_J = \mathbf{I}$ and $\rho(J) = 0$, recovering the full-parameter bound in (16) under the same local assumptions.

E Empirical Diagnostics for the Error Bound

We empirically evaluate the comparison condition in (17) on the MNIST model used for parameter selection. All selectors retain 10% of the parameters and values are averaged over 20 seeds. We use the empirical local estimate $\widehat{C}_{\mathbf{H}} = 130.47$, obtained from ten random perturbation directions of radius 0.01 with 15 power iterations. For clarity, define the common-factor-free bound gap

$$G(J) := \frac{\widehat{C}_{\mathbf{H}}}{2} \|\mathbf{H}^{-1} \boldsymbol{g}_{\boldsymbol{w}}\|_2^2 - \frac{\widehat{C}_{\mathbf{H}}}{2} \|\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}\|_2^2 - \rho(J). \tag{37}$$

Positive $G(J)$ means that the measured restricted bound term is smaller than the full-IF term. Both solves use conjugate gradient with regularization 10^{-4} and the same convergence criterion.

Table 4: MNIST error-bound diagnostics and retained accuracy. Curvature terms and $\rho(J)$ omit the common factor $C_f / \sigma_{\min}(\mathbf{H}_{\mathbf{1}-\boldsymbol{w}})$.

Selector	Full IF	Restricted	$\rho(J)$	$G(J)$	Retain (%)
CAPS	1.09484	1.08018	$4.84 \cdot 10^{-3}$	$9.82 \cdot 10^{-3}$	94.95
Top gradients	1.09484	1.08393	$4.86 \cdot 10^{-3}$	$6.05 \cdot 10^{-3}$	94.85
Top outputs	1.09484	1.08144	$4.87 \cdot 10^{-3}$	$8.52 \cdot 10^{-3}$	94.97
Random	1.09484	1.20962	$4.69 \cdot 10^{-3}$	$-1.19 \cdot 10^{-1}$	87.85
Bottom outputs	1.09484	1.14978	$5.06 \cdot 10^{-3}$	$-6.00 \cdot 10^{-2}$	57.18
Bottom gradients	1.09484	1.25156	$5.34 \cdot 10^{-3}$	$-1.62 \cdot 10^{-1}$	60.83
Reverse CAPS	1.09484	1.31670	$5.50 \cdot 10^{-3}$	$-2.27 \cdot 10^{-1}$	57.10

The well-selected subsets have $G(J) > 0$ and retained accuracy above the 93.38% classical IF baseline. Random and deliberately poor selectors have $G(J) < 0$ and lower retained accuracy, as shown in the table. This is an empirical consistency check and does not verify the local assumptions.

Table 5: Target-gradient capture and retraining fidelity for the GIF selector. Gap is Retrain minus GIF retain accuracy in percentage points.

Dataset	Retrain	GIF	Gap	Capture	Normalized residual
MNIST	98.09	94.95	3.14	66.45%	0.7473
CIFAR-10	85.86	85.87	-0.01	95.49%	0.2968
SVHN	95.51	95.14	0.37	84.98%	0.5271
20 Newsgroups	65.52	65.45	0.07	98.53%	0.1707
PubMed RCT-20k	87.64	87.20	0.44	99.42%	0.1073

Here capture denotes $\|\mathbf{\Pi}_J \boldsymbol{g}_{\boldsymbol{w}}\|_2 / \|\boldsymbol{g}_{\boldsymbol{w}}\|_2$ and normalized residual denotes $\rho(J) / \|\boldsymbol{g}_{\boldsymbol{w}}\|_2$. MNIST has the lowest capture and largest normalized residual, as well as the largest retain-accuracy gap from retraining. This supports the conditional interpretation that a selected subspace must capture the target-gradient direction as well as maintain stable restricted curvature.

F Proof of Theorem 2

Theorem 2 (Response of the self-loss)**.** Suppose that $\mathcal{L}_{\boldsymbol{w}}$ is locally convex and let $\mathbf{\Pi}_J = \mathbf{H}_J \mathbf{H}_J^+$. If $\boldsymbol{g}_{\boldsymbol{w}}$ and Δ are positively aligned, i.e., $\boldsymbol{g}_{\boldsymbol{w}}^\mathsf{T} \Delta \geq 0$, then, for some $c \in [0, 1]$, we have

$$\Delta \mathcal{L}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) \geq \left(1 - c + \frac{\sigma_{\min}(\mathbf{H}_{\boldsymbol{w}})}{2\sigma_{\max}(\mathbf{H}_J)^2} \right) \|\mathbf{\Pi}_J \boldsymbol{g}_{\boldsymbol{w}}\|_2^2. \tag{38}$$

As in **Theorem 1**, the masked empirical-risk response must consider that $\mathbf{H}_J \in \mathbb{R}^{n\times k}$ $(n > k)$ cannot span the whole space $\mathbb{R}^n$ and thus leaves a component $(\mathbf{I} - \mathbf{\Pi}_J)\boldsymbol{g_w}$ of the target gradient unrepresented. Recall that the GIF displacement is $\Delta = \left[\mathbf{H}_J^+ \boldsymbol{g_w} \mid \mathbf{0}\right]^\mathsf{T}$.

