Beyond Binary Erasure: Soft-Weighted Unlearning for Fairness and Robustness

Xinbao Qiao[1], **Ningning Ding**[2], **Yushi Cheng**[1], **Meng Zhang**[*1]

[1]Zhejiang University, China
[2]The Hong Kong University of Science and Technology (Guangzhou), China
xinbaoqiao@zju.edu.cn

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 a binary data removal framework designed for privacy-driven motivation, even when repurposed for fairness or robustness improvements. This leads to significant utility loss, a phenomenon known as "over-unlearning". While over-unlearning has been largely described in many studies as primarily causing utility degradation, we investigate deeper insights in this work through counterfactual leave-one-out analysis. Based on insights, we introduce a soft weighting strategy that assigns tailored weights to each sample by solving a convex quadratic programming problem analytically, which enables fine-grained model adjustments to address the over-unlearning. We demonstrate that the proposed soft-weighted scheme 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 algorithm as an effective correction solution.

1 Introduction

Modern machine learning (ML) models benefit greatly from the quantity and quality of the training data they are built upon. As a recent advancement, machine unlearning, originally conceived as a privacy-preserving mechanism to comply with data protection regulations' "right to be forgotten" by allowing users to remove personal data from models, has significantly broadened its scope. Beyond its privacy-oriented motivation (Wang et al. 2025a), 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; Wang et al. 2025b; Li et al. 2024; Kurmanji, Triantafillou, and Triantafillou 2024). These provide a fast way to edit a trained model without prohibitively expensive process of retraining from scratch, catalyzing a paradigm shift in methodologies to address critical challenges beyond privacy concerns.

[*]Corresponding Author

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 a hard-weighted unlearning framework in this paper, characterized by the complete elimination of undesired data influences. This framework, 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.

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 leads to suboptimal outcomes, particularly when dealing with complex objectives. We illustrate it concretely as evidence in Figure 1, where we trained a model on the CelebA face recognition dataset (Liu et al. 2015) (See Appendix B.4 for the results of other datasets) and analyzed the performance of leave-one-out (LOO) 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); robustness, quantified by loss on perturbed datasets (Megyeri, Hegedüs, and Jelasity 2019); and utility (generalization), determined by the loss on the test set. These results allowed us to uncover the underlying causes of over-unlearning:
(1) Detrimental samples (negative score in fairness/robustness) removal does not necessarily lead to utility improvements. The red-highlighted samples in Figure 1 indicate that removing the most biased (vulnerable) samples (top figure) does not lead to accuracy gains (bottom figure).
(2) Borderline samples (score near 0 in fairness/robustness) are treated equivalently to highly detrimental samples by unlearning algorithms. Borderline samples in Figure 1 do not significantly affect model bias or vulnerability. However, in hard-weighted frameworks, such as gradient ascent algorithms (Jia et al. 2023), these samples are treated uniformly in an attempt to remove the most biased (vulnerable) ones, which can lead to excessive unlearning of borderline samples.

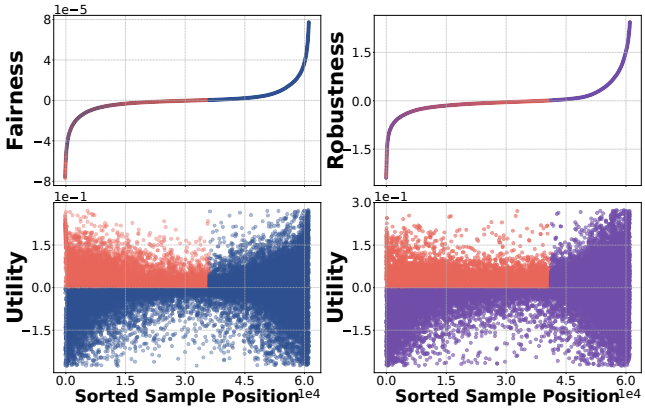

Figure 1: Actual Changes in Utility and Fairness/Robustness on CelebA for each sample's leave-one-out model. The X-axis represents represents sample indices sorted by fairness (left) or robustness (right) metrics. The Y-axis for Fairness (Robustness) displays changes in demographic parity (adversarial loss) on the test set, 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. Scatter points marked in Red indicate samples where fairness or robustness improves, but utility declines.

For instance, it may cause borderline samples to be flipped to unprivileged groups, resulting in opposite biases.

(3) Most of the samples with negative influence are maintained in remain dataset. Approximately 70-75% of samples with values below 0 in Figure 1 exacerbate bias and vulnerability. However, existing algorithms for improving fairness and robustness (Chen et al. 2024b) remove a small subset (e.g., 20%) and struggle to support further deletions.

