Soft Weighted Machine Unlearning

Anonymous Authors[1]

Abstract

Machine unlearning, as a post-hoc processing technique, has gained widespread adoption in addressing challenges like bias mitigation and robustness enhancement. However, existing non-privacy unlearning-based solutions persist in using binary data removal framework designed for privacy-driven motivation, leading to significant information loss, a phenomenon known as "over-unlearning". While over-unlearning has been largely described in many studies as primarily causing utility degradation, we investigate its fundamental causes and provide deeper insights in this work through counterfactual leave-one-out analysis. In this paper, we introduce a weighted influence function that assigns tailored weights to each sample by solving a convex quadratic programming problem analytically. Building on this, we propose a soft-weighted framework enabling fine-grained model adjustments to address the over-unlearning challenge. We demonstrate that the proposed soft-weighted scheme is versatile and can be seamlessly integrated into most existing unlearning algorithms. Extensive experiments show that in fairness- and robustness-driven tasks, the soft-weighted scheme significantly outperforms hard-weighted schemes in fairness/robustness metrics and alleviates the decline in utility metric, thereby enhancing unlearning as an effective correction solution. Code is available at Soft Weighted Machine Unlearning.

1. Introduction

Modern machine learning (ML) models benefit greatly from the quantity and quality of the training data they are built upon. Depending on the type of the trained model being used, the impact of training samples can be either beneficial or detrimental. As a recent advancement, machine unlearning, originally conceived as a privacy-preserving mechanism to comply with data protection regulations' "the right to be forgotten" by allowing users to remove their personal data from models, has significantly broadened its scope. Beyond its privacy-oriented motivation, machine unlearning, as a post-hoc technique, has recently addressed broader practical concerns in trained models through efficient data removal, e.g., correcting bias (Chen et al., 2024b; Oesterling et al., 2024) and mitigating the detrimental effects (Liu et al., 2022; Zhang et al., 2023; Li et al., 2024; Kurmanji et al., 2024). These applications provide a fast way to adapt and edit a trained model without the prohibitively expensive process of retraining from scratch, catalyzing a paradigm shift in unlearning methodologies to address critical challenges beyond privacy concerns. However, influenced by the inertia of prior research rooted in privacy-centric considerations, these traditional methods solving non-privacy challenges operate under a binary framework: data is to remove or not to remove, which we refer to as hard-weighted unlearning framework in this paper, characterized by the complete elimination of undesired data influences. This framework,

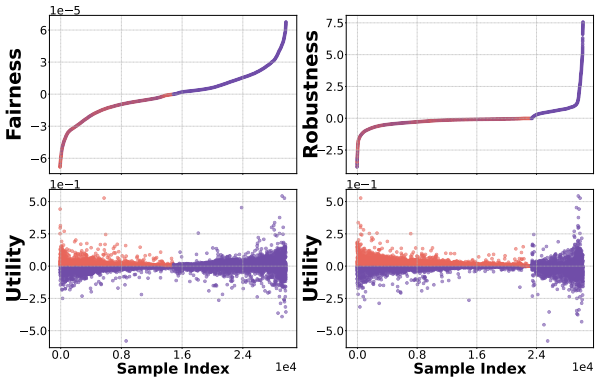


Figure 1. **Actual Changes in Utility and Fairness/Robustness** for each sample's leave-one-out model. **The X-axis represents** the sample indices. **The Y-axis for Fairness (Robustness)** displays changes in demographic parity (adversarial robustness loss) on the test data, with negative values indicating improved fairness (robustness) and positive values indicating reduced fairness (robustness). **The Y-axis for Utility** shows changes in test loss, with negative values indicating improved utility and positive values indicating diminished utility. Scatter points marked in **Red** indicate sample indices where fairness/robustness improves, but utility declines. This underscores the complex balance for unlearning algorithms, requiring finer-grained adjustments rather than hard data removal.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

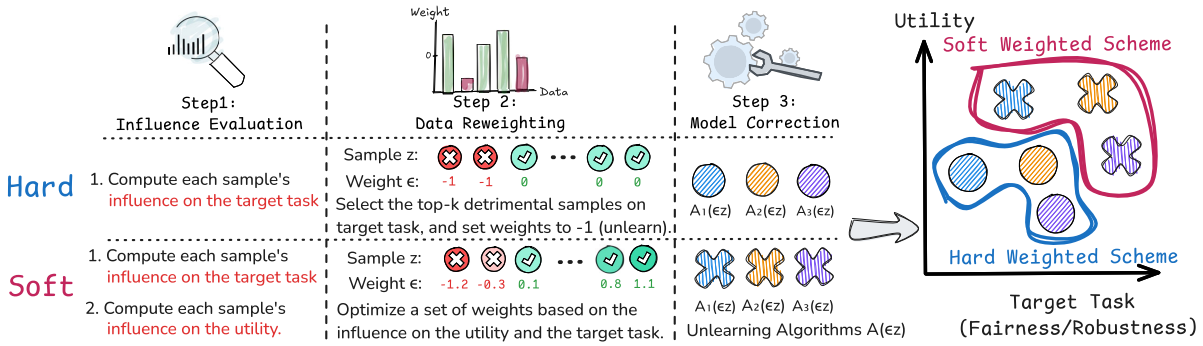


Figure 2. Illustration of difference of the proposed **soft** weighted framework versus the **hard** weighted framework in three steps.

while suitable for stringent privacy requirements, presents significant limitations when addressing more complex non-privacy-oriented challenges in modern ML systems, where the objective has transformed from regulatory-mandated data deletion to tasks such as enhancing model fairness, adversarial robustness, and generalization capabilities.

Specifically, the hard-weighted unlearning framework introduces several critical challenges: potential overcorrection, significant information loss, and compromised model generalization, collectively defined as **over-unlearning** by numerous studies (Hu et al., 2024; Chen et al., 2024a). The binary nature of hard-weighted decisions can lead to suboptimal outcomes, particularly when dealing with nuanced data distributions or complex objectives. We illustrate this concretely as evidence in Figure 1, where we trained a linear model on Adult dataset (Becker & Kohavi, 1996) and analyzed the performance of leave-one-out models obtained by removing each sample individually. Specifically, we evaluated changes in the following metrics as the differences between their post-removal and pre-removal values: fairness, quantified by Demographic Parity (Dwork et al., 2012); adversarial robustness, assessed through the loss on perturbed datasets (Megyeri et al., 2019); and generalization utility, determined by the loss on the test set. These results allowed us to uncover the underlying causes of over-unlearning:

❶ Fairness/Robustness and utility could be uncorrelated. The overall Spearman correlation coefficients for fairness/robustness and utility across all samples are -0.11 and -0.16 respectively, meaning that improvements in the target task do not always translate to better utility. Similarly, the red-highlighted samples in Figure 1 indicate that removing the most detrimental samples does not lead to accuracy gains, highlighting the primary cause of over-unlearning.

❷ Borderline forgetting samples are treated equivalently to highly detrimental samples by unlearning algorithm. The majority of forgetting samples are not the main contributors to model bias (vulnerability). However, in hard-weighted frameworks, such as gradient ascent algorithms (Jia et al., 2023), the samples are treated uniformly in an attempt to remove the most biased (vulnerable) ones, which can lead to excessive unlearning of these borderline samples. This, in turn, may cause these borderline samples to be flipped to unprivileged groups, resulting in opposite biases, or it may have the opposite effect on robustness.

❸ The majority of detrimental samples are maintained in remaining dataset. Approximately the top 50% (75%) of samples with values below 0 in Figure 1 exacerbate model bias and vulnerability. However, existing algorithms under the hard-weighted framework (Chen et al., 2024b) for unlearning 20% of the samples can only remove a limited number of samples and struggle to support further removal.

In this paper, we take the first step in addressing the challenge of over-unlearning when applying machine unlearning to other domains. To the best of our knowledge, our work is the first to uncover the root causes of over-unlearning and propose a framework to tackle this issue. Figure 2 illustrates our conceptual framework and highlights its differences from prior works (Chen et al., 2024b). We use influence functions as a tool, enabling the interchangeable use of various influence-based methods, and extend their applicability to a wider range of domains and scenarios, such as adversarial robustness, that were not previously considered. The key difference lies in our departure from the binary removal scheme inherited from privacy-driven motivations, instead adopting an optimization approach that allocates weights to each data. This more nuanced, soft treatment empirically shows improved performance on target tasks while enhancing utility. Our contributions are summarized as:

- We reveal the deeper causes of over-unlearning challenge from the perspective of counterfactual analysis in §1, offering insights for the development of machine unlearning.
- We introduce the weighted influence function in §4.1, a refined solution to address this challenge, with the weights through solving a convex quadratic programming problem in §4.2. We demonstrate that the soft weighted framework in §4.3 can be integrated into most unlearning methods.
- We empirically show in §5 that the proposed framework significantly boosts the performance of most existing algorithms in fairness/robustness tasks as well as utility, with only a few seconds of additional time overhead.

2. Related Works

Machine Unlearning, including recent cutting-edge methods such as (Kurmanji et al., 2023; Goel et al., 2022; Chen & Yang, 2023), is claimed to address challenges beyond its original privacy concerns, e.g., tackling issues like debiasing or enhancing robustness in well-trained models. These methods typically follow a paradigm where data to be forgotten is provided through deletion requests, after which the unlearning process is executed. These algorithms require prior knowledge to identify which data needs to be forgotten. (Chen et al., 2024b; Zhang et al., 2023) thus advanced an "Evaluation then Removal" framework, utilizing influence functions (Koh & Liang, 2017) for model debiasing. By using influence functions, the framework can first estimate the subset of data most responsible for model bias or vulnerability, thereby resolving the challenge of identifying forgetting dataset and subsequently unlearning undesired data. Building on these works, we take a step forward in addressing over-unlearning in fairness/robustness tasks when applying machine unlearning as a post-hoc processing technique.

Fairness and related ethical principles are crucial in ML research. Most methods for addressing unfairness rely on the concept of (un)privileged groups, which are disproportionately (less) likely to receive favorable outcomes. Fairness definitions in the literature focus on either group or individual fairness. Group fairness compares outcomes across groups but may harm within-group fairness, while individual fairness, such as counterfactual fairness which requires generating counterfactual samples, aims to ensure fairness across individuals (Hutchinson & Mitchell, 2019). As pointed out in (Caton & Haas, 2024), fairness notions are often incompatible and have limitations, with no universal metric or guideline for measuring fairness (Chouldechova, 2017; Kleinberg et al., 2018). Our study does not compare different fairness definitions but instead focuses on succinctly quantifying fairness using group fairness metrics, including Demographic Parity (DP) (Dwork et al., 2012) and Equal Opportunity (EOP) (Hardt et al., 2016), which are widely adopted in ML contexts (Chhabra et al., 2024).