By Taylor's theorem applied to $\mathcal{L}_{\boldsymbol{w}}$ along the segment $\hat{\boldsymbol{\theta}} + t\Delta$, $t \in [0,1]$, and by the local convexity lower bound,

$$\Delta\mathcal{L}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) = \mathcal{L}_{\boldsymbol{w}}(\hat{\boldsymbol{\theta}} + \Delta) - \mathcal{L}_{\boldsymbol{w}}(\hat{\boldsymbol{\theta}}) \tag{39}$$

$$\geq \boldsymbol{g}_{\boldsymbol{w}}^\mathsf{T}\Delta + \frac{\sigma_{\min}(\mathbf{H}_{\boldsymbol{w}})}{2}\|\Delta\|_2^2. \tag{40}$$

Since $\boldsymbol{g}_{\boldsymbol{w}}^\mathsf{T}\Delta \geq 0$, there exists $c \in [0,1]$ such that

$$\boldsymbol{g}_{\boldsymbol{w}}^\mathsf{T}\Delta \geq (1-c)\|\mathbf{\Pi}_J \boldsymbol{g_w}\|_2^2. \tag{41}$$

In the weakest nonnegative-alignment case, this is satisfied by taking $c = 1$.

Moreover, since $\mathbf{\Pi}_J \boldsymbol{g_w} = \mathbf{H}_J\mathbf{H}_J^+\boldsymbol{g_w}$, we have

$$\|\mathbf{\Pi}_J \boldsymbol{g_w}\|_2 = \|\mathbf{H}_J\mathbf{H}_J^+\boldsymbol{g_w}\|_2 \tag{42}$$

$$\leq \sigma_{\max}(\mathbf{H}_J)\|\mathbf{H}_J^+\boldsymbol{g_w}\|_2 \tag{43}$$

$$= \sigma_{\max}(\mathbf{H}_J)\|\Delta\|_2. \tag{44}$$

Therefore,

$$\|\Delta\|_2^2 \geq \frac{\|\mathbf{\Pi}_J \boldsymbol{g_w}\|_2^2}{\sigma_{\max}(\mathbf{H}_J)^2}. \tag{45}$$

Substituting these two lower bounds gives

$$\Delta\mathcal{L}_J(\boldsymbol{w} \mid \hat{\boldsymbol{\theta}}) \geq (1-c)\|\mathbf{\Pi}_J \boldsymbol{g_w}\|_2^2 + \frac{\sigma_{\min}(\mathbf{H}_{\boldsymbol{w}})}{2}\frac{\|\mathbf{\Pi}_J \boldsymbol{g_w}\|_2^2}{\sigma_{\max}(\mathbf{H}_J)^2} \tag{46}$$

$$= \left(1 - c + \frac{\sigma_{\min}(\mathbf{H}_{\boldsymbol{w}})}{2\sigma_{\max}(\mathbf{H}_J)^2}\right)\|\mathbf{\Pi}_J \boldsymbol{g_w}\|_2^2. \tag{47}$$

G Proof of Theorem 3

Theorem 3 (Convergence of p-LiSSA)**.** Assume that $\mathbf{H}_J$ has full column rank.
i) The p-LiSSA is scale-invariant, i.e., scalar multiplication on $\ell(\cdot)$ does not change $\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J | \hat{\boldsymbol{\theta}}, \epsilon\big)$.
ii) If $0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J)}$, then (22) converges to $\mathbf{H}_J^+\boldsymbol{g_w}$.
iii) With $\mu = \frac{2}{\lambda_{\max}(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J) + \lambda_{\min}(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J)}$, $\|\mathcal{I}_k - \mathbf{H}_J^+\boldsymbol{g_w}\|_2 \leq \left(\frac{\kappa-1}{\kappa+1}\right)^k \|\mathcal{I}_0 - \mathbf{H}_J^+\boldsymbol{g_w}\|_2$, where $\kappa = \frac{\lambda_{\max}(\mathbf{H}_J^\top\mathbf{H}_J)}{\lambda_{\min}(\mathbf{H}_J^\top\mathbf{H}_J)}$.

Proof of i). We first show the scale-invariance of the GIF. For any $\mu > 0$, scaling $\ell(\cdot)$ and $\mathcal{L}(\cdot)$ by μ scales both $\mathbf{H}_J$ and $\boldsymbol{g_w}$ by μ. Hence,

$$\big((\mu\mathbf{H}_J)^\mathsf{T}(\mu\mathbf{H}_J)\big)^{-1}(\mu\mathbf{H}_J)^\mathsf{T}(\mu\boldsymbol{g_w}) = \big(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J\big)^{-1}\mathbf{H}_J^\mathsf{T}\boldsymbol{g_w} \tag{48}$$

$$= \mathbf{H}_J^+\boldsymbol{g_w} = \mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J | \hat{\boldsymbol{\theta}}, \epsilon\big). \tag{49}$$

Thus, GIF is scale-invariant.

