




Challenging Forgets: Unveiling the Worst-Case Forget Sets in Machine Unlearning

Chongyu Fan^{*,1}, Jiancheng Liu^{*,1}, Alfred Hero², Sijia Liu^{1,3}

¹ OPTML@CSE, Michigan State University

² EECS, University of Michigan, Ann Arbor

³ MIT-IBM Watson AI Lab, IBM Research

*Equal contribution

{fanchon2,liujia45,liusiji5}@msu.edu, hero@eecs.umich.edu

Abstract. The trustworthy machine learning (ML) community is increasingly recognizing the crucial need for models capable of selectively ‘unlearning’ data points after training. This leads to the problem of *machine unlearning* (MU), aiming to eliminate the influence of chosen data points on model performance, while still maintaining the model’s utility post-unlearning. Despite various MU methods for data influence erasure, evaluations have largely focused on *random* data forgetting, ignoring the vital inquiry into which subset should be chosen to truly gauge the authenticity of unlearning performance. To tackle this issue, we introduce a new evaluative angle for MU from an adversarial viewpoint. We propose identifying the data subset that presents the most significant challenge for influence erasure, *i.e.*, pinpointing the *worst-case* forget set. Utilizing a bi-level optimization principle, we amplify unlearning challenges at the upper optimization level to emulate worst-case scenarios, while simultaneously engaging in standard training and unlearning at the lower level, achieving a balance between data influence erasure and model utility. Our proposal offers a worst-case evaluation of MU’s resilience and effectiveness. Through extensive experiments across different datasets (including CIFAR-10, 100, CelebA, Tiny ImageNet, and ImageNet) and models (including both image classifiers and generative models), we expose critical pros and cons in existing (approximate) unlearning strategies. Our results illuminate the complex challenges of MU in practice, guiding the future development of more accurate and robust unlearning algorithms. The code and supplementary material (appendix) are available at <https://github.com/OPTML-Group/Unlearn-WorstCase>.

1 Introduction

In this work, we study the problem of machine unlearning (MU) [6, 7, 39, 45], which aims to erase unwanted data influences (*e.g.*, specific data points, classes, or knowledge concepts) from a machine learning (ML) model, while preserving the utility of the model post-unlearning (termed ‘unlearned model’) for information not targeted by the unlearning process. The concept of MU was initially developed to meet data protection regulations, *e.g.*, the ‘right to be

forgotten’ [30, 50]. Given its ability to evaluate data’s impact on model performance, the application of MU has expanded to address a variety of trustworthy ML challenges. These include defending against ML security threats [33, 40], removing data biases for enhanced model fairness [12, 46, 52], protecting copyright and privacy [1, 19, 76], and mitigating sociotechnical harms by, *e.g.*, erasing generative models’ propensity for producing toxic, discriminatory, or otherwise undesirable outputs [20, 21, 72].

With the growing significance and popularity of MU, a wide array of unlearning methods has been devised across various domains, such as image classification [20, 25, 32, 33, 60, 64], text-to-image generation [20–22, 29, 37, 72, 76], federated learning [9, 41, 63, 65], graph neural networks [11, 13, 14], and large language modeling [19, 66, 69, 70, 73]. In this work, we delve into MU in vision tasks, primarily concentrating on image classification but also extending our investigation to text-to-image generation. For a detailed review of existing MU methods, we refer readers to Secs. 2 and 3.

Our study concentrates on improving the reliability of MU evaluation, considering the diversity of unlearning tasks and methodologies. Our motivation stems from the limitations in current MU evaluation methods, which heavily rely on artificially constructed *random data forgetting* scenarios [16, 20, 33, 38], particularly noticeable in MU for image classification. However, the observation in [20] and our own investigations (in Sec. 3) indicate that the effectiveness of unlearning methods can significantly vary with the selection of the forget set (*i.e.*, the specific data points designated for forgetting), resulting in substantial performance variance. This unlearning variability based on forget set choices leads us to reconsider the possibility of exploring a *worst-case forget set* selection scenario. Such a scenario would ideally represent the most challenging conditions for an unlearning method’s performance, reducing unlearning variance and facilitating a more reliable assessment. This leads us to the following research question:

(Q) *How can we pinpoint the worst-case forget set for MU to accurately assess its unlearning efficacy under such challenging conditions?*