In this paper, we take the first step in addressing the challenge of over-unlearning when applying machine unlearning to improve fairness and robustness. 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. The key difference lies in our departure from the binary removal scheme inherited from privacy-driven motivations, instead granularly modeling an optimization that allocates weights to each data. We theoretically and empirically demonstrate enhanced performance on target tasks while improving utility. Our main contributions are:

- We introduce the weighted machine unlearning framework in §4.1, a refined solution to address the over-unlearning challenge, with the weights through solving a convex quadratic programming problem in §4.2.
- We theoretically demonstrate in §4.3 that the problem is feasible by the existence of a weight vector enabling fairness or robustness improvement while preserving utility.
- 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, Prabhu, and Kumaraguru 2022; Chen and Yang 2023), is claimed to address challenges beyond its original privacy concerns (Ding et al. 2025; Cui, Ding, and Cheung 2025), 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 and 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 the forgetting subset and subsequently unlearning undesired data. Furthermore, despite existing work exploring the fairness and robustness of unlearning methods, e.g., (Oesterling et al. 2024; Chen et al. 2024c; Tran and Woo 2025; Sheng, Bao, and Ge 2024; Dige et al. 2024), these approaches focus on enhancing the fairness and robustness of unlearning algorithms themselves, rather than leveraging unlearning for fairness (Chen et al. 2024b) and robustness (Huang et al. 2025) tasks.

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 (Wen et al. 2025). 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 and Mitchell 2019). As pointed out in (Caton and Haas 2024), fairness notions are often incompatible and have limitations, with no universal metric or guideline for measuring fairness (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, Price, and Srebro 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, Hegedüs, and Jelasity 2019) targeting a linear model, which can be extended to methods such as FGSM (Goodfellow, Shlens, and Szegedy 2015) and PGD (Madry et al. 2018). We quantify robustness as performance under adversarial attacks, referred to as perturbed accuracy, which is distinguished from utility known as standard 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 (ER) 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 ER that is twice-differentiable and strictly convex[1] in parameter space Θ, we slightly perturb the sample z_j by reweighting it with weight $\epsilon_j\in\mathbb{R}$, leading to:

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

Let $\epsilon_j=-1$ give $\hat{\theta}(z_j;-1)$, the ER minimizer trained without sample z_j, and clearly, $\hat{\theta}=\hat{\theta}(z_j;0)$. Hence, using influence function (Koh and Liang 2017) can efficiently capture model parameter change through a 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, e.g., utility (generalization), fairness or robustness metrics, 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: $\mathcal{I}_{\text{util}}(z_j;-1)=\sum_{z\in\mathcal{T}}\nabla_\theta\ell(z;\hat{\theta})^\top\mathbf{H}_{\hat{\theta}}^{-1}\nabla_\theta\ell(z_j;\hat{\theta})$, which reflects the loss change in the validation set $\mathcal{T}$, where a negative value indicates better generalization in model trained without z_j, while a positive one suggests z_j is detrimental.

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 instantiated by the fairness metrics in the validation 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 $f_{\text{EOP}}(\mathcal{T};\hat{\theta})=\left|\mathbb{E}_{\mathcal{T}}[\ell(z;\theta)\mid g=1,y=1]-\mathbb{E}_{\mathcal{T}}[\ell(z;\theta)\mid g=0,y=1]\right|$. Moreover, we adopt common surrogate functions (Chhabra et al. 2024) to make the above metrics differentiable. Similarly, a negative value indicates a lower $f_{\text{fair}}(\mathcal{T};\theta)$ on a model trained without sample z_j, implying 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 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, adversaries can change the prediction by adding perturbations to move each sample orthogonally. A negative value of $\mathcal{I}_{\text{robust}}(z_j;-1)$ indicates a lower $f_{\text{robust}}(\mathcal{T};\theta)$ on a model trained without z_j, implying improvement in robustness; conversely, a positive one indicates detrimental effect.

Previous machine unlearning methods (Chen et al. 2024b; Huang et al. 2025) for improving fairness and robustness typically follow two steps: (i) compute $\mathcal{I}_{\text{fair}}(z_j;-1)$ or $\mathcal{I}_{\text{robust}}(z_j;-1)$ to estimate each sample's impact and select a forgetting set based on these values; (ii) unlearn the forgetting set using Equation (2). However, as shown in Figure 1, such hard removal may cause significant utility degradation. To address this, we introduce a fine-grained unlearning framework to improve the target metric without utility loss in §4.

[1]While theoretical analysis of influence functions is intractable for non-convex models, they remain effective in practice. Following previous influence-function-based works (Guo et al. 2020), we compute the influence of each sample solely on the last layer to ensure convexity. (Kirichenko, Izmailov, and Wilson 2023) also shows that last-layer retraining can match state-of-the-art group robustness (see also (Izmailov et al. 2022; Qiu et al. 2023; Rudner et al. 2024) for follow-up works) and fairness (Schrouff et al. 2024; Welfert, Stromberg, and Sankar 2024; Tran and Woo 2025). Consequently, this allows our method to generalize to arbitrary neural networks.

4 Methodology

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 unlearning framework in §4.3. We then highlight its broad applicability with diverse unlearning paradigms in §4.4.

4.1 Step 1: Weighted Influence Function

As in prior machine unlearning studies (Chen et al. 2024b; Zhang et al. 2023), we begin by estimating each sample's influence on model fairness and robustness. Distinct from these approaches, we further assess the utility influence of each sample. Moreover, rather than assigning a binary weight $\epsilon=$ -1 (forgetting set) or 0 (remaining set) in Equation (1), we introduce a weighted influence function defined as follows:

- **Weighted Influence Function on the 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 the 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 the 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 above functions, $\mathcal{I}_{(\cdot)}(z_j;\epsilon_j)=-\epsilon_j\mathcal{I}_{(\cdot)}(z_j;-1)$, where ϵ_j is not binary ($\epsilon=-1$ or 0), but can be optimized based on $\mathcal{I}_{(\cdot)}(z_j;-1)$.