Proof of ii). For convergence, since $\mathbf{H}_J$ has full column rank, $\mathbf{H}_J^\mathsf{T}\mathbf{H}_J$ is symmetric positive definite. Let $\lambda_i > 0$ denote the eigenvalues of $\mathbf{H}_J^\mathsf{T}\mathbf{H}_J$. Then, the eigenvalues of $\mathbf{I} - \mu\,\mathbf{H}_J^\mathsf{T}\mathbf{H}_J$ are $1 - \mu\lambda_i$. Therefore,

$$\rho\big(\mathbf{I} - \mu\,\mathbf{H}_J^\mathsf{T}\mathbf{H}_J\big) < 1 \quad \text{whenever} \quad 0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J)}. \tag{50}$$

Under this condition, $\|(\mathbf{I} - \mu\mathbf{H}_J^\mathsf{T}\mathbf{H})^i\| \to 0$ as $k \to \infty$, and the Neumann series converges as follows:

$$\lim_{k\to\infty}\mathcal{I}_k = -\sum_{i=0}^{\infty}\big(\mathbf{I} - \mu\mathbf{H}_J^\mathsf{T}\mathbf{H}_J\big)^i\,\mu\mathbf{H}_J^\mathsf{T}\boldsymbol{g_w} = \big(\mathbf{H}_J^\mathsf{T}\mathbf{H}_J\big)^{-1}\mathbf{H}_J^\mathsf{T}\boldsymbol{g_w} = \mathbf{H}_J^+\boldsymbol{g_w}. \tag{51}$$

Proof of iii). Let $\mathbf{e}_k = \mathcal{I}_k - \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}$. Since $\mathbf{H}_J$ has full column rank, $\mathbf{H}_J^\top \mathbf{H}_J$ is symmetric positive definite and $\mathbf{H}_J^\top \mathbf{H}_J \mathbf{H}_J^+ = \mathbf{H}_J^\top$. Then we have

$$\begin{aligned}
\mathbf{e}_k &= \mathcal{I}_k - \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}} \\
&\overset{(a)}{=} \mu \mathbf{H}_J^\top \boldsymbol{g}_{\boldsymbol{w}} + \left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) \mathcal{I}_{k-1} - \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}} \\
&\overset{(b)}{=} \mu \mathbf{H}_J^\top \boldsymbol{g}_{\boldsymbol{w}} + \left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) \mathcal{I}_{k-1} - \mu \mathbf{H}_J^\top \boldsymbol{g}_{\boldsymbol{w}} - \left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}} \\
&= \left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) \left(\mathcal{I}_{k-1} - \mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}}\right) \\
&= \left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) \mathbf{e}_{k-1},
\end{aligned} \tag{52}$$

where the equality (a) is obtained from (22), and the equality (b) holds because $\mathbf{H}_J^\top \mathbf{H}_J \mathbf{H}_J^+ = \mathbf{H}_J^\top$.

Since $\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J$ is symmetric, $\|(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J)^k\|_2 = \rho(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J)^k$, and therefore

$$\|\mathbf{e}_k\|_2 \leq \left\| \left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right)^k \right\|_2 \|\mathbf{e}_0\|_2 = \rho\left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right)^k \|\mathbf{e}_0\|_2, \tag{53}$$

where the inequality holds because $\|\mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2$.

Let $\lambda_{\min} = \lambda_{\min}(\mathbf{H}_J^\top \mathbf{H}_J)$ and $\lambda_{\max} = \lambda_{\max}(\mathbf{H}_J^\top \mathbf{H}_J)$. The eigenvalues of $\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J$ are $\{1 - \mu\lambda_i\}$, so

$$\rho\left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) = \max(|1 - \mu\lambda_{\min}|,\ |1 - \mu\lambda_{\max}|)\,. \tag{54}$$

The eigenvalues of $\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J$ are $\{1 - \mu\lambda_i\}$, so

$$\rho\left(\mathbf{I} - \mu \mathbf{H}_J^\top \mathbf{H}_J\right) = \max_i |1 - \mu\lambda_i| = \max(|1 - \mu\lambda_{\min}|,\ |1 - \mu\lambda_{\max}|)\,. \tag{55}$$

This is minimized when $|1 - \mu\lambda_{\min}| = |1 - \mu\lambda_{\max}|$, i.e., $1 - \mu\lambda_{\min} = \mu\lambda_{\max} - 1$, yielding

$$\mu^* = \frac{2}{\lambda_{\max} + \lambda_{\min}}, \qquad \rho^* = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{\kappa - 1}{\kappa + 1}, \tag{56}$$

where $\kappa = \lambda_{\max}/\lambda_{\min}$. Substituting ρ^* gives

$$\|\mathbf{e}_k\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\mathbf{e}_0\|_2, \tag{57}$$

where the convergence rate is given by $\frac{\kappa-1}{\kappa+1}$.

H Proof of Theorem 4

In this section, we provide a formal proof that the p-LiSSA algorithm becomes an unbiased estimator of the GIF update vector when utilizing a double sampling strategy.

Stochastic Estimator Definition. Recall that the target update vector for GIF is given by the limit of the Neumann series:

$$\Delta\hat{\boldsymbol{\theta}}_{\mathbf{1}+\epsilon\boldsymbol{w}} \propto -\mathbf{H}_J^+ \boldsymbol{g}_{\boldsymbol{w}} = -\sum_{i=0}^{\infty} \left(\mathbf{I} - \mathbf{H}_J^\intercal \mathbf{H}_J\right)^i \mathbf{H}_J^\intercal \boldsymbol{g}_{\boldsymbol{w}}. \tag{58}$$

Let $\tilde{\mathcal{I}}_k$ denote the stochastic estimate at iteration k. The standard p-LiSSA recurrence using a single batch is biased because

$$\mathbb{E}[\mathbf{H}_{J,B}^\intercal \mathbf{H}_{J,B}] = \mathbb{E}[\mathbf{H}_{J,B}]^\intercal \mathbb{E}[\mathbf{H}_{J,B}] + \underbrace{\mathrm{Cov}(\mathbf{H}_{J,B}, \mathbf{H}_{J,B})}_{\neq \mathbf{0}}. \tag{59}$$

To address this, we introduce the double sampling update rule.