Robustness, or in other words, the vulnerability of ML model predictions to minor sample perturbations (Eykholt et al., 2018), is another key aspect of ML research. In this paper, we focus on the influence of data on robustness. A related work (Xiong et al., 2024) summarizes the effects of data on adversarial robustness and highlights how to select data to enhance robustness. Similar to (Chhabra et al., 2024), we explore a white-box attack strategy to craft adversarial samples (Megyeri et al., 2019) targeting a linear model, which can be extended to methods such as FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2018). We quantify robustness as performance under adversarial attacks, referred to as perturbed accuracy (robustness), which is distinguished from utility known as standard test accuracy.

3. Preliminaries

Let $\ell(z;\theta)$ be a loss function for a given parameter Θ over parameter space Θ and sample z over instance space $\mathcal{Z}$. The empirical risk minimizer on the training dataset $\mathcal{D}=\{z_i=(\mathbf{x}_i,y_i)\}_{i=1}^n$ is given by $\hat{\theta}=\arg\min_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}^n\ell(z_i;\theta)$. For the empirical risk that is twice-differentiable and strictly convex[1] in the parameter space Θ, we slightly perturb the sample z_j by reweighting it with a weight $\epsilon_j\in\mathbb{R}$ as follows:

$$\hat{\theta}(z_j;\epsilon_j)=\arg\min_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}^{n}(\ell(z_i;\theta)+\epsilon_j\ell(z_j;\theta)). \quad (1)$$

Let $\epsilon_j=-1$ give $\hat{\theta}(z_j;-1)$, the empirical risk minimizer trained without sample z_j, and clearly, we have $\hat{\theta}=\hat{\theta}(z_j;0)$. Thus, using influence function (Koh & Liang, 2017) can efficiently capture model change through closed-form update:

$$\hat{\theta}(z_j;-1)-\hat{\theta}(z_j;0)\approx\frac{1}{n}\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_j;\hat{\theta}), \quad (2)$$

where $\mathbf{H}_{\hat{\theta}}\overset{\text{def}}{=}\frac{1}{n}\sum_{i-1}^n\nabla_\theta^2\ell(z_i,\hat{\theta})$ is the Hessian matrix. See more details in Appendix A. For a function f of interest, the actual change of function f is expressed as $\mathcal{I}^*(z_j;\epsilon)=f(\hat{\theta}(z_j;\epsilon))-f(\hat{\theta})$, which can be efficiently estimated by:

Utility: $\mathcal{I}_{\text{util}}(z_j;-1)=\sum_{z\in\mathcal{T}}\nabla_\theta\ell(z;\hat{\theta})^\top\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_j;\hat{\theta})$. $\mathcal{I}_{\text{util}}(z_j;-1)$ reflects the loss change in the test set $\mathcal{T}$, where a negative value indicates a lower test loss in a model trained without sample z_j, implying improvement in generalization.

Fairness: $\mathcal{I}_{\text{fair}}(z_j;-1)=\nabla_\theta f_{\text{fair}}(\mathcal{T};\hat{\theta})^\top\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_j;\hat{\theta})$. $f_{\text{fair}}(\mathcal{T};\hat{\theta})$ is instantiate by the fairness metrics in the test set $\mathcal{T}$. Specifically, consider binary sensitive attribute $g\in\{0,1\}$ and the predicted class probabilities $\hat{y}$. The group fairness metrics, i.e., demographic parity (DP) can be quantified by $f_{\text{DP}}(\mathcal{T};\hat{\theta})=\left|\mathbb{E}_{\mathcal{T}}[\hat{y}\mid g=0]-\mathbb{E}_{\mathcal{T}}[\hat{y}\mid g=1]\right|$, while equal opportunity (EOP) can be quantified by $f_{\text{EOP}}(\mathcal{T};\hat{\theta})=\left|\mathbb{E}_{\mathcal{T}}[\ell(z;\theta)\mid g=1,y=1]-\mathbb{E}_{\mathcal{T}}[\ell(z;\theta)\mid g=0,y=1]\right|$. Similar to the interpretation of utility, a negative value of $\mathcal{I}_{\text{fair}}(z_j;-1)$ indicates a lower $f_{\text{fair}}(\mathcal{T};\theta)$ on a model trained without sample z_j, implying an improvement in fairness.

Robustness: $\mathcal{I}_{\text{robust}}(z_j;-1)=\sum_{\tilde{\mathcal{T}}}\nabla_\theta\ell(\tilde{z};\hat{\theta})^\top\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_j;\hat{\theta})$. For a perturbed dataset $\tilde{\mathcal{T}}$ with adversarial sample $\tilde{z}=z-\gamma\frac{\hat{\theta}^\top z+b}{\hat{\theta}^\top\hat{\theta}}\hat{\theta}$ crafted from test sample $z\in\mathcal{T}$, where $\hat{\theta}$ denotes a linear model, $b\in\mathbb{R}$ is intercept, and $\gamma>1$ controls the magnitude of perturbation. Since the decision boundary is a hyperplane, adversary can change the prediction by adding minimal perturbations to move each sample orthogonally.

[1] The convexity makes the theoretical analysis of influence functions impossible in non-convex models, yet this does not invalidate the use of influence functions in practice. In non-convex scenarios, these strategies are widely adopted: (i) using a convex surrogate model on embeddings from the non-convex model (Guo et al., 2020; Chen et al., 2024b), (ii) adding a damping factor to ensure a positive definite Hessian (Zhang et al., 2024a), and (iii) reweighting gradient updates instead of loss in SGD-trained models, thereby avoiding the inversion of the Hessian (Qiao et al., 2024).

4. Proposed Approaches

We first introduce the weighted influence functions in §4.1, analytically deriving the weights by solving a convex quadratic programming problem in §4.2. This foundation enables fine-grained model adjustments through a soft-weighted machine unlearning framework, as detailed in §4.3. We then highlight its broad applicability and compatibility with diverse unlearning paradigms in §4.4.

4.1. Step 1: Weighted Influence Function

Due to the challenges of directly removing samples stated in §1, we do not explicitly set the binary weight $\epsilon=-1$ or $\epsilon=0$ as in previous works when perturbing Equation (1), but instead introduce the following weighted influence function:

- **Weighted Influence Function on Utility Metric:**

$$\mathcal{I}_{\text{util}}(z_j;\epsilon_j) = -\epsilon_j \sum_{z\in\mathcal{T}} \nabla_\theta \ell(z;\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}). \quad (3)$$

- **Weighted Influence Function on Fairness Metric:**

$$\mathcal{I}_{\text{DP/EOP}}(z_j;\epsilon_j) = -\epsilon_j \nabla_\theta f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}). \quad (4)$$

- **Weighted Influence Function on Robustness Metric:**

$$\mathcal{I}_{\text{robust}}(z_j;\epsilon_j) = -\epsilon_j \sum_{\tilde{z}\in\tilde{\mathcal{T}}} \nabla_\theta \ell(\tilde{z};\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}). \quad (5)$$

Note that for each of the aforementioned functions, $\mathcal{I}(z_j;\epsilon_j) = -\epsilon_j \mathcal{I}(z_j;-1)$, where ϵ_j is not binary (0 or -1), but can be optimized based on $\mathcal{I}(z_j;-1)$ in next step.

4.2. Step 2: Weights Discovery via Optimization

The goal is to discover $\boldsymbol{\epsilon}$ that ensure the model's utility is not adversely affected by the unlearning algorithms across different tasks, colloquially, mitigating over-unlearning. We formulate it as a convex quadratic programming problem:

$$\text{minimize}_{\boldsymbol{\epsilon}} \quad \sum_{i=1}^{n} \mathcal{I}_{\text{metric}}(z_i;\epsilon_i) + \lambda\|\boldsymbol{\epsilon}\|_2^2, \quad (6a)$$

$$\text{subject to} \quad \sum_{i=1}^{n} \mathcal{I}_{\text{metric}}(z_i;\epsilon_i) \geq -\Delta, \quad (6b)$$

$$\sum_{i=1}^{n} \mathcal{I}_{\text{util}}(z_i;\epsilon_i) \leq 0. \quad (6c)$$

In Equation (6a), depending on the target task, the first term $\mathcal{I}_{\text{metric}}(z_i;\epsilon_i)$ represents either $\mathcal{I}_{\text{fair}}(z_i;\epsilon_i)$ or $\mathcal{I}_{\text{robust}}(z_i;\epsilon_i)$. The second term seeks to penalize changes in the weights $\boldsymbol{\epsilon}$, ensuring that perturbations remain infinitesimal. In the first subjective Equation (6b), Δ quantifies the current model's fairness $f_{\text{fair}}(\mathcal{T};\hat{\theta})$ or robustness $\sum_{\tilde{z}\in\tilde{\mathcal{T}}} \nabla_\theta \ell(\tilde{z};\hat{\theta})^\top$. The constraint $-\Delta$ provides lower bound to prevent over-correction, which could lead to reverse bias or vulnerability. The second subjective Equation (6c) ensures that the resulting weights preserve the model's utility without compromise. Building on the problem setting, we can either use a linear solver or derive the following analytical solutions under different conditions to find a set of optimal weights,

$$\boldsymbol{\epsilon}^* = \begin{cases} \boldsymbol{\mathcal{I}}_{\text{metric}}/(2\lambda), & \text{Cond. 1} \\ \Delta/|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 \cdot \boldsymbol{\mathcal{I}}_{\text{metric}}, & \text{Cond. 2} \\ \left(\boldsymbol{\mathcal{I}}_{\text{metric}} - (\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}})/|\mathcal{I}_{\text{util}}|^2 \cdot \boldsymbol{\mathcal{I}}_{\text{util}}\right)/(2\lambda), & \text{Cond. 3} \\ \dfrac{\Delta\left(|\boldsymbol{\mathcal{I}}_{\text{util}}|^2 \boldsymbol{\mathcal{I}}_{\text{metric}} - \boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \boldsymbol{\mathcal{I}}_{\text{util}}\right)}{|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 |\boldsymbol{\mathcal{I}}_{\text{util}}|^2 - (\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}})^2}. & \text{Cond. 4} \end{cases} \quad (7)$$

Cond. 1: $0 \leq \boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 2\lambda\Delta$.

Cond. 2: $|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 - 2\lambda\Delta \geq 0,\ \boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \geq 0$.

Cond. 3: $\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 0$,
$(\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}})^2 \geq |\boldsymbol{\mathcal{I}}_{\text{util}}|^2 (|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 - 2\lambda\Delta)$.

Cond. 4: $\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 0$,
$(\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}})^2 \leq |\boldsymbol{\mathcal{I}}_{\text{util}}|^2 (|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 - 2\lambda\Delta)$.

Where $\boldsymbol{\mathcal{I}} = (\mathcal{I}(z_1;-1), \cdots, \mathcal{I}(z_n;-1))^\top$ for the training dataset $\mathcal{D} = \{z_i\}_{i=1}^n$. See Appendix A.2 for more details.