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 a 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. Based on this problem formulation, one can either employ a linear solver (e.g., Gurobi (2024)) or analytically derive a closed-form solution to obtain the optimal set of weights.

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

Where $\boldsymbol{\mathcal{I}}_{(\cdot)}=\big(\mathcal{I}_{(\cdot)}(z_1;-1),\cdots,\mathcal{I}_{(\cdot)}(z_n;-1)\big)^\top$ for samples $\{z_i\}_{i=1}^n$. See Appendix A.2 for conditions (cond.) and 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:

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

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 last 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 and Liang 2017). Nevertheless, for generative models, the strategies outlined in the footnote of §3 may not be as effective. 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 $\hat{\theta}(z_j;\epsilon_j^*)-\hat{\theta}(z_j;0)\approx-\frac{\epsilon_j^*}{n}\sigma\mathbf{I}\cdot\nabla_\theta\ell(z_j;\hat{\theta})$. The constant σ/n can be interpreted as step size η and estimate $\hat{\theta}(z_j;\epsilon_j^*)$ through multiple update steps indexed by t,

$$\theta_{t+1}(z_j;\epsilon_j^*)-\theta_t(z_j;0)=-\epsilon_j^*\cdot\eta_t\nabla_\theta\ell(z_j;\hat{\theta}). \tag{9}$$

As can be seen in Equation (9), this establishes a connection between first-order and second-order unlearning algorithms. This suggests that the soft-weighted scheme can be naturally extended to other gradient-based unlearning methods, e.g., fine-tuning and gradient ascent algorithms, which are currently cutting-edge methods in both LLM unlearning (Jang et al. 2023; Yao, Xu, and Liu 2023; Zhang et al. 2024c) and non-LLM unlearning (Kurmanji et al. 2023). We empirically demonstrate that soft-weighted scheme can also be effectively applied to other heuristic unlearning algorithms, such as Fisher (Golatkar, Achille, and Soatto 2020a) or Teacher-Student Formulation (Kurmanji et al. 2023). Please refer to Appendix A.3 for details of the soft-weighted version of machine unlearning algorithms.

To justify the effectiveness, we conduct a theoretical analysis showing that soft reweighting improves model fairness and robustness while preserving utility, i.e., optimizing the weights ϵ_i to ensure fairness $\sum_{i=1}^n\mathcal{I}_{\text{fair}}(z_i;\epsilon_i)$ or robustness $\sum_{i=1}^n\mathcal{I}_{\text{robust}}(z_i;\epsilon_i)$, without cost of utility $\sum_{i=1}^n\mathcal{I}_{\text{util}}(z_i;\epsilon_i)$:

Theorem 1 (**Fairness–Utility Pareto Improvement**). *Let $\mathbf{G}=(\nabla_\theta\ell(z_1;\hat{\theta}),\cdots,\nabla_\theta\ell(z_n;\hat{\theta}))^\top$ be the Jacobian matrix of per-sample loss gradients on the training samples $\{z_i\}_{i=1}^n$. If all of the following three conditions are satisfied,*

1. ***Linear Independence.*** *The gradient of the fairness metric, $\nabla_\theta f_{\text{fair}}(\mathcal{T};\hat{\theta})\in\mathbb{R}^d$, and the gradient of the utility metric, $\sum_{z\in\mathcal{T}}\nabla_\theta\ell(z;\hat{\theta})\in\mathbb{R}^d$, are linearly independent in $\mathbb{R}^d$.*
2. ***Influence Function Validity.*** *Standard convexity conditions for influence-function analysis are satisfied.*
3. ***Full Column Rank.*** *The rank of $\mathbf{G}$ $\operatorname{rank}(\mathbf{G})=d$.*

then there exists weights $\boldsymbol{\epsilon}=(\epsilon_1,\ldots,\epsilon_n)^\top\in\mathbb{R}^n$ such that

$$\sum_{i=1}^n\mathcal{I}_{\text{fair}}(z_i;\epsilon_i)<0\quad\textit{and}\quad\sum_{i=1}^n\mathcal{I}_{\text{util}}(z_i;\epsilon_i)<0.$$

Justification. Condition 2 in Theorem 1 holds automatically when θ denotes the parameters of a convex model or network's last layer, even if the model is non-convex. Condition 3 almost surely holds by computing sample influence on a convex model or network's last layer, provided that $n\gg d$ and the samples are sufficiently diverse.

See Appendix A.4 for the detailed proof. Notably, the target task is not restricted to fairness; it also encompasses robustness and can extend to other criteria. In this paper, we focus on fairness and robustness tasks and present results for both. We now formally state the robustness below.