At each iteration k, we independently sample two mini-batches $B_{k,1}$ and $B_{k,2}$ to compute the Hessian sub-matrices $\mathbf{H}_{J,B_{k,1}}$ and $\mathbf{H}_{J,B_{k,2}}$. The stochastic recurrence is defined as:

$$\tilde{\mathcal{I}}_0 = \mathbf{H}_{J,B_0}^\intercal \boldsymbol{g}_{\boldsymbol{w}}, \tag{60}$$

$$\tilde{\mathcal{I}}_k = \tilde{\mathcal{I}}_0 + \left(\mathbf{I} - \mathbf{H}_{J,B_{k,1}}^\mathsf{T}\mathbf{H}_{J,B_{k,2}}\right)\tilde{\mathcal{I}}_{k-1}, \tag{61}$$

where $\tilde{\mathcal{I}}_0$ is also computed using an independent batch B_0.

Theorem 4 (Unbiasedness of p-LiSSA)**.** Assume that the batches $B_{k,1}, B_{k,2}$ are drawn independently to compute $\mathbf{H}_{J,k,1}$ and $\mathbf{H}_{J,k,2}$. Define the stochastic recurrence as

$$\tilde{\mathcal{I}}_k = \tilde{\mathcal{I}}_0 + (\mathbf{I} - \mathbf{H}_{J,k,1}^\mathsf{T}\mathbf{H}_{J,k,2})\tilde{\mathcal{I}}_{k-1},\ \tilde{\mathcal{I}}_0 = \mathbf{H}_{J,0,1}^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}}. \tag{62}$$

Then, for any $k \geq 0$, the expected value of $\tilde{\mathcal{I}}_k$ satisfies

$$\mathbb{E}[\tilde{\mathcal{I}}_k] = \textstyle\sum_{j=0}^{k} \left(\mathbf{I} - \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\right)^j \mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}}. \tag{63}$$

Consequently, as $k \to \infty$, $\mathbb{E}[\tilde{\mathcal{I}}_k]$ converges to $\mathbf{H}_J^+\boldsymbol{g}_{\boldsymbol{w}}$.

We prove the claim by induction on k.

Base case ($k = 0$). By linearity of expectation and the unbiasedness of the mini-batch Hessian,

$$\mathbb{E}\big[\tilde{\mathcal{I}}_0\big] = \mathbb{E}\big[\mathbf{H}_{J,0,1}^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}}\big] = \mathbb{E}\big[\mathbf{H}_{J,0,1}\big]^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}} = \mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}}. \tag{64}$$

This coincides with the $j = 0$ term of the Neumann expansion.

Inductive step. Assume that, for $k-1$, the following identity holds:

$$\mathbb{E}\big[\tilde{\mathcal{I}}_{k-1}\big] = \sum_{j=0}^{k-1} \left(\mathbf{I} - \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\right)^j \mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}}. \tag{65}$$

Taking expectation on both sides of (62) yields

$$\mathbb{E}\big[\tilde{\mathcal{I}}_k\big] = \mathbb{E}\big[\tilde{\mathcal{I}}_0\big] + \mathbb{E}\big[\tilde{\mathcal{I}}_{k-1}\big] - \mathbb{E}\big[\mathbf{H}_{J,k,1}^\mathsf{T}\mathbf{H}_{J,k,2}\tilde{\mathcal{I}}_{k-1}\big]. \tag{66}$$

Since the mini-batches $B_{k,1}$ and $B_{k,2}$ are sampled independently at iteration k, and are independent of $\tilde{\mathcal{I}}_{k-1}$ (which depends only on the batches from earlier iterations), we can factor the expectation as

$$\mathbb{E}\big[\mathbf{H}_{J,k,1}^\mathsf{T}\mathbf{H}_{J,k,2}\tilde{\mathcal{I}}_{k-1}\big] = \mathbb{E}\big[\mathbf{H}_{J,k,1}^\mathsf{T}\big]\mathbb{E}\big[\mathbf{H}_{J,k,2}\big]\mathbb{E}\big[\tilde{\mathcal{I}}_{k-1}\big] = \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\,\mathbb{E}\big[\tilde{\mathcal{I}}_{k-1}\big]. \tag{67}$$

Therefore,

$$\mathbb{E}\big[\tilde{\mathcal{I}}_k\big] = \mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}} + \big(\mathbf{I} - \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\big)\mathbb{E}\big[\tilde{\mathcal{I}}_{k-1}\big]. \tag{68}$$

Applying the inductive hypothesis gives

$$\mathbb{E}\big[\tilde{\mathcal{I}}_k\big] = \mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}} + \big(\mathbf{I} - \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\big)\sum_{j=0}^{k-1}\left(\mathbf{I} - \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\right)^j\mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}} \tag{69}$$

$$= \sum_{j=0}^{k}\left(\mathbf{I} - \mathbf{H}_J^\mathsf{T}\mathbf{H}_J\right)^j\mathbf{H}_J^\mathsf{T}\boldsymbol{g}_{\boldsymbol{w}}. \tag{70}$$

This completes the induction. Hence, p-LiSSA with double sampling provides an unbiased estimate of the truncated Neumann series for the GIF.