Tackling (Q) can also be viewed as creating an adversarial evaluation to challenge the effectiveness of MU. Yet, different from existing adversarial metrics such as membership inference attacks (MIAs) [8, 57] or adversarial prompts in MU for image generation [77], the worst-case forget set method assesses MU from a *data selection* perspective, and is compatible with other metrics for evaluating unlearning effectiveness and utility. We will show that the method of selecting a worst-case forget set can be readily extended to different unlearning scenarios, including class-wise forgetting (aiming to remove the impact of

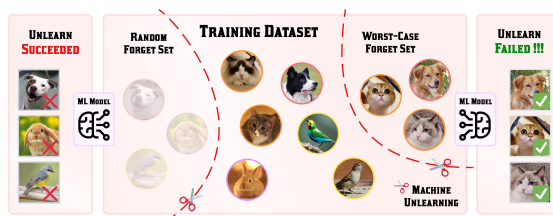


Fig. 1: Overview of unlearning under our proposal (worst-case forget set) vs. random forget set. The data influence is difficult to remove under worst-case forget set vs. random forget set.

an entire image class) and prompt-wise forgetting in text-to-image generation (aimed at avoiding generation conditioned on certain text prompts). **Fig. 1**¹ shows an overview of the worst-case forget set vs. random forget set.

Contributions. We summarize our contributions below.

- We are the first, to our knowledge, to highlight the necessity of identifying the worst-case forget set for MU, and develop a solid formulation and optimization foundation through the lens of bi-level optimization (BLO).
- We introduce an effective algorithmic framework for identifying the worst-case forget set, offering two distinct benefits: optimization efficiency, which reduces the complexity of BLO through the sign-based gradient unrolling method, and optimization generality, making it easily adaptable to worst-case evaluations in both class-wise and prompt-wise forgetting scenarios.
- We assess the empirical effectiveness of worst-case forget set-based MU evaluation, showcasing the strength of our approach and elucidating the rationale behind the chosen forget sets in terms of coreset selection and class-discriminative ability. Additionally, we explore the applicability of the worst-case forget set, extending from image classification to text-to-image generation.

2 Related Works

Machine unlearning (MU). MU involves modifying ML models to eliminate the influence of specific data points, classes, or even broader knowledge-level concepts [7, 23, 44, 54, 62]. A widely recognized *exact* unlearning strategy refers to *retraining the model from scratch* (termed *Retrain*), executed by omitting the data points designated for forgetting from the training set [60, 61]. While Retrain offers a solid guarantee of the data influence erasure [18, 26], it demands significant training costs, rendering it practically challenging for deep models. To overcome the efficiency challenges of MU, many research efforts have shifted focus towards creating *approximate* unlearning methods [20, 23, 25, 27, 32, 33, 44, 54, 60, 62, 64]. Some representative methods will be reviewed in Sec. 3.

In the above literature, the applications of MU have mainly focused on image classification tasks, targeting either class-wise forgetting, which seeks to erase the impact of an entire image class, or random data forgetting, aimed at removing randomly chosen data points from the training set. However, the scope and use cases of MU have significantly broadened recently. For instance, within the field of text-to-image generation using diffusion models (DMs), several studies [20–22, 29, 37, 72, 76] have applied the concept of MU to mitigate the harmful effects of inappropriate or sensitive prompts on image generation, aiming to enhance the safety of DMs. In addition, the significance of MU in enhancing the trustworthiness of data-models has been recognized across other non-vision domains, including graph neural networks [11, 13, 14], federated learning [9, 41, 63, 65], and the rapidly evolving field of large language models (LLMs) [19, 66, 69, 70, 73]. In this study, we focus on vision-related tasks.

¹ Thanks to Naughty, Fries, Crescent, Catcat, Wula, and other cuties for their appearance in Fig. 1.