4.3. Step 3: Weighted Model Unlearning

Given the aforementioned optimization yielding weights $\boldsymbol{\epsilon}^*$, the influence function based unlearning algorithm can be updated in the following closed-form expression:

$$\hat{\theta}(\mathcal{D};\boldsymbol{\epsilon}^*) - \hat{\theta}(\mathcal{D};0) \approx -\frac{1}{n} \sum_{i\in\mathcal{D}} \epsilon_i^* \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_i;\hat{\theta}). \quad (8)$$

For the majority of classification models, Equation (8) can efficiently update the non-convex model's convex surrogate, i.e., by treating the earlier layers as feature extractors and updating the final fully connected linear layer, and its effectiveness has been demonstrated in many studies, such as, (Chen et al., 2024b; Chhabra et al., 2024; Guo et al., 2020; Koh & Liang, 2017). Nevertheless, for generative models, the strategies outlined in the footnote of §3 may not be as effective. In practice, for high-dimensional non-convex models, the statistical noise introduced by estimation can degrade the numerical stability of second-order information, diminishing its potential advantages. As a result, a more practical approach to updating the model is to use a diagonal matrix $\sigma\mathbf{I}$ with a constant σ to approximate the inverse of Hessian, and scaling it by the gradient variance as follows,

$$\hat{\theta}(z_j;\epsilon_j^*) - \hat{\theta}(z_j;0) \approx -\frac{\epsilon_j^*}{n} \sigma\mathbf{I} \cdot \nabla_\theta \ell(z_j;\hat{\theta}). \quad (9)$$

For simplicity, we define σ/n as learning rate η and estimate $\hat{\theta}(z_j;\epsilon_j^*)$ through multiple update rounds indexed by t,

$$\theta_{t+1}(z_j;\epsilon_j^*) - \theta_t(z_j;0) = -\epsilon_j^* \cdot \eta_t \nabla_\theta \ell(z_j;\hat{\theta}). \quad (10)$$

As can be seen in Equation (10), the soft-weighted scheme can be naturally applied to other unlearning algorithms, e.g., fine-tuning and gradient ascent algorithms, which are currently popular cutting-edge methods in both LLM unlearning (Jang et al., 2023; Yao et al., 2023; Zhang et al., 2024c) and non-LLM unlearning (Kurmanji et al., 2023).

Algorithm 1 Soft-Weighted Unlearning Framework

input Model $\hat{\theta}$, Training Dataset $\mathcal{D}$, Testing Dataset $\mathcal{T}$, Adversarial Samples $\tilde{z} \in \tilde{\mathcal{T}}$, Threshold δ

 # Step 1: Influence Evaluation.

for each sample $i \in \mathcal{D}$ **do**

  Utility: $\mathcal{I}_{\text{util}}(z_i; -1) \leftarrow$ Equation (3).

  Fairness: $\mathcal{I}_{\text{fair}}(z_i; -1) \leftarrow$ Equation (4),

  Robustness: $\mathcal{I}_{\text{robust}}(z_i; -1) \leftarrow$ Equation (5).

end for

 # Step 2: Weights Optimization.

Weights $\{\epsilon_i^*\}_{i=1}^n \leftarrow$ Equation (7)

 # Step 3: Model Correction.

if $f \leftarrow f_{\text{fair}}(\mathcal{T}; \theta)$ or $\sum_{\tilde{z} \in \tilde{\mathcal{T}}} \nabla_\theta \ell(\tilde{z}; \theta) \leq \delta$ **then**

  $\theta \leftarrow$ Equation (8) or Other Unlearning Algorithms

end if

output θ

4.4. Soft-Weighted Unlearning Framework

To further explore the effectiveness and applicability of the soft-weighted scheme, we elaborate on its relationship with previous baseline methods. Specifically, we define the weight of the forgetting sample as ϵ_f and the weight for the remaining sample as ϵ_r. In this context, the previous hard-weighted fine-tuning algorithm can be viewed as a special case of our scheme where $\epsilon_f = 0$ and $\epsilon_r = 1$, while the gradient ascent algorithm represents another special case where $\epsilon_f = -1$ and $\epsilon_r = 0$. Since each sample contributes differently to the model, assigning uniform weights can result in the loss of crucial information for prediction, highlighting the issue of over-unlearning as discussed in §1. In contrast, the soft scheme aligns with our intuition: mitigating highly detrimental effects while amplifying beneficial ones. Moreover, we empirically demonstrate that soft-weighted scheme can also be effectively applied to other heuristic unlearning algorithms, such as Fisher (Golatkar et al., 2020a) or Teacher-Student Formulation (Kurmanji et al., 2023) et al. Please refer to Appendix A.3 for further technical details and the soft-weighted version of the unlearning algorithms.

Accordingly, we propose the **Soft-Weighted Unlearning Framework** in Algorithm 1 to effectively address the over-unlearning challenges commonly encountered in existing non-privacy-oriented tasks, such as bias mitigation and robustness enhancement. This framework introduces a more nuanced approach to unlearning by assigning differentiated weights to samples based on their contributions to the model's objective. This insight underscores the effectiveness of the proposed soft-weighting method. Specifically, samples that positively contribute to the objective function are given higher weights, while those that conflict with it are assigned lower weights. The process of model correction is systematically structured into the following three key steps:

Step 1: Influence Evaluation. We use Eqs. (4) and (5) to evaluate the fairness or robustness impact of removing each sample. In contrast to previous work (Chen et al., 2024b) on fairness, we also use Equation (3) to evaluate utility impact.
Step 2: Weights Optimization. Based on the results from Step 1, we solve the optimization problem in Equation (6) to obtain a set of optimal weights for the training dataset.
Step 3: Model Correction. A straightforward way to update the model is through Equation (8). Nevertheless, our framework is not limited to influence-function-based methods; other unlearning algorithms can also leverage the weights obtained in Step 2 to perform model correction.

5. Experiments and Discussion

In this section, we conduct two types of experiments to evaluate our findings comprehensively. The first explanatory experiments in §5.1, designed to validate the rationale behind motivation discussed in §1 and methodology presented in §4. The second is applied experiments in §5.2, which assess the performance of the soft-weighted framework outlined in Algorithm 1 in addressing specific challenges, including bias mitigation and robustness improvement.

Datasets: In this work, we follow the experiments setup from (Chhabra et al., 2024) to evaluate on standard fairness and robustness datasets Specifically, we conducted experiments on **four real-world datasets**, including two tabular datasets **UCI Adult** (Becker & Kohavi, 1996), **Bank** (Moro et al., 2014), one visual human face dataset **CelebA** (Liu et al., 2015), one textual dataset **Jigsaw Toxicity** (Noever, 2018). Further details can be found in Appendix B.2.

Baselines: We follow the machine unlearning repository in (Kurmanji et al., 2023) with the following **nine unlearning algorithms**: Gradient Ascent (**`GA`**) combined with a regularizer Fine-Tuning (**`FT`**) for utility preservation (Following the definitions in (Shi et al., 2024), we denote these combinations as $\texttt{GA}_{\texttt{FT}}$.), Influence Function (**`IF`**) (Koh & Liang, 2017), Fisher Forgetting (**`Fisher`**) (Golatkar et al., 2020a) and NTK Forgetting (**`NTK`**) (Golatkar et al., 2020b), Teacher-Student Formulation (**`SCRUB`**) (Kurmanji et al., 2023) and (**`Bad-T`**) (Chundawat et al., 2023), Freezing and Forgetting Last k-layers Followed by Catastrophic Forgetting-k (**`CF-k`**) and Exact Unlearning-k (**`EU-k`**)(Goel et al., 2022), along with their **S**oft-**W**eighted (**`SW-`**) versions. Technical details can be found in Appendix A.3. We evaluated aforementioned unlearning methods on tasks involving fairness and robustness, where we defer EOP to the appendix. Similar to (Chhabra et al., 2024), we train a Logistic Regression (LR) and a Neural Network (NN) with two-layer non-linear structure followed by a linear layer. During the retraining or unlearning process, the last linear layer of NN is treated as a convex surrogate for the non-convex model, and only this part of parameters is updated.

5.1. Explanatory Experiments

❶ Correctness of Influence Evaluations. Whether using the hard- or soft-weighted scheme, it is necessary to evaluate the influence of each sample. However, due to high cost of retraining, it is impractical to train leave-one-out models to determine their actual influence. The soft-weighted framework offers Eqs. (3) to (5) to approximate the actual influence on the utility, fairness, and robustness metrics. The first question naturally is to verify its validity, colloquially,

(***Q***): How accurate is the influence evaluation in Step 1?

Note that while the validation of influence evaluation has been well established in previous studies, including traditional ML model (Koh & Liang, 2017; Chen et al., 2024b) and non-convex models (Jia et al., 2024; Zhang et al., 2024b), we provide additional justifications for the actual influence and its estimation in Step 1 for the setting in the main text to validate the reliability of influence estimation.

Result. The influence evaluation values are obtained using Eqs. (3) to (5), while the actual values are obtained by retraining a leave-one-out model for each sample. As illustrated in Figure 3 and Figure 4, the results from Step 1 exhibit a strong correlation with the actual values in terms of utility, fairness, and robustness metrics, with Spearman (Spearman, 1961) and Pearson (Wright, 1921) correlation coefficients close to or equal to 1 as depicted in Figure 3.

❷ Intuition of Over-Unlearning. After analyzing the counterfactual influence of each sample across different metrics, it becomes essential to understand how adjustments in the weighting strategy influence the model's behavior. In particular, we focus on the intuition behind the transition from the previous hard weights to the softened weights in Step 2.

(***Q***): What the over-unlearning intuition is between the pervious hard weights and softened weights in Step 2?

In §1, we discussed three main causes of over-unlearning, which led to the adoption of the soft-weighted scheme. To illustrate its advantages, we compare the weighting strategies of hard- and soft-weighted schemes in Step 2.

Results. As shown in Figure 5 (**A** and **D**), hard-weighted schemes (blue line) involves directly removing the most of biased or adversarially susceptible samples based on their counterfactual influence on fairness $\mathcal{I}_{\text{fair}}$ or robustness $\mathcal{I}_{\text{robust}}$, where the samples are sorted in ascending order based on influence value. While this approach effectively reduces bias/vulnerability, it neglects both the potential utility of these samples and the residual bias in the remaining data, potentially leading to degraded generalization performance and missed opportunities for further improvements in fairness or robustness. In contrast, the soft-weighted scheme employs a more refined adjustment mechanism. As illustrated in Figure 5 (**A**), the soft weights (red curve) exhibit a smoother

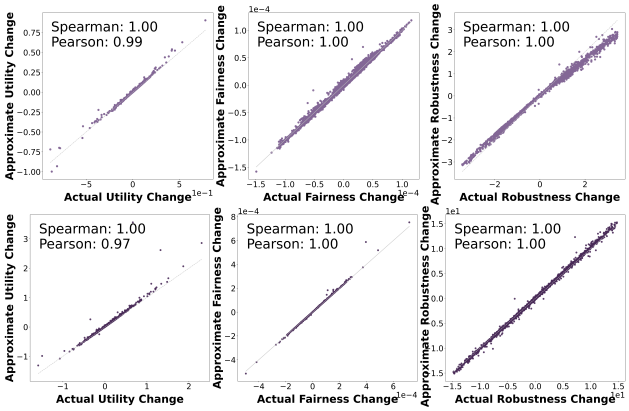


Figure 3. **Actual Changes vs. Approximate Changes.** We evaluated the leave-one-out influence for all training sample, with the **First Row** for LR and **Second Row** for the last layer of NN, on different performance metrics as follows: **(Left)** The model's utility, evaluated as the loss on test dataset. **(Middle)** The model's fairness, evaluated as the DP loss on test dataset. **(Right)** The model's robustness, evaluated as the loss on adversarial samples.