I A Blockwise Interpretation of CAPS

This section provides a derivation that motivates the blockwise CAPS score. The derivation should be understood as an interpretation of the restricted least-squares structure of GIF. Our goal is to explain why a block score should jointly depend on a target-gradient term and a curvature term, and why the practical score in Sec. 5 preserves this structure.

Recall that GIF is defined through the restricted linearized problem

$$\hat{\theta}_J(1 + \epsilon\boldsymbol{w}) := \underset{\theta \in \mathbb{R}^k}{\arg\min} \left\| \epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_J(\theta - \hat{\theta}_J) \right\|_2^2, \tag{71}$$

whose solution is

$$\Delta\hat{\theta}_{J,1+\epsilon\boldsymbol{w}} = -\epsilon\mathbf{H}_J^{+}\boldsymbol{g}_{\boldsymbol{w}}. \tag{72}$$

Thus, GIF is a restricted update that minimizes the residual of the linearized first-order condition on the editable subspace.

To assess the quality of a candidate editable block B, consider the same restricted problem when only the parameters in B are allowed to move. Let $\Delta_B := \theta_B - \hat{\theta}_B$. Then the corresponding blockwise linearized problem is

$$\min_{\Delta_B} \|\epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_B\Delta_B\|_2^2, \tag{73}$$

where $\mathbf{H}_B$ denotes the columns of the Hessian indexed by block B. Expanding the squared norm gives

$$\|\epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_B\Delta_B\|_2^2 = \|\epsilon\boldsymbol{g}_{\boldsymbol{w}}\|_2^2 + 2\epsilon\,\boldsymbol{g}_{\boldsymbol{w}}^{\mathsf{T}}\mathbf{H}_B\Delta_B + \Delta_B^{\mathsf{T}}\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B\Delta_B. \tag{74}$$

The constant term does not affect the ranking of candidate blocks. Block quality therefore depends on both the linear term $\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}}$ and the quadratic term $\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B$. The former measures the response of block B to the target gradient, while the latter describes the local geometry of the restricted solve.

Assume that $\mathbf{H}_B$ has full column rank. Then the unique minimizer is

$$\Delta_B^{\star} = -\epsilon(\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B)^{-1}\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}} = -\epsilon\mathbf{H}_B^{+}\boldsymbol{g}_{\boldsymbol{w}}. \tag{75}$$

Substituting this solution back into the objective yields

$$\min_{\Delta_B} \|\epsilon\boldsymbol{g}_{\boldsymbol{w}} + \mathbf{H}_B\Delta_B\|_2^2 = \epsilon^2\|\boldsymbol{g}_{\boldsymbol{w}}\|_2^2 - \epsilon^2(\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}})^{\mathsf{T}}(\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B)^{-1}(\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}}). \tag{76}$$

Therefore, up to the constant factor ϵ^2, the predicted reduction achieved by block B is

$$(\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}})^{\mathsf{T}}(\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B)^{-1}(\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}}). \tag{77}$$

For unlearning, where $\epsilon = -1$, the factor ϵ^2 is fixed and can be ignored for ranking. Equation (77) shows that a favorable block should both i) have a strong response to the target gradient and ii) admit a stable restricted solve through a well-behaved Gram matrix $\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B$. This is exactly the same gradient–curvature structure that appears in the GIF solution and in the p-LiSSA iteration through $\mathbf{H}_J^{\mathsf{T}}\mathbf{H}_J$.

The exact quantity in Eq. (77) is expensive to evaluate for every candidate block, as it requires explicitly forming or repeatedly applying $\mathbf{H}_B$ and $\mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B$. For this reason, Sec. 5 adopts a practical proxy that preserves the same structure while replacing the exact blockwise least-squares terms by computable quantities. Specifically, we replace

$$\mathbf{H}_B^{\mathsf{T}}\boldsymbol{g}_{\boldsymbol{w}} \quad \text{by} \quad \boldsymbol{g}_{\boldsymbol{w},B}, \qquad \mathbf{H}_B^{\mathsf{T}}\mathbf{H}_B \quad \text{by} \quad \mathbf{C}_B + \lambda\mathbf{I}, \tag{78}$$

where $\boldsymbol{g}_{\boldsymbol{w},B}$ is the target-data gradient restricted to block B, $\mathbf{C}_B$ is the local curvature proxy defined in Sec. 5, and $\lambda \geq 0$ is a small numerical stabilizer. This leads to the blockwise CAPS score

$$S(B) := \boldsymbol{g}_{\boldsymbol{w},B}^{\mathsf{T}}(\mathbf{C}_B + \lambda\mathbf{I})^{-1}\boldsymbol{g}_{\boldsymbol{w},B}. \tag{79}$$

Accordingly, $S(B)$ should be interpreted as a practical blockwise proxy for the gain suggested by the restricted GIF solve, rather than as an exact identity derived from Eq. (15).

This interpretation also clarifies the role of the two components of CAPS. The gradient term favors blocks that are strongly associated with the target data, while the inverse-curvature term suppresses blocks whose local geometry is too flat or ill-conditioned to support a stable restricted update. Consistent with **Theorem 1**, we therefore discard numerically unstable blocks before ranking them by $S(B)$. This yields a block selection rule that is aligned with the restricted least-squares structure of GIFs while remaining computationally tractable.