Evaluation of MU. Assessing the effectiveness of MU presents its own set of challenges [21, 33, 61, 77]. Generally speaking, *unlearning effectiveness* and post-unlearning *model utility* are the two primary factors considered for assessing the performance of MU. When applying MU to classification tasks, effectiveness-oriented metrics include unlearning accuracy, which relates to the model’s performance accuracy after unlearning on the forget set [25], and MIAs to determine if a data point in the forget set was part of the model’s training set post-unlearning [59]. Utility-oriented metrics include remaining accuracy, which evaluates the performance of the updated model post-unlearning on the retain set [2], and testing accuracy, assessing the updated model’s generalization capability. When applying MU to generation tasks, accuracy-based metrics are also employed through the use of a post-generation classifier applied to the generated content [76, 77], while quality metrics of generations are employed to assess utility [21]. However, a notable limitation of the above evaluation metrics is their significant dependency on the specific unlearning tasks at hand. For example, the task of random data forgetting in image classification may result in considerable variance in the measurements of unlearning effectiveness [20], attributed to the randomness in selecting the forget set.

Data selection for deep learning. Data selection methods, such as dataset pruning [47, 48, 53, 68, 75] and coreset selection [5, 31, 34, 55, 67], serve as valuable strategies for enhancing the efficiency of model training [15]. The aim of data selection is to identify the most representative training samples or eliminate the least influential ones, remaining the model’s performance unaffected after training on the chosen data points. In addition to efficiency, data selection is also employed to enhance model security by detecting and filtering out poisoned data points [71], as well as to increase fairness by eliminating biased data points [52]. In this work, the challenge of determining the worst-case forget set resonates with data selection strategies but applies to the novel realm of machine unlearning for the first time. Methodology-wise, the idea of bi-level optimization (BLO), previously applied in data selection contexts [5, 68], will also be used for addressing our focused problem. Nonetheless, we will adapt BLO for the unlearning context and devise computationally efficient optimization strategies.

3 Preliminaries, Motivation, and Problem Statement

Objective and setup of MU. The objective of MU is to negate the impact of a specific subset of training data points on a (pre-trained) model, while preserving its utility for data not subject to unlearning. For a concrete setup of MU, consider the training dataset $\mathcal{D} = \{\mathbf{z}_i\}_{i=1}^N$, consisting of N data samples. Each sample \mathbf{z}_i includes a feature vector \mathbf{x}_i and a possible label \mathbf{y}_i for supervised learning. Let $\mathcal{D}_f \subseteq \mathcal{D}$ represent the subset of data targeted for unlearning, with its complement, $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$, being the dataset to retain. We refer to \mathcal{D}_f as the *forget set* and \mathcal{D}_r as the *retain set*, respectively. Prior to unlearning, we have access to an initial model, denoted by θ_o , which could be trained on the full dataset \mathcal{D} using methods like empirical risk minimization (ERM).

Given the above setup, Retrain, an exact yet expensive unlearning approach, entails retraining the model θ_o from scratch, exclusively utilizing the retain set \mathcal{D}_r . It is typically regarded as the gold standard in MU [33, 60]. However, due to the prolonged training time and the high cost, Retrain is often impractical. Consequently, approximate unlearning methods have emerged as efficient alternatives. Their objective is to efficiently create an *unlearned model*, denoted as θ_u , leveraging prior knowledge of θ_o and the forget set \mathcal{D}_f and/or the retain set \mathcal{D}_r . Following the conceptual framework of MU in [39], the optimization problem to obtain θ_u can be expressed as

$$\theta_u = \arg \min_{\theta} \ell_{\text{MU}}(\theta) := \ell_r(\theta; \mathcal{D}_r) + \lambda \ell_f(\theta; \mathcal{D}_f), \quad (1)$$

where ℓ_f and ℓ_r represent the forget loss and the retain loss, respectively, with $\lambda \geq 0$ acting as a regularization parameter. For instance, fine-tuning using the retain set \mathcal{D}_r equates to setting $\lambda = 0$, aimed to impose catastrophic forgetting of over \mathcal{D}_f after model fine-tuning. Note that the specifics of ℓ_f , ℓ_r , and λ can differ across various MU methodologies.

Reviewing representative MU methods.