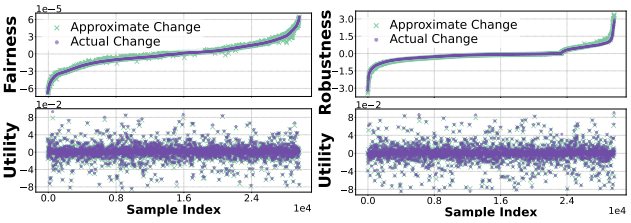


Figure 4. **Utility Changes vs. Fairness/Robustness Changes.** We evaluated the impact of all training data on different performance metrics as follows: **(Left)** The model's generalization ability, evaluated as the loss on the test dataset. **(Right)** The model's robustness, evaluated as the loss on adversarial test samples.

distribution compared to the hard weights (blue line). This reflects the scheme's ability to balance the influence of each sample more precisely, ensuring that moderately biased samples are not entirely removed but instead appropriately reweighted. Similarly, Figure 5 (**D**) demonstrates how the soft-weighted scheme integrates robustness considerations $\mathcal{I}_{\text{robust}}$, striking a delicate balance between mitigating vulnerabilities and preserving informative samples. Furthermore, the scheme accounts for utility $\mathcal{I}_{\text{util}}$, safeguarding valuable information to maintain the model's generalization capability. By preventing the excessive removal of samples with marginal yet meaningful contributions, the soft-weighted scheme mitigates the risk of over-unlearning. This balanced approach allows the model to achieve improved fairness/robustness without compromising its utility performance.

❸ Explanation of Model Correction. Building on the insights from Step 2, the next natural step is to explore how soft-weighted adjustments refine the model. A key aspect of this process is to observe the model's decision boundary dynamics. Specifically, we aim to understand:

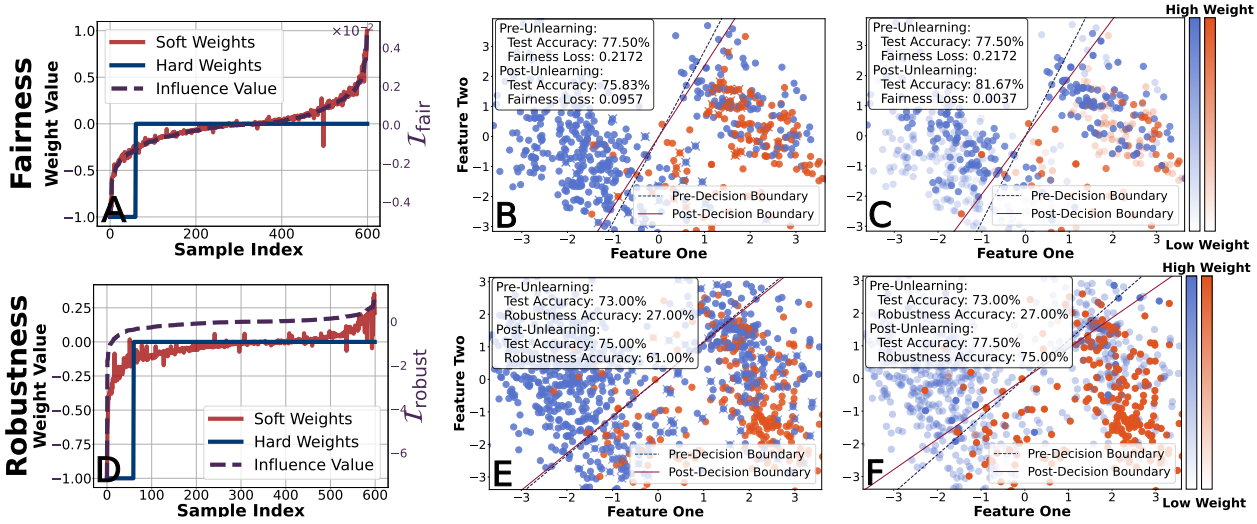


Figure 5. **Hard Weighted Scheme vs. Soft Weighted Scheme.** We use `IF` as the unlearning method to update model. **The First Row for Fairness** compares the hard- and soft-weighted schemes: **A** compares the weighting schemes with corresponding fairness influences, **B** presents fairness and utility before and after applying hard-weighted `IF`, and **C** shows the same for soft-weighted `IF`. **The Second Row for Robustness** follows a similar structure: **D** compares the weighting schemes and corresponding robustness influences, **E** presents robustness and utility before and after applying hard-weighted `IF`, and **F** shows the same for soft-weighted `IF`. Moreover, we use opacity to represent the value of weights. As shown, hard weighting applies the same weight after removing the most harmful samples, while soft weighting assigns different weights to each data, enabling the soft-weighted scheme to better capture the overall information.

(Q): How does the decision boundary change before and after soft-weighted model correction in Step 3?

To better visualize the decision boundary, we use a subset from the training set to obtain a well-trained linear model. As shown in Figure 5 (**B** and **E**), the hard-weighted scheme operates with limited information, focusing solely on the most harmful samples while lacking a global view of the other samples. This uniform weighting leads to a lack of information for the remaining data, resulting in limited adjustments. In contrast, the soft-weighted scheme provides a more holistic understanding of sample importance, allowing the decision boundary to align more closely with higher-weighted samples during classification. Consequently, samples with greater weights are more likely to be correctly classified. This intuition is clearly reflected in Figure 5 (**C** and **F**): compared to the decision boundary of the original pre-unlearning model, the post-unlearning model's decision boundary successfully classifies the high-weight samples in the upper-right region while ignoring the low-weight samples in the lower-left region.

This observation aligns well with our intuitions, namely that the unlearning process prioritizes the proper classification of high-weight samples, which are considered more influential in terms of model performance and fairness. Consequently, the model devotes more effort to aligning its decision boundary with these samples, even introducing misclassifications for low-weight samples. This trade-off reflects the inherent nature of soft-weighted schemes, where the emphasis is deliberately shifted toward optimizing outcomes for samples deemed more critical. Such behavior underscores the importance of selecting an appropriate weighting strategy. This balance is particularly crucial in applications where fairness, robustness, and utility are highly valued.

5.2. Applied Experiments

In this section, we evaluate the performance of different unlearning algorithms under a fixed budget of 30 epochs. For algorithms utilizing gradient descent, we set a learning rate of 0.01, while for those using gradient ascent, we set a learning rate of 0.0005, using full-batch updates. Unless otherwise specified, we use the entire training dataset by default. For LR, we demonstrate its performance on small datasets using 1,000 training samples from the Adult and Bank datasets. For the hard-weighted scheme, we consistently remove 20% of the data. It is important to note that unlearning methods' performance may vary across datasets/models depending on hyperparameter choices, and our selected configurations might not be optimal. Our goal is not to assess the superiority of each algorithm, but rather to compare the differences between hard-weighted and soft-weighted schemes, under the same setup and cost constraints.

Results. From Figure 6, we can observe the following: (i) In all scenarios **(A-P)** compared to the hard-weighted method, the soft-weighted scheme outperforms it in terms of target task performance. This improvement stems from optimizing the sample weights through objective Equation (6a) and constraint in Equation (6b). Moreover, considering the constraint in Equation (6c), the soft-weighting effectively alleviates utility degradation, which is a limitation often observed

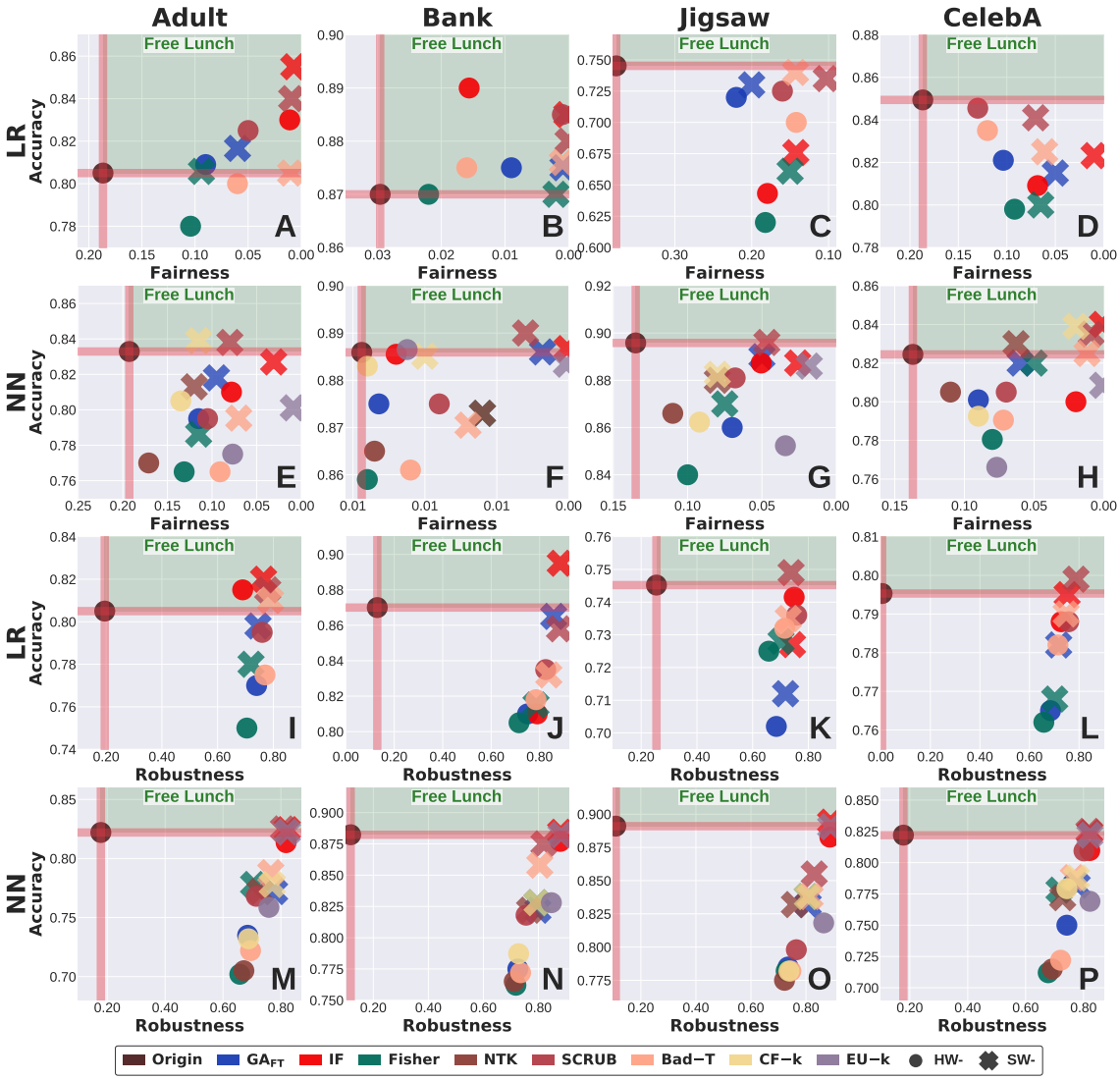


Figure 6. **Performance on Fairness/Robustness Tasks.** Different colors represent various unlearning algorithms: ● for the **H**ard-**W**eighted scheme and ✖ for the **S**oft-**W**eighted scheme. **The First Two Rows** (LR, NN) evaluate utility and fairness metrics, while **The Last Two Rows** (LR, NN) evaluate utility and robustness metrics across datasets. **The Green Region** highlights that **Free Lunch** cases occurs when unlearning improve both task performance and utility compared to original model. The soft weighting outperforms the hard weighting by enhancing task performance and mitigating decline in utility, even achieving free lunch in some of the unlearning algorithms.