J Reproducibility and Compute Resources

Dataset. We use five public benchmark datasets. For MNIST [31], we use the Hugging Face `ylecun/mnist` dataset entry, which lists the MIT License. For CIFAR-10 [27], we use the standard CIFAR-10 benchmark. The original distribution page does not state an explicit dataset license, and the UCI entry directs users there for licensing information.[4] For SVHN [36], we use the Street View House Numbers benchmark from its public research distribution, which is marked for non-commercial use only.[5] For 20 Newsgroups [30], we use the standard 20 Newsgroups text-classification benchmark. The original distribution page does not specify a corpus-wide license.[6] For PubMed 20k RCT [12], we use the public PubMed RCT benchmark. The dataset authors refer users to NLM's PubMed terms, which note that some abstracts may be protected by copyright.[7] For text classification, we initialize from the Hugging Face checkpoint `google/bert_uncased_L-2_H-128_A-2`, which is associated with BERT [13] and lists the Apache-2.0 license.

Model training. The five base checkpoints used in the model-editing experiments were trained under a common supervised-learning protocol. All checkpoints were optimized for 100 epochs with cross-entropy loss, cosine annealing of the learning rate, and model selection by validation accuracy. Epoch-wise checkpoints were also saved to provide the training trajectories used by TracIn [38]. For class-removal experiments, the corresponding retrained reference models were trained with the same procedure after excluding the target class from each training batch.

For MNIST [31], we trained an eight-layer fully connected network with hidden width 512 and dropout probability 0.1 using SGD with learning rate 0.05, momentum 0.9, weight decay 5×10^{-4}, and batch size 256. For CIFAR-10 [27] and SVHN [36], we trained ResNet-18 and VGG-11 checkpoints, respectively, using the same SGD hyperparameters and batch size. The CIFAR-10 and SVHN models use the Hugging Face dataset pipeline, which reserves a validation split from the training data when no official validation split is provided. For 20 Newsgroups [30] and PubMed 20k RCT [12], we fine-tuned `google/bert_uncased_L-2_H-128_A-2` with a linear classification head. The text models were trained with AdamW, learning rate 2×10^{-5}, weight decay 10^{-2}, batch size 32, maximum sequence length 80, dropout probability 0.1, and gradient clipping at norm 1.0. After 100 training epochs, the selected checkpoints achieved held-out test accuracies of 98.30% on MNIST, 86.09% on CIFAR-10, 95.24% on SVHN, 65.84% on 20 Newsgroups, and 83.96% on PubMed 20k RCT. All numerical experiments were run on a local Linux workstation. The workstation contains an AMD Ryzen Threadripper 7970X processor with 32 physical cores and 64 hardware threads, 256 GB of system memory, and two NVIDIA RTX 6000 Ada Generation GPUs with 48 GB of memory per GPU. The NVIDIA driver version is 580.126.09, and CUDA version is 13.0.

Model edit. Each empirical study was repeated across independent random seeds, and all aggregate results in Sec. 5 were computed from the corresponding per-seed measurements. For the solver study, we additionally recorded elapsed wall-clock time and peak memory for each run. These measurements are reported together with the residual metric in Fig. 4. We use class-level data removal as the model-editing task. In the theoretical formulation of Eq. (1), setting $\epsilon = -1$ gives

$$\frac{1}{m}\sum_{i=1}^{m}(1-w_i)\ell(z_i,\boldsymbol{\theta}), \tag{80}$$

which removes the target examples from the original empirical risk. The retrained reference model, however, minimizes the same retained loss after renormalizing by the number of retained examples,

$$\frac{1}{m-\|\boldsymbol{w}\|_1}\sum_{i=1}^{m}(1-w_i)\ell(z_i,\boldsymbol{\theta}). \tag{81}$$

This normalization matters in implementation because the target loss is computed as an average over the selected target examples, while the retained loss is computed as an average over the retained

[4] `https://www.cs.toronto.edu/~kriz/cifar.html` and `https://archive.ics.uci.edu/dataset/691/cifar+10`

[5] `http://ufldl.stanford.edu/housenumbers/`

[6] `http://qwone.com/~jason/20Newsgroups/`

[7] `https://github.com/Franck-Dernoncourt/pubmed-rct` and `https://ftp.ncbi.nlm.nih.gov/pubmed/baseline/README.txt`

examples. Consequently, the reduced-risk first-order condition around $\hat{\boldsymbol{\theta}}$ is linearized as

$$0 \approx -\frac{1}{m - \|\boldsymbol{w}\|_1} \sum_{i=1}^{m} w_i \nabla \ell(z_i, \hat{\boldsymbol{\theta}}) + \frac{1}{m - \|\boldsymbol{w}\|_1} \sum_{i=1}^{m} (1 - w_i) \nabla^2 \ell(z_i, \hat{\boldsymbol{\theta}}) \Delta \tag{82}$$

$$= -\frac{m}{m - \|\boldsymbol{w}\|_1} \boldsymbol{g}_{\boldsymbol{w}} + \frac{m}{m - \|\boldsymbol{w}\|_1} \mathbf{H}_{\mathbf{1}-\boldsymbol{w}} \Delta. \tag{83}$$

Multiplying the last line by $(m - \|\boldsymbol{w}\|_1)/m$ recovers the same linearized system as Eq. (5) at $\epsilon = -1$. Empirically, this corresponds to computing HVPs on retained examples and multiplying the averaged target loss by $\|\boldsymbol{w}\|_1/(m - \|\boldsymbol{w}\|_1)$.