Assisted by (1), we provide an overview of 9 existing (approximate) unlearning methods examined in this study; see **Table 1** for a summary. These methods can be roughly categorized into two main groups based on the choice of the forget loss ℓ_f : *relabeling-free* and *relabeling-based*. The latter, relabeling-based methods, assign an *altered* label, distinct from the true label, to the data point targeted for forgetting. Consequently,

minimizing ℓ_f compels the unlearned model to discard the accurate label of the points to be forgotten. These methods include random labeling (**RL**) [25], boundary expanding (**BE**) [10], boundary shrinking (**BS**) [10], and saliency unlearning (**SalUn**) [20]. In contrast, relabeling-free methods utilize fine-tuning on the retain set \mathcal{D}_r to induce catastrophic forgetting or apply gradient ascent on the forget set \mathcal{D}_f to achieve the forgetting objective. These methods include fine-tuning (**FT**) [64], exact unlearning restricted to the last k layers (**EU- k**) [24], catastrophically forgetting the last k layers (**CF- k**) [24], scalable remembering and unlearning unbound (**SCRUB**) [38] and ℓ_1 -sparse MU [33].

In addition to relabeling differences, the aforementioned MU methods also vary in several other aspects. For instance, RL can apply exclusively to the forget loss, corresponding to $\lambda \rightarrow +\infty$ in (1). Methods including SalUn, EU- k , and CF- k target only a subset of model parameters, not the entire model θ during the optimization process. Furthermore, methods like BE and EU- k necessitate re-initializing the pre-trained model state. Table 1 summarizes the configurations for the examined unlearning methods in this study.

Table 1: Overview of examined MU methods highlighting differences in relabeling-based forget loss, necessity of random re-initialization, partial model updates during unlearning, and the retain-forget regularization parameter λ within (1).

Method	Relabeling	Random re-initialization	Partial model update	$\lambda = 0$
Retrain	✗	✓	✗	✓
FT [64]	✗	✗	✗	✓
EU- k [24]	✗	✓	✓	✓
CF- k [24]	✗	✗	✓	✓
SCRUB [38]	✗	✗	✗	✗
ℓ_1 -sparse [33]	✗	✗	✗	✗
RL [25]	✓	✗	✗	✗
BE [10]	✓	✓	✗	✗
BS [10]	✓	✗	✗	✗
SalUn [20]	✓	✗	✓	✗

Challenge in MU evaluation: Sensitivity to forget set selection. When assessing the effectiveness of MU, a typical approach is *random data forgetting*, which measures the unlearning ability when eliminating the influence of *randomly* selected data points from the training set. However, evaluations based on the random selection of both data points and their quantity for forgetting can lead to *high variance* in the performance of a specific unlearning method, complicating fair comparisons across different MU methods. Most importantly, a randomly selected forget set *cannot* reveal the *worst-case* unlearning performance, raising concerns about the reliability of an MU method.

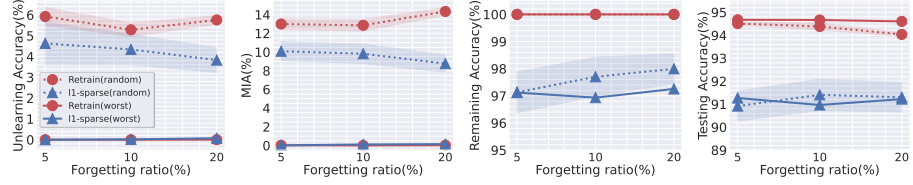


Fig. 2: Performance of Retrain and ℓ_1 -sparse unlearning under random and worst-case forgetting scenarios at different forgetting data ratios on (CIFAR-10, ResNet-18). Variance over 10 random selections is indicated by the shaded areas of the dashed lines.

Fig. 2 provides a motivating example showcasing the sensitivity of MU evaluation to the (random) forget set selection. We examine two unlearning approaches: Retrain (exact unlearning) and ℓ_1 -sparse (a state-of-the-art approximate unlearning method), and explore two evaluation scenarios: traditional random data forgetting and our proposed worst-case forget set evaluation, which will be elaborated in Sec. 4. For a given unlearning method and forget set selection, we assess the unlearning effectiveness and model utility of the unlearned model θ_u following the evaluation metrics used in [33]. The effectiveness metrics include unlearning accuracy (UA), calculated as *1 - the accuracy of θ_u on the forget set \mathcal{D}_f* , and MIA (membership inference attack) on \mathcal{D}_f , which determines whether a data point in \mathcal{D}_f was unused in training θ_u . The utility metrics include remaining accuracy (RA), measured by the accuracy of θ_u on the retain set \mathcal{D}_r , and testing accuracy (TA), the accuracy of θ_u on the test set.