Corollary 2 (**Robustness–Utility Pareto Improvement**). *If all of the conditions of Theorem 1 hold except that condition (i) is replaced by linear independence between the robustness gradient $\sum_{\tilde{z}\in\tilde{\mathcal{T}}}\nabla_\theta\ell(\tilde{z};\hat{\theta})$ and the utility gradient $\sum_{z\in\mathcal{T}}\nabla_\theta\ell(z;\hat{\theta})$, then there exist weights $\boldsymbol{\epsilon}$ such that*

$$\sum_{i=1}^n\mathcal{I}_{\text{robust}}(z_i;\epsilon_i)<0\quad\textit{and}\quad\sum_{i=1}^n\mathcal{I}_{\text{util}}(z_i;\epsilon_i)<0.$$

Remark 1. *The conditions in Theorem 1 are mild and generally hold for any convex model or the last layer of non-convex model. In contrast to privacy-driven unlearning, which requires entirely removing a sample's influence from the overall model, our objective is less stringent: since we do not seek to fully erase the sample's effect, it suffices to estimate influence and update only on the last layer, as justified in §3's footnote.*

Remark 2. *Theorem 1 implies that evaluating per-sample influence through influence functions ensures the feasibility of the aforementioned optimization problem. This desirable outcome aligns with the concept of the Pareto Improvement. As a result, when followed by appropriate reweighting during the unlearning process, this approach allows downstream tasks (e.g., fairness or robustness) to benefit from machine unlearning algorithms without degrading utility.*

Algorithm 1: Soft-Weighted Unlearning Framework

Input: Model $\hat{\theta}$, Training Dataset $\mathcal{D}$, Validation Dataset $\mathcal{T}$, Adversarial Samples $\tilde{z} \in \tilde{\mathcal{T}}$

 # Step 1: Influence Evaluation.

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

  Evaluate influence of z_i on validation set;

  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

 # Step 2: Weights Optimization.

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

 # Step 3: Model Correction.

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

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

end

Output: θ

4.4 Soft-Weighted Unlearning Framework

To further explore the 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 ascent algorithm represents another 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. In contrast, the soft scheme aligns with our intuition: mitigating highly detrimental effects while amplifying beneficial ones.

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, e.g., bias mitigation and robustness enhancement. This framework introduces a finer-grain approach by assigning differentiated weights to samples based on their contributions to the model's objective. 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 utilize Equation (4) and Equation (5) to quantify the impact of each training sample on the fairness and robustness of the model, as measured on the validation set. In contrast to prior work (Chen et al. 2024b), our approach incorporates Equation (3) to assess the utility contribution of training samples on the validation set.

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

In this section, we conduct experiments that assess the performance of the soft-weighted unlearning framework in mitigating bias and improving robustness. We first estimate the influence of each training sample using a validation set before unlearning, and then evaluate the utility, robustness, and fairness metrics on testing set after the unlearning process is completed. The results are averaged over five random seeds.

Metrics: For the fairness task, we directly adopt *DP* and *EOP* on the test set as evaluation metrics, while for robustness, we use the *adversarial loss* on the test set. For utility, we measure *test accuracy*. These metrics directly reflect how the unlearning algorithm improves performance on the corresponding tasks. Importantly, unlearning is not to forget specific samples in this paper, but rather to leverage unlearning to enhance fairness and robustness, i.e., unlearning serves as a means, not an end. Therefore, traditional unlearning evaluation metrics, e.g., forgetting set accuracy or membership inference attack (Qiao et al. 2025), are not sufficient to indicate improvement in fairness or robustness. As illustrated in Figure 1, even if a sample is proven to be completely forgotten (i.e., removed from retrained models), this does not necessarily lead to better corresponding tasks and utility.

Model: 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, as well as **ResNet-18** and **ResNet-50** (He et al. 2016). During the unlearning process, similar to (Feldman and Zhang 2020), we compute influence values by treating the last layer of the neural network or ResNet as a convex surrogate of the full non-convex model. When applying the unlearning algorithm, we similarly restrict updates to this last layer only.

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 **five real-world datasets**, including two tabular datasets **UCI Adult** (Becker and Kohavi 1996), **Bank** (Moro, Cortez, and Rita 2014), one visual human face dataset **CelebA** (Liu et al. 2015), one textual dataset **Jigsaw Toxicity** (Noever 2018). These four datasets are widely adopted benchmarks for evaluating fairness and robustness (Chhabra et al. 2024; Wang et al. 2025c). In addition, we evaluate fairness on **CelebA** with ResNet-18 and robustness on **CIFAR-100** with ResNet-50 (Krizhevsky, Hinton et al. 2009). See details of datasets in Appendix B.2. We defer EOP to the Appendix B.

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 (Shi et al. 2024), we denote the combinations as $\texttt{GA}_{\texttt{FT}}$.), Influence Function (**`IF`**) (Koh and Liang 2017), Fisher Forgetting (**`Fisher`**) (Golatkar, Achille, and Soatto 2020a) and NTK Forgetting (**`NTK`**) (Golatkar, Achille, and Soatto 2020b), Teacher-Student Formulation **`(SCRUB)`** (Kurmanji et al. 2023) and (**`Bad-T`**) (Chundawat et al. 2023), Freezing Last k-layers Followed by Catastrophic Forgetting-k (**`CF-k`**) and Exact Unlearning-k (**`EU-k`**)(Goel, Prabhu, and Kumaraguru 2022), along with their **S**oft-**W**eighted (**`SW-`**) versions. Technical details can be found in Appendix A.3.