The remaining discrepancy is that influence functions provide a local linearized direction, whereas the experiments require finite model edits. Following the standard observation that fixed theoretical influence magnitudes can underestimate finite data-removal effects in neural networks [24], we use the solver output as an edit direction and choose the finite displacement empirically. We normalize the influence direction and apply a sequence of candidate edits

$$\boldsymbol{\theta}_J^{(t+1)} = \boldsymbol{\theta}_J^{(t)} + \gamma \frac{\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J \mid \hat{\boldsymbol{\theta}}, \epsilon\big)}{\|\mathcal{I}\big(\boldsymbol{w}, \hat{\boldsymbol{\theta}}_J \mid \hat{\boldsymbol{\theta}}, \epsilon\big)\|_2}, \quad \boldsymbol{\theta}_J^{(0)} = \hat{\boldsymbol{\theta}}_J, \tag{84}$$

where γ is the finite edit scale. After each candidate step, we evaluate target-class accuracy and retained accuracy. The reported edited model is the candidate with the best model-editing score, with GIF additionally required to satisfy the target-accuracy constraint used in Table 1.

K Do GIF Updates Behave Like From-Scratch Retraining?

The GIF only updates a small model segment, leaving most parameters unchanged from their original values. However, this might contradict the nature of influence functions, which require the updated model to behave similarly to the model retrained from scratch with the edited dataset. We further verify how the updated model internally behaves by observing output distributions and Gradient-weighted Class Activation Mapping (Grad-CAM) visualization [41] of its outputs.

Inference histogram. We first train the VGG-11 network (**Before**) with the CIFAR-10 dataset, and then perform a data removal task using the GIF. The **Top-ρ% gradients** method selects 5, 15, 30% of model parameters. The selected parameters are then updated via the GIF, referred to as **MR@5, 15, 30%**, respectively. Lastly, we retrain the network from scratch (**Retraining**) while excluding the target removal data.

Fig. 6 shows how often each label is inferred when deleted data is provided to each network. The histograms obtained from **MR@5, 15, 30%** show a pattern similar to the **Retraining**. The results indicate that the updated networks properly encode features of the removed data, similar to the **Retraining**, even though only a small portion of parameters are updated. This contrasts with the approach proposed in [55], which renders features of removed data randomly.

Grad-CAM visualization. We visualize Grad-CAM results in Fig. 7 by feeding the removed and remaining images through the five models obtained by the same process as the inference histogram. The fifth and sixth convolutional layers (Conv2d) are selected for analysis because gradient outputs from earlier layers provide overly detailed outputs, while those from later layers are excessively coarse. The updated models **MR@5, 15, 30%** show heatmaps similar to those of **retraining** for the removed images. Interestingly, these four models tend to avoid the heatmap obtained from the model **before**. For the remaining images, the heatmap patterns from **MR@5, 15, 30%** are more similar to the **before** than the **retraining**, as they mostly share the same parameters. However, all patterns show a negligible discrepancy since the five models provide correct inferences for the remaining images.

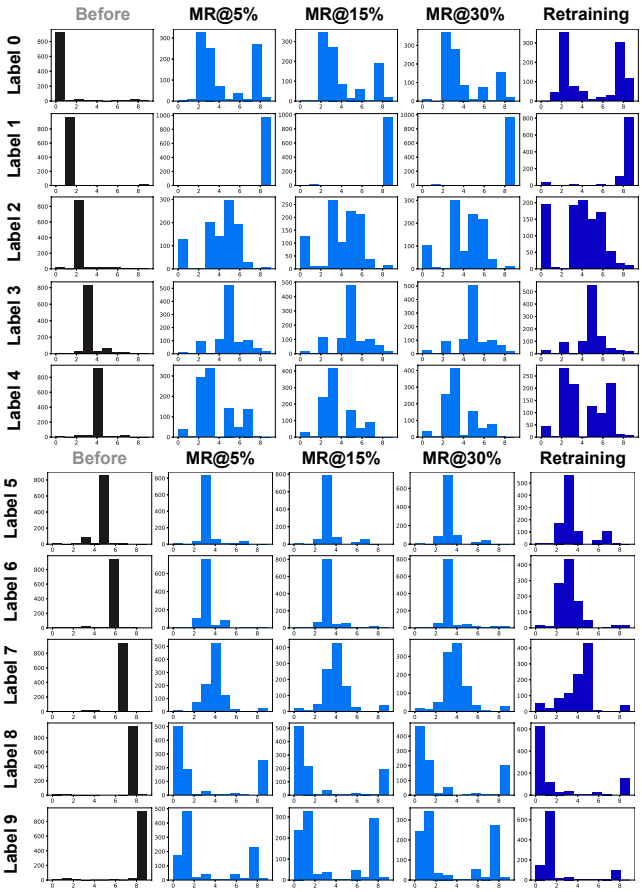


Figure 6: Inference histograms for the original model before removal, the three models updated by GIF, and the retrained model.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contributions on generalized influence functions, target-parameter selection, and p-LiSSA. These claims are supported by the theoretical development in Secs. 3 and 3.2, and by the numerical experiments in Sec. 5.

   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?