Two main observations can be drawn from Fig. 2. First, both exact unlearning (Retrain) and approximate unlearning (ℓ_1 -sparse) with random forget sets result in *increased variance* in UA and MIA, compared to the performance with identified worst-case forget sets (Sec. 4). Second, our approach effectively highlights unlearning challenges, demonstrated by the lowest UA and MIA at various forgetting ratios (a lower UA or MIA corresponds to a higher unlearning difficulty). Importantly, these identified worst-case forget sets do *not* compromise model utility, as shown by the RA and TA of Retrain.

Problem of interest: Identification of the worst-case forget set. As inspired by Table 1, evaluating MU through random data forgetting can lead to a high performance variance and provides limited insight into the worst-case performance of MU. To tackle these challenges, we propose to devise a systematic strategy to identify the data subset that presents the most significant challenge for influence erasure in MU, while preserving the utility of the unlearned model.

We define this identified subset as the **worst-case forget set**. Approaching from an adversarial perspective, our interest also lies in identifying the forget set that could diminish the unlearning performance, unveiling the worst-case unlearning scenario. In the next section, we will address the problem of identifying the worst-case forget set via a BLO-based data selection framework.

4 Identifying the Worst-case Forget Set through BLO

A BLO view on the worst-case forget set identification for MU. BLO (bi-level optimization) offers a hierarchical learning framework, featuring two tiers of optimization tasks, *i.e.*, the upper and lower levels. In this structure, the objective and variables of the upper-level problem are contingent upon the solution of the lower-level problem. In the context of identifying the worst-case forget set, we optimize the selection of a forget set at the *upper* level to maximize the difficulty of unlearning. Concurrently, the *lower* level is dedicated to generating the unlearned model, aiming to meet the unlearning objectives without compromising the utility on non-forgetting data points.

We introduce an optimization variable $\mathbf{w} \in \{0, 1\}^N$, recalling that N represents the total number of training data points. Here $w_i = 1$ signifies that the i -th training data point is included in the forget set, *i.e.*, $\mathcal{D}_f = \{\mathbf{z}_i | w_i = 1\}$. Thus, our objective is to optimize the data selection scheme \mathbf{w} , such that the associated \mathcal{D}_f can characterize the worst-case performance of an unlearned model, *i.e.*, challenging the model θ_u in (1) post the unlearning of the designated forget set.

We first form the *lower-level* optimization problem to determine the unlearned model θ_u based on the forget set defined by \mathbf{w} . By integrating \mathbf{w} into (1), the unlearning problem in *lower-level optimization* can be cast as

$$\theta_u(\mathbf{w}) = \arg \min_{\theta} \ell_{\text{MU}}(\theta; \mathbf{w}) := \sum_{\mathbf{z}_i \in \mathcal{D}} [w_i \ell_f(\theta; \mathbf{z}_i) + (1 - w_i) \ell_r(\theta; \mathbf{z}_i)], \quad (2)$$

where $\theta_u(\mathbf{w})$ signifies the resulting unlearned model that is a function of \mathbf{w} , and the loss terms $\sum_{\mathbf{z}_i \in \mathcal{D}} [w_i \ell_f(\theta; \mathbf{z}_i)]$ and $\sum_{\mathbf{z}_i \in \mathcal{D}} [(1 - w_i) \ell_r(\theta; \mathbf{z}_i)]$ correspond to the forget loss and the retain loss in (1) on the forget set \mathcal{D}_f and the retain set \mathcal{D}_r , respectively. Unless specified otherwise, we specify the unlearning objective (2) through the FT-based unlearning strategy, with $\lambda = 1$ and $\ell_f = -\ell_r$ in (1). Here, both loss functions are given by the training loss ℓ (*e.g.*, the cross-entropy loss for image classification) over θ , with the forget loss $\ell_f = -\ell$ designed to counteract the training, thereby enforcing the unlearning.