in the hard-weighted approach. (ii) In most scenarios **(A-B, E-P)** compared to the original model, the soft-weighted scheme not only improves the target task performance but also enhances utility in certain algorithms, which we refer to as the "free lunch" cases in this paper, highlighting the dual improvement in both target performance and utility. (iii) In smaller datasets **(A-B, I-J)** compared to the original model, the free lunch cases becomes especially pronounced. Intuitively, this is because our method estimates the influence value of each data to compute the weights. In larger datasets, the cumulative estimation error becomes more pronounced, which can lead to a slight utility decline. Finally, compared to hard weighting, soft weighting incurs negligible overhead (<0.03% runtime increase) to calculate the weights, yet it yields substantial improvements. Due to space constraints, we defer the visualization results to Appendix B.3.

6. Conclusion

We investigate the underlying causes of over-unlearning through counterfactual contribution analysis. To address this challenge, we propose an innovative soft-weighted machine unlearning framework that is simple to apply for non-privacy tasks including but not limited to fairness and robustness. Specifically, we introduce weighted influence functions, and obtain weights by solving convex quadratic programming problem. In contrast to hard-weighted schemes, the finer-grained soft scheme empirically maintains superior task-specific performance and utility with negligible overhead.

Impact Statement

The method presented in this study demonstrates considerable potential across a range of applications, especially within the field of machine unlearning. This research is groundbreaking in its investigation of the underlying causes of over-unlearning in non-privacy tasks, with a specific emphasis on fairness and robustness. By providing insights into these challenges, the study seeks to facilitate the development of more advanced and effective unlearning algorithms.

The proposed framework effectively tackles the problem of over-unlearning, offering support to a diverse array of existing machine unlearning algorithms in navigating their respective challenges. However, it is important to note that while the framework is designed to be broadly applicable, its evaluation is constrained by limited resources and the lack of established benchmarks for assessing fairness and robustness in the context of Large Language Model (LLM) unlearning. Consequently, the performance of popular LLM unlearning algorithms, for instance, gradient ascent, have not been evaluated within LLMs, leaving the effectiveness of the framework in this domain unverified. Future research should prioritize exploring the applicability and performance of this framework in LLM-related tasks.

Moreover, it is also essential to clarify that this research does not aim to introduce new arguments advocating for algorithmic fairness, as interventions designed to promote fairness do not always align with the intended societal outcomes. This raises ongoing questions about the suitability of concepts like group fairness DP and EOP metrics for evaluating the equity of decision-making systems. An important avenue for future research involves investigating whether the findings of this study can be applied to other fairness concepts, such as individual fairness. Beyond fairness and robustness, the implications of this work extend to critical areas such as the removal of poisoned data and management of outdated data, which warrants further investigation.

References

Becker, B. and Kohavi, R. Adult. UCI Machine Learning Repository, 1996. 2, 5

Caton, S. and Haas, C. Fairness in machine learning: A survey. *ACM Comput. Surv.*, 56(7):166:1–166:38, 2024. 3

Chen, H., Zhu, T., Yu, X., and Zhou, W. Machine unlearning via null space calibration. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 358–366. ijcai.org, 2024a. 2

Chen, J. and Yang, D. Unlearn what you want to forget: Efficient unlearning for llms. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12041–12052. Association for Computational Linguistics, 2023. 3

Chen, R., Yang, J., Xiong, H., Bai, J., Hu, T., Hao, J., Feng, Y., Zhou, J. T., Wu, J., and Liu, Z. Fast model debias with machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024b. 1, 2, 3, 4, 5, 6

Chhabra, A., Li, P., Mohapatra, P., and Liu, H. "what data benefits my classifier?" enhancing model performance and interpretability through influence-based data selection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 3, 4, 5

Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. 3

Chundawat, V. S., Tarun, A. K., Mandal, M., and Kankanhalli, M. S. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In Williams, B., Chen, Y., and Neville, J. (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 7210–7217. AAAI Press, 2023. 5, 16

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. Fairness through awareness. In Goldwasser, S. (ed.), *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pp. 214–226. ACM, 2012. 2, 3

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 1625–1634. Computer Vision Foundation / IEEE Computer Society, 2018. 3

Goel, S., Prabhu, A., and Kumaraguru, P. Evaluating inexact unlearning requires revisiting forgetting. *CoRR*, abs/2201.06640, 2022. 3, 5, 16

Golatkar, A., Achille, A., and Soatto, S. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9301–9309. Computer Vision Foundation / IEEE, 2020a. 5, 16

Golatkar, A., Achille, A., and Soatto, S. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pp. 383–398. Springer, 2020b. 5, 16

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

Guo, C., Goldstein, T., Hannun, A. Y., and van der Maaten, L. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020. 3, 4

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3315–3323, 2016. 3

Hu, H., Wang, S., Chang, J., Zhong, H., Sun, R., Hao, S., Zhu, H., and Xue, M. A duty to forget, a right to be assured? exposing vulnerabilities in machine unlearning services. In *31st Annual Network and Distributed System Security Symposium, NDSS 2024, San Diego, California, USA, February 26 - March 1, 2024*. The Internet Society, 2024. 2

Hutchinson, B. and Mitchell, M. 50 years of test (un)fairness: Lessons for machine learning. In danah boyd and Morgenstern, J. H. (eds.), *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, pp. 49–58. ACM, 2019. 3

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In Rogers, A., Boyd-Graber, J. L., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 14389–14408. Association for Computational Linguistics, 2023. 4

Jia, J., Liu, J., Ram, P., Yao, Y., Liu, G., Liu, Y., Sharma, P., and Liu, S. Model sparsity can simplify machine unlearning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 2

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. SOUL: unlocking the power of second-order optimization for LLM unlearning. In Al-Onaizan, Y., Bansal, M., and Chen, Y. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 4276–4292. Association for Computational Linguistics, 2024. 6

Kleinberg, J., Ludwig, J., Mullainathan, S., and Rambachan, A. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pp. 22–27, 2018. 3

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. 3, 4, 5, 6, 16

Kurmanji, M., Triantafillou, P., Hayes, J., and Triantafillou, E. Towards unbounded machine unlearning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 3, 4, 5, 16

Kurmanji, M., Triantafillou, E., and Triantafillou, P. Machine unlearning in learned databases: An experimental analysis. *Proceedings of the ACM on Management of Data*, 2(1):1–26, 2024. 1

Li, W., Li, J., de Witt, C. S., Prabhu, A., and Sanyal, A. Delta-influence: Unlearning poisons via influence functions. *arXiv preprint arXiv:2411.13731*, 2024. 1

Liu, Y., Fan, M., Chen, C., Liu, X., Ma, Z., Wang, L., and Ma, J. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pp. 280–289. IEEE, 2022. 1

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to

adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3

Megyeri, I., Hegedüs, I., and Jelasity, M. Adversarial robustness of linear models: regularization and dimensionality. In *27th European Symposium on Artificial Neural Networks, ESANN 2019, Bruges, Belgium, April 24-26, 2019*, 2019. 2, 3

Moro, S., Cortez, P., and Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 62:22–31, 2014. 5

Noever, D. Machine learning suites for online toxicity detection. *CoRR*, abs/1810.01869, 2018. 5

Oesterling, A., Ma, J., Calmon, F., and Lakkaraju, H. Fair machine unlearning: Data removal while mitigating disparities. In *International Conference on Artificial Intelligence and Statistics*, pp. 3736–3744. PMLR, 2024. 1

Qiao, X., Zhang, M., Tang, M., and Wei, E. Efficient online unlearning via hessian-free recollection of individual data statistics. *arXiv preprint arXiv:2404.01712*, 2024. 3

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. MUSE: machine unlearning six-way evaluation for language models. *CoRR*, abs/2407.06460, 2024. 5

Spearman, C. The proof and measurement of association between two things. 1961. 6

Wright, S. Correlation and causation. *Journal of agricultural research*, 20(7):557–585, 1921. 6

Xiong, P., Tegegn, M., Sarin, J. S., Pal, S., and Rubin, J. It is all about data: A survey on the effects of data on adversarial robustness. *ACM Comput. Surv.*, 56(7):174:1–174:41, 2024. 3

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. *CoRR*, abs/2310.10683, 2023. 4

Zhang, B., Dong, Y., Wang, T., and Li, J. Towards certified unlearning for deep neural networks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024a. 3

Zhang, H., Zhang, Z., Zhang, Y., Zhai, Y., Peng, H., Lei, Y., Yu, Y., Wang, H., Liang, B., Gui, L., and Xu, R. Correcting large language model behavior via influence function, 2024b. 6

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868, 2024c. 4

Zhang, Y., Hu, Z., Bai, Y., Wu, J., Wang, Q., and Feng, F. Recommendation unlearning via influence function. *ACM Transactions on Recommender Systems*, 2023. 1, 3

Contents of Appendix

A. Technique Details

We provide a more detailed explanation in §3 to avoid any misleading interpretations, including an explanation of the influence function and quantitative definitions of fairness and robustness.