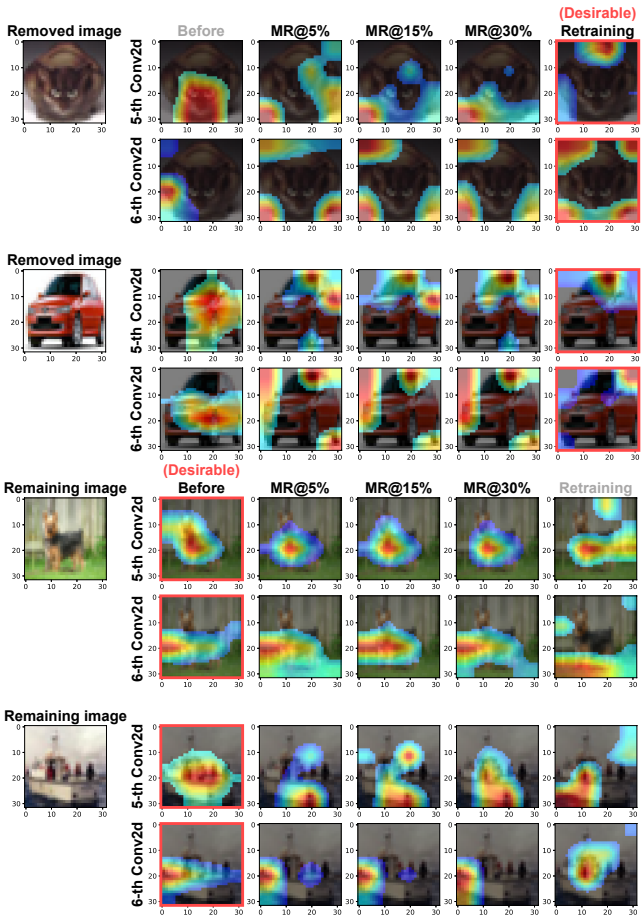


Figure 7: Grad-CAM visualization from the five models for the removed and retained images.

Answer: [Yes]

Justification: Sec. 6 discusses the trade-off between conditioning and target-gradient coverage, curvature changes during large edits, and the cost of repeated HVPs. It also identifies computational and evaluation challenges in extending GIFs to generative model editing.

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 theorem statements specify the assumptions used in the analysis, including the local smoothness and strong-convexity conditions for the GIF error bound. Proofs of the error bound, self-loss response, p-LiSSA convergence, and p-LiSSA unbiasedness are provided in Appendices D, F, G, and H.

   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: Sec. 5 specifies the experimental tasks, datasets, baselines, evaluation metrics, parameter-selection setting, and seed aggregation used for the main results. Appendix J further reports the training protocols, model architectures, data-split procedure, hyperparameters, checkpointing, retraining reference models, hardware, software, and per-run timing/memory measurements needed to reproduce 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: [Yes]

   Justification: The implementation and experiment scripts are publicly available at `https://github.com/hslyu/GIF`. Appendix J describes the public dataset sources, model configurations, optimization hyperparameters, and compute environment used in the experiments.

   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 J provides the experimental settings needed to understand and reproduce the results, including the training protocol, validation-based model selection, retraining procedure for class-removal references, dataset split pipeline, model architectures, optimizers, learning rates, weight decay, batch sizes, dropout, sequence length, gradient clipping, and number of epochs. Sec. 5 additionally defines the evaluation tasks, baselines, metrics, parameter-selection ratios, and seed aggregation used in the reported results.

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: Sec. 5 states that reported means and error bands are computed over 20 random seeds, with table $\pm$ values and shaded figure regions denoting sample standard deviations. Table 1 reports mean $\pm$ standard deviation for the model-editing results, and Figs. 3 and 4 report seed-averaged curves with variability bands.

   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 J reports the workstation CPU, system memory, GPU type, GPU memory, NVIDIA driver, and CUDA version used for the numerical experiments. Sec. 5 and Fig. 4 additionally report elapsed time and peak memory for the iHVP solver 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 experiments use publicly available benchmark datasets and do not involve new data collection, crowdsourcing, human-subject experiments, or deployment in high-stakes settings. Dataset sources, usage conditions, and reproducibility details are documented in Appendix J.

   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 work may have positive impact by making model editing and machine unlearning more reliable and reproducible, which can support responsible correction or removal of learned behavior. The paper is methodological and limits its empirical claims to public benchmark datasets and controlled class-removal experiments rather than deployed high-stakes systems, so we do not identify direct negative societal impacts beyond the general limitations of model-editing research.

    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 does not release high-risk pretrained models, image generators, scraped datasets, or other artifacts that require special access controls. The planned public release consists of research code and instructions for experiments on publicly available benchmark datasets.

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: Appendix J cites the dataset and checkpoint creators, lists the sources, and summarizes available licenses and usage conditions. It records SVHN's non-commercial use restriction, the PubMed terms, and the absence of an explicit dataset license on the original CIFAR-10 and 20 Newsgroups distribution pages.

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 introduce new datasets, pretrained models, or benchmarks. The experiments use existing public datasets and the public Hugging Face checkpoint identified in Appendix J.

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, participant recruitment, user studies, annotation labor, or other 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 or research with human subjects, so IRB approval or equivalent human-subjects review is not applicable.

    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: LLMs were not used as an important, original, or non-standard component of the core research method. They were used only for English polishing, sentence refinement, LaTeX cleanup, reference sanity checks, and assistance with implementing some baseline code, all of which were reviewed and validated by the authors.

    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.