With the unlearned model $\theta_u(\mathbf{w})$ defined as a function of the data selection scheme \mathbf{w} , we proceed to outline the BLO framework by incorporating an *upper-level* optimization. This is designed to optimize \mathbf{w} for the worst-case unlearning performance, yielding the *overall BLO problem*:

$$\min_{\mathbf{w} \in \mathcal{S}} \underbrace{\sum_{\mathbf{z}_i \in \mathcal{D}} [w_i \ell(\theta_u(\mathbf{w}); \mathbf{z}_i)] + \gamma \|\mathbf{w}\|_2^2}_{\text{Upper-level objective} := f(\mathbf{w}, \theta_u(\mathbf{w}))}; \quad \text{subject to } \underbrace{\theta_u(\mathbf{w}) = \arg \min_{\theta} \ell_{\text{MU}}(\theta; \mathbf{w})}_{\text{Lower-level optimization}}, \quad (3)$$

where \mathbf{w} is the upper-level optimization variable subject to the data selection constraint set \mathcal{S} , *e.g.*, $\mathcal{S} = \{\mathbf{w} | \mathbf{w} \in \{0, 1\}^N, \mathbf{1}^\top \mathbf{w} = m\}$ with m being the forget set size, the lower-level objective function ℓ_{MU} has been defined in (2), and ℓ denotes the training loss. In addition, minimizing $\sum_{\mathbf{z}_i \in \mathcal{D}} [w_i \ell(\boldsymbol{\theta}_u(\mathbf{w}); \mathbf{z}_i)]$ renders the worst-case scenario of the unlearned model $\boldsymbol{\theta}_u(\mathbf{w})$ (derived from the lower-level optimization), *i.e.*, making it ineffective at erasing the influence of the forget set (corresponding to $\{w_i = 1\}$) on model performance. Furthermore, we introduce an ℓ_2 regularization term with the regularization parameter $\gamma \geq 0$ in the upper-level objective function. This has dual purposes: it encourages sparsity in the data selection scheme \mathbf{w} (when relaxed to continuous variables) and enhances the stability of BLO by including a strongly convex regularizer.

A scalable BLO solver for worst-case forget set identification. In (3), addressing the upper-level optimization presents a significant complexity, as illustrated in the following gradient descent framework. Consider the gradient of the upper-level objective function of (3), $f(\mathbf{w}, \boldsymbol{\theta}_u(\mathbf{w}))$:

$$\frac{df(\mathbf{w}, \boldsymbol{\theta}_u(\mathbf{w}))}{d\mathbf{w}} = \nabla_{\mathbf{w}} f(\mathbf{w}, \boldsymbol{\theta}_u(\mathbf{w})) + \frac{d\boldsymbol{\theta}_u(\mathbf{w})^\top}{d\mathbf{w}} \nabla_{\boldsymbol{\theta}} f(\mathbf{w}, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_u(\mathbf{w})}, \quad (4)$$

where $\frac{d}{d\mathbf{w}}$ denotes the *full derivative* with respect to (*w.r.t.*) \mathbf{w} , while $\nabla_{\mathbf{w}} f$ and $\nabla_{\boldsymbol{\theta}} f$ represent the *partial derivatives* of the bi-variate function f *w.r.t.* \mathbf{w} and $\boldsymbol{\theta}$, respectively. In (4), the vector-wise full derivative $\frac{d\boldsymbol{\theta}_u(\mathbf{w})^\top}{d\mathbf{w}}$ is commonly known as implicit gradient (IG) [74] since $\boldsymbol{\theta}_u(\mathbf{w})$ is an implicit function of \mathbf{w} , determined by lower-level optimization of (3). Considering the difficulty of obtaining the closed-form $\boldsymbol{\theta}_u(\mathbf{w})$, the computation of IG introduces high complexity.

In the optimization literature, two primary methods are used to derive the IG: (1) The influence function (IF) approach [35, 74], which leverages the stationarity condition of the lower-level objective function that $\boldsymbol{\theta}_u(\mathbf{w})$ satisfies; And (2) the gradient unrolling (GU) approach [56, 74], which utilizes an unrolled version of a specific lower-level optimizer as an intermediate step, linking the solution of the lower-level problem to the upper-level optimization process. However, the IF approach necessitates computing the *inverse-Hessian* gradient product [35, 74] for the lower-level loss *w.r.t.* $\boldsymbol{\theta}$. Consequently, it encounters scalability issues given the fact that $\boldsymbol{\theta}$ represents the parameters of a neural network. Therefore, we adopt GU to solve the BLO problem (3), as illustrated below.