A.1. Influence Function

The empirical risk minimizer for the training dataset $\mathcal{D} = \{z_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$ is given by $\hat{\theta} = \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^n \ell(z_i;\theta)$. For an empirical risk that is twice-differentiable and strictly convex in the parameter space Θ, we perturb the loss for sample z_j (or alternatively, the training input) by reweighting it with a weight $\epsilon_j \in \mathbb{R}$, as follows:

$$\hat{\theta}(z_j;\epsilon_j) = \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \left(\ell(z_i;\theta) + \epsilon_j \ell(z_j;\theta)\right). \tag{11}$$

① We define the actual change between the empirical risk minimizer trained without sample z_j, denoted by $\hat{\theta}(z_j;-1)$ and the original empirical risk minimizer, denoted by $\hat{\theta}(z_j;0)$ as $\mathcal{I}^*_{\text{param}}(z_j;-1) = \hat{\theta}(z_j;-1) - \hat{\theta}(z_j;0)$. The influence function, using implicit function theory, can effectively approximate the true change in model parameters.

$$\textbf{Parameter Influence:}\quad \mathcal{I}^*_{\text{param}}(z_j;-1) \approx \mathcal{I}_{\text{param}}(z_j;-1) \overset{\text{def}}{=} -\frac{1}{n}\left.\frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} = \frac{1}{n}\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta \ell(z_j;\hat{\theta}). \tag{12}$$

For a function f of interest in the model, such as a utility, fairness and robustness metrics, the actual change in the function f can be expressed as $\mathcal{I}^*(z_j;\epsilon) = f(\hat{\theta}(z_j;-1)) - f(\hat{\theta})$, where $f(\hat{\theta}(z_j;-1))$ denotes the function value on the retraining empirical risk minimizer, and $f(\hat{\theta})$ denotes the function value on the original empirical risk minimizer.

② For the utility metric, we are interested in the loss on the test dataset $\mathcal{T}$, which is given by $\sum_{z\in\mathcal{T}} \ell(z;\hat{\theta})$. By applying the chain rule, we can estimate the actual change in the utility metric of each sample z_j,

$$\begin{aligned}
\textbf{Utility Influence:}\quad \mathcal{I}^*_{\text{util}}(z_j;-1) \approx \mathcal{I}_{\text{util}}(z_j;-1) &\overset{\text{def}}{=} -\left.\frac{d\left(\sum_{z\in\mathcal{T}} \ell(z;\hat{\theta})^\top\right)}{d\epsilon}\right|_{\epsilon-0} \\
&= -\left.\frac{d\left(\sum_{z\in\mathcal{T}} \ell(z;\hat{\theta})^\top\right)}{d\hat{\theta}(z_j;\epsilon_j)}\ \frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} \\
&= -\sum_{z\in\mathcal{T}} \nabla_\theta \ell(z;\hat{\theta})^\top\ \left.\frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} \\
&= \sum_{z\in\mathcal{T}} \nabla_\theta \ell(z;\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}).
\end{aligned} \tag{13}$$

Therefore, $\mathcal{I}_{\text{util}}(z_j;-1)$ reflects the change in loss on the test set $\mathcal{T}$. A negative value of $\mathcal{I}_{\text{util}}(z_j;-1)$ indicates that the retraining empirical risk, obtained without sample z_j, results in a lower test set loss compared to the original empirical risk, meaning that the utility improves when sample z_j is removed.

③ For the fairness metric, we focus on the fairness loss calculated on the test dataset $\mathcal{T}$, which is expressed as $f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})$.

As an example, consider a binary sensitive attribute $g \in 0, 1$ and the predicted probabilities $\hat{y}$.
Demographic Parity (which is also referred to as Statistical Parity) is defined as

$$f_{\text{DP}}(\mathcal{T};\theta) = \left|\mathbb{E}_{\mathcal{T}}[\hat{y} \mid g=0] - \mathbb{E}_{\mathcal{T}}[\hat{y} \mid g=1]\right|,$$

and it holds when the likelihood of receiving a positive predicted probabilities $\hat{y}$ (e.g., being classified as a good credit risk) is independent of the sensitive attribute $g \in 0, 1$. On the other hand, the Equality of Opportunity (EOP) metric is defined by

$$f_{\text{EOP}}(\mathcal{T};\theta) = \left|\mathbb{E}_{\mathcal{T}}[\ell(z;\theta) \mid g=1, y=1] - \mathbb{E}_{\mathcal{T}}[\ell(z;\theta) \mid g=0, y=1]\right|,$$

which ensures that the true positive rates are equal across subgroups, thereby offering equal opportunities for all groups. The fairness of the two metrics increases as their absolute values decrease.

Therefore, by applying the chain rule, we can approximate the change in the fairness metric of each sample z_j.

$$\begin{aligned}\textbf{Fairness Influence:}\quad \mathcal{I}^*_{\text{fair}}(z_j;-1) \approx \mathcal{I}_{\text{fair}}(z_j;-1) &\stackrel{\text{def}}{=} -\left.\frac{d\left(f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})\right)}{d\epsilon}\right|_{\epsilon-0} \\ &= -\left.\frac{d\left(f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})\right)^\top}{d\hat{\theta}(z_j;\epsilon_j)}\ \frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_\theta f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})^\top\ \left.\frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}).\end{aligned} \tag{14}$$

Similarly, $\mathcal{I}_{\text{fair}}(z_j;-1)$ reflects the change in fairness loss on the test set $\mathcal{T}$. A negative value of $\mathcal{I}_{\text{fair}}(z_j;-1)$ indicates that the empirical risk after retraining without sample z_j, leads to a lower fairness loss than the original empirical risk, which suggests that removing sample z_j improves fairness.

④ For the robustness metric, we focus on the loss $\sum_{\tilde{\mathcal{T}}} \ell(\tilde{z};\hat{\theta})^\top$ calculated on the perturbed test dataset $\tilde{\mathcal{T}}$ with adversarial sample $\tilde{z} = z - \gamma \frac{\hat{\theta}^\top z + b}{\hat{\theta}^\top \hat{\theta}} \hat{\theta}$ crafted from test sample $z \in \mathcal{T}$, where $\hat{\theta}$ denotes a linear model, $b \in \mathbb{R}$ is intercept, and $\gamma > 1$ controls the magnitude of perturbation. Since the decision boundary is a hyperplane, adversary can change the prediction by adding minimal perturbations to move each sample orthogonally.

Therefore, by applying the chain rule, we can approximate the change in the robustness metric of each sample z_j.

$$\begin{aligned}\textbf{Robustness Influence:}\quad \mathcal{I}^*_{\text{robust}}(z_j;-1) \approx \mathcal{I}_{\text{robust}}(z_j;-1) &\stackrel{\text{def}}{=} -\left.\frac{d\left(\sum_{\tilde{z}\in\tilde{\mathcal{T}}} \ell(\tilde{z};\hat{\theta})\right)}{d\epsilon}\right|_{\epsilon-0} \\ &= -\left.\frac{d\left(\sum_{\tilde{z}\in\tilde{\mathcal{T}}} \ell(\tilde{z};\hat{\theta})^\top\right)}{d\hat{\theta}(z_j;\epsilon_j)}\ \frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} \\ &= -\sum_{\tilde{z}\in\tilde{\mathcal{T}}} \nabla_\theta \ell(\tilde{z};\hat{\theta})^\top\ \left.\frac{d\hat{\theta}(z_j;\epsilon_j)}{d\epsilon}\right|_{\epsilon=0} \\ &= \sum_{\tilde{z}\in\tilde{\mathcal{T}}} \nabla_\theta \ell(\tilde{z};\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}).\end{aligned} \tag{15}$$

Similarly, $\mathcal{I}_{\text{robust}}(z_j;-1)$ reflects the change in the robustness loss on the perturbed test dataset $\mathcal{T}$. A negative value of $\mathcal{I}_{\text{robust}}(z_j;-1)$ indicates that the empirical risk after retraining without sample z_j, leads to a lower robustness loss than the original empirical risk, which suggests that removing sample z_j improves robustness.

Correspondingly, when we do not explicitly set $\epsilon = -1$, the weighted influence function is given as follows:

- **Weighted Influence Function on Model Parameter:**

$$\mathcal{I}_{\text{param}}(z_j;\epsilon_j) = -\frac{1}{n}\sum_{i\in\mathcal{D}} \epsilon_i^* \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_i;\hat{\theta}) \tag{16}$$

- **Weighted Influence Function on Utility Metric:**

$$\mathcal{I}_{\text{util}}(z_j;\epsilon_j) = -\epsilon_j \sum_{z\in\mathcal{T}} \nabla_\theta \ell(z;\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}). \tag{17}$$

- **Weighted Influence Function on Fairness Metric:**

$$\mathcal{I}_{\text{DP/EOP}}(z_j;\epsilon_j) = -\epsilon_j \nabla_\theta f_{\text{DP/EOP}}(\mathcal{T};\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}). \tag{18}$$

- **Weighted Influence Function on Robustness Metric:**

$$\mathcal{I}_{\text{robust}}(z_j;\epsilon_j) = -\epsilon_j \sum_{\tilde{z}\in\tilde{\mathcal{T}}} \nabla_\theta \ell(\tilde{z};\hat{\theta})^\top \mathbf{H}_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_j;\hat{\theta}). \tag{19}$$

A.2. Analytical Solution of Problem 6

The objective function in Equation (6) contains the squared L^2 norm with inequality constraint equation constrain Eqs. (6b) and (6c). Let $\boldsymbol{\mathcal{I}} = \left(\mathcal{I}\left(z_1; -1\right), \cdots, \mathcal{I}\left(z_n; -1\right)\right)^\top$. The problem in Equation (6) is equivalent to the following problem:

$$\begin{aligned} \text{minimize}_{\boldsymbol{\epsilon}} \quad & -\boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{metric}} + \lambda \|\boldsymbol{\epsilon}\|_2^2 && \text{(20a)} \\ \text{subject to} \quad & -\boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 0 && \text{(20b)} \\ & \boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{metric}} \leq \Delta. && \text{(20c)} \end{aligned}$$

We formulate the Lagrangian to obtain the following unconstrained optimization problem:

$$L(\boldsymbol{\epsilon}, \beta_1, \beta_2) = -\boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{metric}} + \lambda \|\boldsymbol{\epsilon}\|_2^2 - \beta_1 \boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{util}} + \beta_2 (\boldsymbol{\epsilon}^\top \cdot \boldsymbol{\mathcal{I}}_{\text{metric}} - \Delta), \tag{21}$$

where $\beta_1 \geq 0$ and $\beta_2 \geq 0$ are the dual variables corresponding to Equation (20b) and Equation (20c), respectively. Note that $\mathcal{I}_{\text{metric}}\left(z_j; \epsilon_j\right) = -\epsilon_j \mathcal{I}_{\text{metric}}\left(z_j; -1\right)$. The feasible solution $\boldsymbol{\epsilon}$ needs to satisfy the following KKT conditions:

$$\begin{aligned} \nabla_{\boldsymbol{\epsilon}} L(\boldsymbol{\epsilon}, \beta_1, \beta_2) = -\boldsymbol{\mathcal{I}}_{\text{metric}} + 2\lambda\boldsymbol{\epsilon} - \beta_1 \boldsymbol{\mathcal{I}}_{\text{util}} + \beta_2 \boldsymbol{\mathcal{I}}_{\text{metric}} = \mathbf{0}, && \text{(22a)} \\ -\boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 0, && \text{(22b)} \\ \boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{metric}} - \Delta \leq 0, && \text{(22c)} \\ -\beta_1 \boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{util}} = 0 && \text{(22d)} \\ \beta_2 (\boldsymbol{\epsilon}^\top \boldsymbol{\mathcal{I}}_{\text{metric}} - \Delta) = 0 && \text{(22e)} \\ \beta_1, \beta_2 \geq 0 && \text{(22f)} \end{aligned}$$