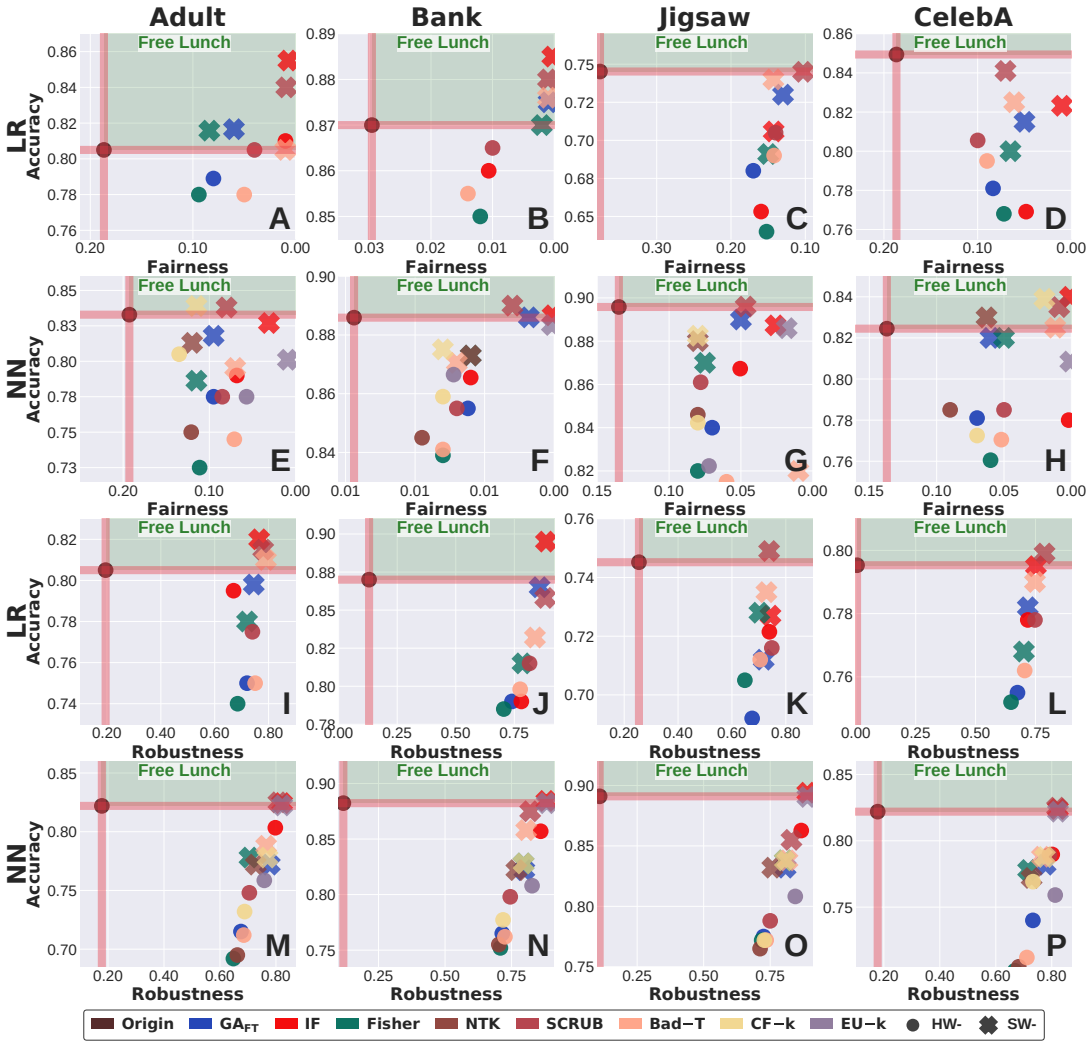

Figure 2: Performance on fairness and robustness tasks. Different colors represent different machine unlearning algorithms: the circle marker denotes the Hard-Weighted algorithms, and the cross marker denotes the Soft-Weighted algorithms. The first row (LR) and second row (NN) plot utility (Y-axis) against fairness (X-axis) metrics, while the third row (LR) and fourth row (NN) plot utility (Y-axis) against robustness (X-axis) metrics across datasets. The Green Region highlights the Free Lunch cases, where the unlearning algorithms improve both the fairness (or robustness) and utility compared to the original model.

We evaluate the performance of different unlearning algorithms under a maximum budget of 30 epochs. For the hard-weighted scheme, we perform unlearning by iteratively removing the most detrimental samples until no further improvement is observed in fairness or robustness. It is essential to note that the performance of unlearning methods may vary across datasets/models, depending on the hyperparameter choices, and the selected configurations may not be optimal. Our goal is not to assess the superiority of each algorithm, but rather to compare the differences between hard- and soft-weighted schemes, under the same cost constraints. Finally, we evaluate ResNet-50 on CIFAR-100 for robustness and ResNet-18 on CelebA for fairness, with the configurations and results deferred to Appendix B.4 due to space limitations.