We have

$$\boldsymbol{\epsilon}^* = \frac{(1-\beta_2) \cdot \boldsymbol{\mathcal{I}}_{\text{metric}} + \beta_1 \cdot \boldsymbol{\mathcal{I}}_{\text{util}}}{2\lambda}. \tag{23}$$

In the following, we consider four cases:

Case 1: For $\beta_1 = 0,\ \beta_2 = 0$, we obtain:
Case 1 condition: When $0 \leq \boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 2\lambda\Delta$ and $|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 < 2\lambda\Delta$, the analytical solution is given as follows:

$$\begin{aligned} \boldsymbol{\epsilon}^* &= \frac{(1-\beta_2) \cdot \boldsymbol{\mathcal{I}}_{\text{metric}} + \beta_1 \cdot \boldsymbol{\mathcal{I}}_{\text{util}}}{2\lambda} \\ &= \frac{\boldsymbol{\mathcal{I}}_{\text{metric}}}{2\lambda}. \end{aligned} \tag{24}$$

Case 2: For $\beta_1 = 0,\ \beta_2 = 1 - \frac{2\lambda\Delta}{|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2} \geq 0$, we obtain:
Case 2 Condition: $|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 - 2\lambda\Delta \geq 0,\ \boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \geq 0$, the analytical solution is given as follows:

$$\begin{aligned} \boldsymbol{\epsilon}^* &= \frac{(1-\beta_2)\boldsymbol{\mathcal{I}}_{\text{metric}} + \beta_1 \boldsymbol{\mathcal{I}}_{\text{util}}}{2\lambda} \\ &= \frac{\Delta}{|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2} \cdot \boldsymbol{\mathcal{I}}_{\text{metric}}. \end{aligned} \tag{25}$$

Case 3: For $\beta_1 = -\frac{\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}}}{|\mathcal{I}_{\text{util}}|^2} \geq 0,\ \beta_2 = 0$, we obtain:
Case 3 Condition: $\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}} \leq 0,\ (\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}})^2 \geq |\boldsymbol{\mathcal{I}}_{\text{util}}|^2 (|\boldsymbol{\mathcal{I}}_{\text{metric}}|^2 - 2\lambda\Delta)$, the analytical solution is:

$$\begin{aligned} \boldsymbol{\epsilon}^* &= \frac{(1-\beta_2)\boldsymbol{\mathcal{I}}_{\text{metric}} + \beta_1 \boldsymbol{\mathcal{I}}_{\text{util}}}{2\lambda} \\ &= \frac{\boldsymbol{\mathcal{I}}_{\text{metric}} - \frac{\boldsymbol{\mathcal{I}}_{\text{metric}}^\top \boldsymbol{\mathcal{I}}_{\text{util}}}{|\mathcal{I}_{\text{util}}|^2} \cdot \boldsymbol{\mathcal{I}}_{\text{util}}}{2\lambda}. \end{aligned} \tag{26}$$

Case 4: For $\beta_1 = -\frac{2\lambda\Delta\mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}}}{\left(|\mathcal{I}_{\text{metric}}|^2|\mathcal{I}_{\text{util}}|^2-(\mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}})^2\right)} \geq 0,\ \beta_2 = 1 - \frac{2\lambda\Delta|\mathcal{I}_{\text{util}}|^2}{|\mathcal{I}_{\text{metric}}|^2|\mathcal{I}_{\text{util}}|^2-(\mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}})^2} \geq 0$, we obtain:

Case 4 Condition: $\mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}} \leq 0,\ (\mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}})^2 \leq |\mathcal{I}_{\text{util}}|^2(|\mathcal{I}_{\text{metric}}|^2 - 2\lambda\Delta)$, the analytical solution is:

$$\begin{aligned}\boldsymbol{\epsilon}^* &= \frac{(1-\beta_2)\cdot\mathcal{I}_{\text{metric}} + \beta_1\cdot\mathcal{I}_{\text{util}}}{2\lambda} \\ &= \frac{\Delta\Big(|\mathcal{I}_{\text{util}}|^2\cdot\mathcal{I}_{\text{metric}} - \mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}}\cdot\mathcal{I}_{\text{util}}\Big)}{|\mathcal{I}_{\text{metric}}|^2|\mathcal{I}_{\text{util}}|^2 - (\mathcal{I}_{\text{metric}}^\top\mathcal{I}_{\text{util}})^2}.\end{aligned} \tag{27}$$

A.3. Weighted Machine Unlearning Algorithms

In this paper, we follow the experimental repository in (Kurmanji et al., 2023) with the following **nine unlearning algorithms**: Gradient Ascent **(GA)**, Fine-Tuning **(FT)**, Influence Function **(IF)** (Koh & Liang, 2017), Fisher Forgetting **(Fisher)** (Golatkar et al., 2020a) and NTK Forgetting **(NTK)** (Golatkar et al., 2020b), Teacher-Student Formulation **(SCRUB)** (Kurmanji et al., 2023) and **(Bad-T)** (Chundawat et al., 2023), Catastrophic Forgetting-k **(CF-k)** and Exact Unlearning-k **(EU-k)** (Goel et al., 2022), along with their **S**oft-**W**eighted **(SW-)** versions. Specifically, for training sample $z_j \in \mathcal{D}$, we define ϵ_r as the weight of the remaining data $z_r \in \mathcal{D}_r$ and ϵ_f as the weight of the forgetting data $z_f \in \mathcal{D}_f$. The following are the technical details of the different machine unlearning methods:

① Gradient Update Methods: **GA** and **FT**.
GA updates the model by adjusting the parameters according to the negative of the update direction computed from the forgetting dataset, thereby maximizing the loss on the forgetting data z_f,

$$\theta_{t+1}(z_f;-1) = \theta_t(z_f;-1) + \eta_t\nabla_\theta\ell(z_f;\theta_t(z_f;-1)), \tag{28}$$

FT updates the model by adjusting the parameters based on the gradient of the loss function computed over the remaining dataset, optimizing the model to retain knowledge while minimizing the loss on the remaining data z_r.

$$\theta_{t+1}(z_r;-1) = \theta_t(z_r;-1) - \eta_t\nabla_\theta\ell(z_r;\theta_t(z_r;-1)), \tag{29}$$

Therefore, the soft-weighted $\textbf{GA}_{\textbf{FT}}$ can be updated in a manner analogous to weighted gradients update.

$$\theta_{t+1}(z_j;\epsilon_j) = \theta_t(z_j;\epsilon_j) + \epsilon_j\cdot\eta_t\nabla_\theta\ell(z_j;\theta_t(z_j;\epsilon_j)), \tag{30}$$

② Closed-form Update Mehtods: **IF**, **Fisher**, and **NTK**.
IF performs a closed-form Newton step to estimate the empirical risk minimizer trained without forgetting data z_f.

$$\hat{\theta}(z_f;-1) - \hat{\theta}(z_f;0) \approx \frac{1}{n}\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_f;\hat{\theta}), \tag{31}$$

The **Fisher** and **NTK** both require Hessian approximation. **Fisher** approximates the Hessian using the Fisher Information Matrix. **NTK** provides a neural tangent kernel (NTK)-based approximation of the training process and uses it to estimate the updated network parameters after forgetting. Formally, **NTK**, **Fisher**, and **IF** are similar and can be interchangeable in special cases. For instance, in the case of an L^2 loss, the NTK model **NTK** coincides with the Fisher Matrix.

Therefore, **IF**, **NTK**, and **Fisher** can all be weighted in a manner analogous to the following soft-weighted **IF**,

$$\hat{\theta}(z_f;\epsilon_f) - \hat{\theta}(z_f;0) \approx -\epsilon_f\cdot\frac{1}{n}\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_f;\hat{\theta}), \tag{32}$$

③ Teacher-Student (T-S) Framework Methods: **SCRUB** and **Bad-T**.
SCRUB considers two sets of teachers: the original model as the "teacher" and the student model. The student is encouraged to stay close to the teacher on the remaining dataset and move away from it on the forgetting dataset. **SCRUB** aims to optimize the following objective function:

$$\min_\theta \frac{\alpha}{|\mathcal{D}_r|}\sum_{z_r\in\mathcal{D}_r} d\left(z_r;\theta(z_f;-1)\right) + \frac{\gamma}{|\mathcal{D}_r|}\sum_{z_r\in\mathcal{D}_r} f\left(z_r;\theta(z_f;-1)\right) - \frac{1}{|\mathcal{D}_f|}\sum_{z_f\in\mathcal{D}_f} d\left(z_f;\theta(z_f;-1)\right). \tag{33}$$

where $d\left(z;\theta(z_f;-1)\right) = D_{\mathrm{KL}}\left(p\left(f\left(z;\theta(z_f;0)\right)\right)\|p\left(f\left(z;\theta(z_f;-1)\right)\right)\right)$ is the KL-divergence between the student and teacher output distributions (softmax probabilities) for the sample z_j, with hyperparameters α and γ. Specifically, in Equation (33), the third term involves maximizing the distance between the student and teacher on the forget dataset $\mathcal{D}_f$. The first term is analogous to the third but encourages the student to remain proximal to the teacher on remaining dataset $\mathcal{D}_r$. Finally, the second term optimizes for the loss on the remaining dataset $\mathcal{D}_r$,

The optimization process alternates between the remaining dataset (*the min-step*) and forgetting dataset (*the max-step*),

$$\textit{the min-step: } \theta(z_r;-1) \leftarrow \theta(z_r;-1) + \eta\nabla_\theta \frac{1}{|\mathcal{D}_r|}\sum_{z_r\in\mathcal{D}_r} d\left(z_r;\theta(z_r;-1)\right). \tag{34}$$

$$\textit{the max-step: } \theta(z_f;-1) \leftarrow \theta(z_f;-1) + \eta\nabla_{\theta(z_f;-1)} \frac{1}{|b|}\sum_{z_f\in b} d\left(z_f;\theta(z_f;-1)\right) + \gamma f\left(x_r;\theta(z_f;-1)\right). \tag{35}$$

Considering soft-weighted **SCRUB**, the objective function in Equation (33) takes the following form:

$$\min_\theta \frac{\alpha}{|\mathcal{D}_r|}\sum_{z_r\in\mathcal{D}_r} \epsilon_r \cdot d\left(z_r;\theta(z_f;\epsilon)\right) + \frac{\gamma}{|\mathcal{D}_r|}\sum_{z_r\in\mathcal{D}_r} \epsilon_r \cdot f\left(z_r;\theta(z_f;\epsilon_r)\right) + \frac{1}{|\mathcal{D}_f|}\sum_{z_f\in\mathcal{D}_f} \epsilon_f \cdot d\left(z_f;\theta(z_f;\epsilon_f)\right), \tag{36}$$

with following weighted optimization process:

$$\textit{the min-step: } \theta(z_r;\epsilon_r) \leftarrow \theta(z_r;\epsilon_r) + \epsilon_r\eta\nabla_\theta \frac{1}{|\mathcal{D}_r|}\sum_{z_r\in\mathcal{D}_r} d\left(z_r;\theta(z_r;\epsilon_r)\right). \tag{37}$$

$$\textit{the max-step: } \theta(z_f;\epsilon_f) \leftarrow \theta(z_f;\epsilon_f) + \eta\nabla_{\theta(z_f;\epsilon_f)} \frac{1}{|b|}\sum_{z_f\in b} d\left(z_f;\theta(z_f;\epsilon_f)\right) + \gamma f\left(x_r;\theta(z_f;\epsilon_f\right). \tag{38}$$

Bad-T considers two sets of teachers: the original model as the good teacher and random models as the bad teacher. The student is encouraged to follow the good teacher on the remaining dataset and the bad teacher on the forgetting dataset.

$$\min_\theta (1-y_f) * \mathcal{KL}\left(T_s(x)\|S(x)\right) + y_f * \left(\mathcal{KL}\left(T_d(x)\|S(x)\right)\right), \tag{39}$$

where $T_s(x)$ represents the competent/smart teacher, and $T_d(x)$ is the incompetent/dumb teacher, with y_f being the label of forgetting dataset and x the sample. The optimization process also alternates between the remaining and forgetting datasets. Due to the similar form of **Bad-T** and **SCRUB**, we omit the formulation for soft-weighted **Bad-T**.

④ Freezing the layers of the neural network Methods: **CF-k** and **EU-k**. The **CF-k** (Catastrophic Forgetting-k) and **EU-k** (Exact Unlearning-k) methodologies are specifically designed for neural network applications. These approaches operate by first freezing a predefined number of initial layers in the neural architecture, then subsequently either: Fine-tuning the final k layers using the remaining dataset (**CF-k**), or Performing complete retraining of the final k layers with the remaining dataset (**EU-k**). For implementation convenience, we constrain parameter updates exclusively to the final layer. Consequently, the soft-weighted **CF-k** and **EU-k** adopt the same mathematical formulation presented in Equation (30).

We observe that the overwhelming majority of unlearning algorithms (with the exception of closed-form update methodologies) are predominantly grounded in gradient ascent (GA) and fine-tuning (FT) mechanisms. This analysis delineates their operational specifics through three principal implementation paradigms under fixed epoch constraints:

- **GA$_{\mathbf{FT}}$** employs a two-phase approach, first applying **GA** on the forgetting dataset for half the total epochs, then **FT** on the remaining dataset for the latter half.
- **SCRUB** and **Bad-T** implement an alternating optimization strategy, interleaving gradient ascent and descent steps using their respective objective functions throughout the training process.
- **CF-k** conducts **FT** on remaining dataset across all epochs, contrasting with **EU-k**'s complete model reinitialization and retraining model.

B. Experiment Details

B.1. Hardware, Software and Source Code

The experiments were conducted on an NVIDIA GeForce RTX 4090. The code was implemented in PyTorch 2.0.0 and utilizes the CUDA Toolkit version 11.8. Tests were performed on an AMD EPYC 7763 CPU @1.50GHz with 64 cores, running Ubuntu 20.04.6 LTS. Our code is available at Soft Weighted Machine Unlearning. We provide the implementation of soft weights and will organize a complete framework that supports a broad range of unlearning methods.

B.2. Datasets

Adult Dataset: Income prediction dataset with 45,222 samples. Divided into 30,162 training, 7,530 validation, and 7,530 test samples. Gender (male/female) serves as the sensitive attribute for fairness evaluation.

Bank Dataset: Bank client subscription analysis dataset containing 30,488 entries. Training set (18,292), validation/test sets (6,098 each). Gender (male/female) is designated as the sensitive attribute.

CelebA Dataset: Facial image dataset comprising 104,163 samples, split into 62,497 training, validation/test sets (20,833 each). Gender (male/female) serves as the sensitive attribute for fairness evaluation.

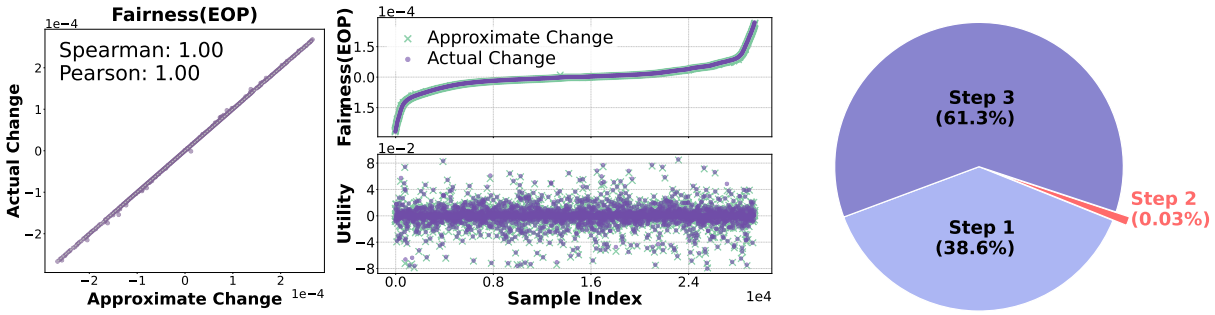
Jigsaw Toxicity Dataset: Toxic comment detection corpus with 30,000 social media texts. Training data (18,000), validation/test sets (6,000 each). Ethnicity (Black/Other) serves as the sensitive attribute for fairness evaluation.

B.3. Additional Experiments

This section presents additional experiments for §5, including (i) the time distribution for each step in the soft-weighted machine unlearning framework. (ii) the actual changes in the fairness EOP metric and the estimated changes in influence values, (iii) the performance of different unlearning algorithms on EOP metric.

First, Figure 7 (a) illustrates that influence functions provide an effective estimation of the true leave-one-out model changes for Equality of Opportunity (EOP) metrics. Importantly, similar to the Demographic Parity metric, selectively removing detrimental samples does not guarantee improvements in the Utility metric.

Second, Figure 7 (b) shows that the time overhead for weight acquisition accounts for only 0.03% of the total **`IF`** unlearning procedure in Step 2. It is noteworthy that the hard weighting framework also necessitates executing Step 1 for sample influence estimation to identify the forgetting dataset, as well as Step 3 to implement the unlearning algorithm. In contrast, the soft weighted machine unlearning framework incurs a smaller overhead in Step 2 to obtain a set of optimal weights while achieving superior performance in Step 3. This underscores the scalability of the soft weighted machine unlearning framework and highlights its strong potential for real-world deployment scenarios.



(a) Actual EOP Change vs. Influence EOP Change

(b) Time Cost in Each Step

Figure 7. **(a)** The leave-one-out influence of all training samples on the EOP metric. The first plot evaluates the correlation coefficient, indicating an effective approximation of the influence function **(Left)**. The second plot ranks the samples based on their actual EOP metric from smallest to largest, illustrating the utility of each sample, and suggesting that removing the detrimental samples does not necessarily increase utility **(Middle)**. **(b)** We use **`IF`** as the unlearning method to update the model. Step 1 (evaluation) and Step 3 (removal) are common to both hard and soft weighting. Therefore, soft weighting requires only minimal additional time in Step 2 **(Right)**.

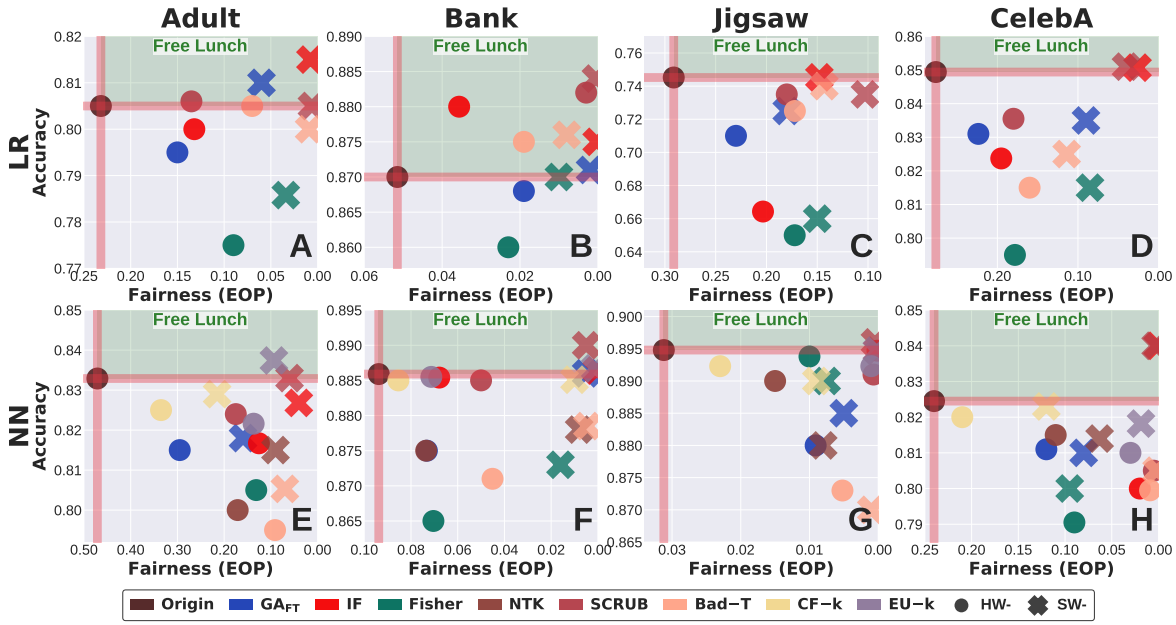


Figure 8. **Performance on EOP Metric.** Different colors represent various unlearning algorithms: ● for the hard-weighted scheme and ✖ for the soft-weighted scheme. **The First Two Rows** (LR, NN) evaluate utility and fairness metrics, while **The Last Two Rows** (LR, NN) evaluate utility and robustness metrics across datasets. **The Green Region** highlights **Free Lunch** cases where unlearning algorithms improve both target task performance and utility compared to the original model. The soft-weighted scheme outperforms the hard-weighted scheme by enhancing task performance and utility, even achieving free lunch in part of unlearning algorithms' results.

Finally, Figure 8 presents a series of additional experimental evaluations focused on the EOP (Error of Prediction) metric. These evaluations clearly demonstrate the superiority of the soft-weighted machine unlearning framework compared to traditional approaches in various scenarios and datasets. This evidence underscores the framework's potential to address the challenges associated with machine unlearning, making it a promising solution for future applications in this field.