Performance on fairness and robustness tasks. Figure 2 visualizes how the soft-weighted method improves fairness and robustness while avoiding excessive utility degradation. We can observe the following: (i) Compared to the hard-weighted method, in all scenarios **(A–P)**, the soft-weighted scheme consistently achieves superior performance on the target task. This improvement results from optimizing sample weights through the objective in Equation (6a), subject to the target-specific constraint in Equation (6b), which helps prevent over-unlearning. Furthermore, by incorporating the utility constraint in Equation (6c), the soft-weighted method effectively mitigates the degradation of generalization information that is often observed in the hard-weighted approach. (ii) Compared to the original model, in most scenarios **(A–B, E–P)**, the soft-weighted scheme not only improves target task performance but also enhances utility under certain algorithms. We refer to these instances as *free lunch* cases, where both the target tasks and utility are simultaneously improved.

Method	Adult				Bank				CelebA				Jigsaw			
	Fair.	Utility	Robust.	Utility	Fair.	Utility	Robust.	Utility	Fair.	Utility	Robust.	Utility	Fair.	Utility	Robust.	Utility
GA$_{FT}$	25.0	3.5	3.5	6.4	98.9	2.3	15.9	9.5	40.2	4.4	6.4	3.6	23.5	7.4	6.4	2.9
IF	28.6	5.6	14.2	3.1	97.2	2.9	13.5	13.3	79.3	7.0	4.7	2.2	10.6	8.1	0.5	0.8
Fisher	10.8	4.6	5.1	5.4	83.3	2.4	11.3	3.8	9.7	4.2	8.3	2.1	1.4	8.0	8.3	3.3
SCRUB	77.5	4.3	5.4	5.2	90.0	1.7	8.2	5.3	29.7	4.4	5.3	2.7	26.4	5.7	1.3	4.6
Bad-T	80.0	3.2	5.3	8.0	92.9	2.5	8.1	4.3	32.2	3.8	6.1	3.7	-0.7↓	7.2	3.3	3.2

Table 1: Percentage (%) improvement of soft-weighted scheme over hard-weighted counterpart on the convex model.

Method	Adult				Bank				CelebA				Jigsaw			
	Fair.	Utility	Robust.	Utility	Fair.	Utility	Robust.	Utility	Fair.	Utility	Robust.	Utility	Fair.	Utility	Robust.	Utility
GA$_{FT}$	0.0	5.5	15.3	8.0	70.6	3.6	11.6	7.5	14.3	5.0	6.2	5.7	28.6	6.0	11.4	7.4
IF	54.7	4.7	3.0	2.6	98.1	2.4	2.5	3.2	50.0	7.7	2.5	4.3	48.7	2.3	2.6	3.6
Fisher	-3.4 ↓	8.5	8.9	12.4	25.0	4.1	12.3	10.1	16.7	7.8	7.1	10.8	6.3	6.1	12.1	8.5
NTK	0.8	8.4	10.0	11.1	36.8	3.3	9.4	8.9	30.0	5.7	6.7	9.5	0.0	4.0	6.5	8.8
SCRUB	5.9	8.1	14.7	10.3	57.1	4.1	10.1	9.6	82.0	6.4	3.4	4.5	40.1	4.1	10.7	8.5
Bad-T	1.4	6.8	11.7	10.7	12.5	3.5	11.0	12.6	76.9	7.1	7.6	10.7	83.3	0.6	10.9	8.5
CF-k	14.8	4.2	11.9	6.3	0.0	1.9	10.7	6.5	71.4	8.6	5.8	2.4	0.0	4.8	10.6	8.5
EU-k	84.4	3.4	8.4	8.4	97.8	2.0	6.7	9.2	90.0	8.4	1.2	8.3	74.9	7.7	5.4	10.2

Table 2: Percentage (%) improvement of soft-weighted scheme over hard-weighted counterpart on the non-convex model.

Comparison between influence evaluation and actual value. Figure 3 shows that the influence estimations from Step 1 exhibit strong correlation with the actual values in terms of utility, fairness, and robustness metrics, with Spearman and Pearson correlations close to 1, which aligns with the findings of existing studies (Koh and Liang 2017).

Percentage improvement of the soft-weighted scheme over the hard-weighted counterpart. Tables 1 and 2 demonstrate that the soft-weighted framework enhances the performance of unlearning algorithms. Except for two gradient-based baselines, which exhibit slight reductions in fairness (attributed to overhead constraints), all other methods achieve joint improvements in both fairness (or robustness) and utility. Remarkably, the soft-weighted scheme yields significant gains in fairness-related tasks, achieving improvements of up to 98.9% over the hard-weighted scheme. Moreover, prior empirical evidence from (Chen et al. 2024b) and the results in Appendix B suggest that even hard-weighted **IF** outperforms traditional fairness solutions, while maintaining the efficient runtime. This collectively indicates that the soft-weighted framework has strong potential as an advancement in improving fairness/robustness within machine unlearning.

Runtime. Both the hard- and soft-weighted schemes share Step 1 (influence evaluation) and Step 3 (model correction). Remarkably, the soft-weighted framework introduces a lightweight procedure in Step 2 to replace the binary assignment, which incurs negligible overhead (only 0.03% of the total execution time for **IF**) while yielding superior performance in Step 3. Due to space constraints, we defer the visualization of runtime results to Appendix B.4. Hence, an advantage of our framework is *tuning-free* nature. In contrast to hard-weighted methods that require repetitive deletion trials at various rates to manually identify the optimal rate, our scheme requires only a closed-form solution in Step 2.

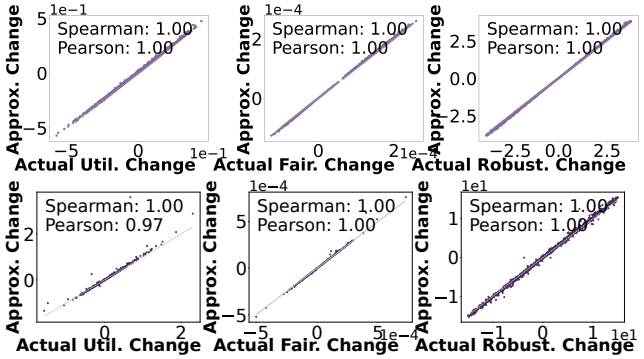

Figure 3: Actual Changes vs. Approximate Changes. We evaluated the leave-one-out influence on the Adult dataset, with the first row for LR and the second row for the last layer of NN, on different performance metrics as follows: utility (loss on test set) (Left), fairness (DP loss on test set) (Middle), robustness (loss on adversarial sample) (Right).

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.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2024YFE020010, and in part by the National Natural Science Foundation of China under Grants 62202427, 62502412, and 62271280.

References

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

Caton, S.; and Haas, C. 2024. Fairness in Machine Learning: A Survey. *ACM Comput. Surv.*, 56(7): 166:1–166:38.

Chen, H.; Zhu, T.; Yu, X.; and Zhou, W. 2024a. 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*.

Chen, J.; and Yang, D. 2023. Unlearn What You Want to Forget: Efficient Unlearning for LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 12041–12052.

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

Chen, Z.; Wang, J.; Zhuang, J.; Reddy, A. G.; Silvestri, F.; Huang, J.; Nag, K.; Kuang, K.; Ning, X.; and Tolomei, G. 2024c. Debiasing Machine Unlearning with Counterfactual Examples. *CoRR*, abs/2404.15760.

Chhabra, A.; Li, P.; Mohapatra, P.; and Liu, H. 2024. "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*.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. S. 2023. Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks Using an Incompetent Teacher. In *Proc. 37th AAAI Conf. Artif. Intell. (AAAI), 35th IAAI, 13th EAAI*, 7210–7217. AAAI Press.

Cui, Y.; Ding, N.; and Cheung, M. H. 2025. AoI-Aware Federated Unlearning for Streaming Data with Online Client Selection and Pricing. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*, 1–10. IEEE.

Dige, O.; Arneja, D.; Yau, T. F.; Zhang, Q.; Bolandraftar, M.; Zhu, X.; and Khattak, F. K. 2024. Can Machine Unlearning Reduce Social Bias in Language Models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*.

Ding, N.; Sun, Z.; Wei, E.; and Berry, R. 2025. Incentivized federated learning and unlearning. *IEEE Transactions on Mobile Computing*.

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

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *CVPR*, 1625–1634. Computer Vision Foundation / IEEE Computer Society.

Feldman, V.; and Zhang, C. 2020. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. In *NeurIPS, virtual*.

Goel, S.; Prabhu, A.; and Kumaraguru, P. 2022. Evaluating Inexact Unlearning Requires Revisiting Forgetting. *CoRR*, abs/2201.06640.

Golatkar, A.; Achille, A.; and Soatto, S. 2020a. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In *CVPR*. Computer Vision Foundation / IEEE.

Golatkar, A.; Achille, A.; and Soatto, S. 2020b. Forgetting Outside the Box: Scrubbing Deep Networks of Information Accessible from Input-Output Observations. In *ECCV*, volume 12374, 383–398. Springer.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, USA, 2015, Conference Track Proceedings*.

Guo, C.; Goldstein, T.; Hannun, A. Y.; and van der Maaten, L. 2020. 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*, Proceedings of Machine Learning Research. PMLR.

Gurobi. 2024. Gurobi Optimizer Reference Manual. https://www.gurobi.com. Accessed: 2025-06-12.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 3315–3323.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hu, H.; Wang, S.; Chang, J.; Zhong, H.; Sun, R.; Hao, S.; Zhu, H.; and Xue, M. 2024. A Duty to Forget, a Right to be Assured? Exposing Vulnerabilities in Machine Unlearning Services. In *NDSS*.

Huang, L.; Su, T.; Gao, C.; Liu, N.; and Huang, Q. 2025. AUTE: Peer-Alignment and Self-Unlearning Boost Adversarial Robustness for Training Ensemble Models. In *AAAI*.

Hutchinson, B.; and Mitchell, M. 2019. 50 Years of Test (Un)fairness: Lessons for Machine Learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA*. ACM.

Izmailov, P.; Kirichenko, P.; Gruver, N.; and Wilson, A. G. 2022. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*.

Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; and Seo, M. 2023. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. In *ACL*, 14389–14408. Association for Computational Linguistics.

Jia, J.; Liu, J.; Ram, P.; Yao, Y.; Liu, G.; Liu, Y.; Sharma, P.; and Liu, S. 2023. Model Sparsity Can Simplify Machine Unlearning. In *NeurIPS 36*.

Jia, J.; Zhang, Y.; Zhang, Y.; Liu, J.; Runwal, B.; Diffenderfer, J.; Kailkhura, B.; and Liu, S. 2024. SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning. In *EMNLP*, 4276–4292. Association for Computational Linguistics.

Kirichenko, P.; Izmailov, P.; and Wilson, A. G. 2023. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In *The Eleventh International Conference on Learning Representations*.

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

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.(2009).

Kurmanji, M.; Triantafillou, E.; and Triantafillou, P. 2024. Machine unlearning in learned databases: An experimental analysis. *Proceedings of the ACM on Management of Data*.

Kurmanji, M.; Triantafillou, P.; Hayes, J.; and Triantafillou, E. 2023. Towards Unbounded Machine Unlearning. In *NeurIPS*.

Li, W.; Li, J.; de Witt, C. S.; Prabhu, A.; and Sanyal, A. 2024. Delta-Influence: Unlearning Poisons via Influence Functions. *arXiv preprint arXiv:2411.13731*.

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

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. 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*.

Megyeri, I.; Hegedüs, I.; and Jelasity, M. 2019. Adversarial robustness of linear models: regularization and dimensionality. In *ESANN*.

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

Noever, D. 2018. Machine Learning Suites for Online Toxicity Detection. *CoRR*, abs/1810.01869.

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

Qiao, X.; Zhang, M.; Tang, M.; and Wei, E. 2025. Hessian-Free Online Certified Unlearning. In *The Thirteenth International Conference on Learning Representations*.

Qiu, S.; Potapczynski, A.; Izmailov, P.; and Wilson, A. G. 2023. Simple and fast group robustness by automatic feature reweighting. In *ICML*. PMLR.

Rudner, T. G. J.; Shi Zhang, Y.; Wilson, A. G.; and Kempe, J. 2024. Mind the GAP: Improving Robustness to Subpopulation Shifts with Group-Aware Priors. In *AISTATS*.

Schrouff, J.; Bellot, A.; Rannen-Triki, A.; Malek, A.; Albuquerque, I.; Gretton, A.; D'Amour, A.; and Chiappa, S. 2024. Mind the graph when balancing data for fairness or robustness. *Advances in Neural Information Processing Systems*.

Sheng, X.; Bao, W.; and Ge, L. 2024. Robust Federated Unlearning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*. ACM.

Shi, W.; Lee, J.; Huang, Y.; Malladi, S.; Zhao, J.; Holtzman, A.; Liu, D.; Zettlemoyer, L.; Smith, N. A.; and Zhang, C. 2024. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. *CoRR*, abs/2407.06460.

Tran, K.; and Woo, S. S. 2025. Fairness and Robustness in Machine Unlearning. *arXiv preprint arXiv:2504.13610*.

Wang, Q.; Xu, R.; He, S.; Berry, R.; and Zhang, M. 2025a. Unlearning Incentivizes Learning under Privacy Risk. In *Proceedings of the ACM on Web Conference 2025*, 1456–1467.

Wang, S.; Shen, Z.; Qiao, X.; Zhang, T.; and Zhang, M. 2025b. DynFrs: An Efficient Framework for Machine Unlearning in Random Forest. In *The Thirteenth International Conference on Learning Representations*.

Wang, S.; Wang, P.; Zhou, T.; Dong, Y.; Tan, Z.; and Li, J. 2025c. CEB: Compositional Evaluation Benchmark for Fairness in Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Welfert, M.; Stromberg, N.; and Sankar, L. 2024. Fairness-Enhancing Data Augmentation Methods for Worst-Group Accuracy. *Proceedings of Machine Learning Research*.

Wen, S.; Zhang, M.; Yang, Y.; and Ding, N. 2025. FedShard: Federated Unlearning with Efficiency Fairness and Performance Fairness. *arXiv preprint arXiv:2508.09866*.

Xiong, P.; Tegegn, M.; Sarin, J. S.; Pal, S.; and Rubin, J. 2024. It Is All about Data: A Survey on the Effects of Data on Adversarial Robustness. *ACM Comput. Surv.*

Yao, Y.; Xu, X.; and Liu, Y. 2023. Large Language Model Unlearning. *CoRR*, abs/2310.10683.

Zhang, B.; Dong, Y.; Wang, T.; and Li, J. 2024a. 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.

Zhang, H.; Zhang, Z.; Zhang, Y.; Zhai, Y.; Peng, H.; Lei, Y.; Yu, Y.; Wang, H.; Liang, B.; Gui, L.; and Xu, R. 2024b. Correcting Large Language Model Behavior via Influence Function. arXiv:2412.16451.

Zhang, R.; Lin, L.; Bai, Y.; and Mei, S. 2024c. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. *CoRR*, abs/2404.05868.

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