The GU strategy mainly contains two steps: **(S1)** Identifying a particular lower-level optimizer to approximate the solution $\boldsymbol{\theta}_u(\mathbf{w})$ through a finite sequence of unrolled optimization steps; And **(S2)** employing auto-differentiation to calculate the IG by tracing the solution path unfolded in (S1). Consequently, the computation complexity of GU is dependent on the choice of the lower-level optimizer in (S1). In our study, we propose employing the *sign*-based stochastic gradient descent (signSGD) [4] as the lower-level optimizer in (S1). As we will demonstrate, the adoption of signSGD greatly simplifies the computation of the IG. Specifically, we derive an approximate solution of the lower-level problem by implementing a K -step unrolling using signSGD. This yields

$$\boldsymbol{\theta}_u(\mathbf{w}) = \boldsymbol{\theta}_K; \quad \boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1} - \beta \cdot \text{sign}(\nabla_{\boldsymbol{\theta}} \ell_{\text{MU}}(\boldsymbol{\theta}_{j-1}; \mathbf{w})), \quad j = 1, 2, \dots, K, \quad (5)$$

where j represents the lower-level iteration index, $\text{sign}(\cdot)$ is the element-wise sign operation, $\beta > 0$ specifies the learning rate, and θ_0 is a random initialization. Given signSGD (5), the computation of IG can be simplified to

$$\text{IG} = \frac{d\theta_u(\mathbf{w})^\top}{d\mathbf{w}} = \frac{d\theta_{j-1}^\top}{d\mathbf{w}} - \beta \frac{d\text{sign}(\nabla_{\theta} \ell_{\text{MU}}(\theta_{j-1}; \mathbf{w}))^\top}{d\mathbf{w}} = \dots = \frac{d\theta_0^\top}{d\mathbf{w}} = \mathbf{0}, \quad (6)$$

where we used the facts that $\frac{d\text{sign}(\mathbf{x})^\top}{d\mathbf{x}} = \mathbf{0}$ which holds almost surely and θ_0 is a random initialization. We highlight that this IG simplification is induced by the sign operation. If we replace signSGD with the vanilla SGD, then the intermediate second-order derivatives, such as $\frac{d\nabla_{\theta} \ell_{\text{MU}}(\theta_{j-1}; \mathbf{w})}{d\mathbf{w}}$, would not be omitted.

By utilizing signSGD, we could effectively address the computational challenge of calculating the IG, enabling us to solve the problem (3) using solely first-order information. For example, substituting (6) into (4), the upper-level gradient *w.r.t.* \mathbf{w} reduces to the first-order partial derivative $\nabla_{\mathbf{w}} f(\mathbf{w}, \theta_u(\mathbf{w}))$. Subsequently, we can solve the BLO problem (3) for identifying the worst-case forget set through an *alternating optimization* strategy, formed by projected gradient descent (PGD) for the upper-level optimization and signSGD for the lower-level optimization. We summarize it below:

$$\text{Upper-level PGD: } \mathbf{w}_i = \text{Proj}_{\mathbf{w} \in \mathcal{S}} (\mathbf{w}_{i-1} - \alpha \nabla_{\mathbf{w}} f(\mathbf{w}, \theta_u(\mathbf{w}_{i-1}))|_{\mathbf{w}=\mathbf{w}_{i-1}}), \quad (7)$$

$$\text{Lower-level signSGD: } \theta_u(\mathbf{w}_{i-1}) = \theta_K, \text{ given by (5) at } \mathbf{w} = \mathbf{w}_{i-1}, \quad (8)$$

where i represents the step in the upper-level optimization process, \mathbf{w}_0 is an initial data selection scheme (*e.g.*, a random binary vector in classification tasks), and $\text{Proj}_{\mathbf{w} \in \mathcal{S}}(\mathbf{a})$ indicates the projection of a constant \mathbf{a} onto the constraint set \mathcal{S} . This projection operation is defined as solving the auxiliary minimization problem $\text{Proj}_{\mathbf{w} \in \mathcal{S}}(\mathbf{a}) = \arg \min_{\mathbf{w} \in \mathcal{S}} \|\mathbf{w} - \mathbf{a}\|_2^2$. For ease of optimization, we relax the binary constraint \mathcal{S} into its continuous counterpart, with $\mathbf{w} \in [0, 1]$ and $\mathbf{1}^\top \mathbf{w} = m$. This facilitates us to obtain a closed-form solution for this projection problem, as shown in Appendix A.1. Additionally, a relaxed version of \mathbf{w} offers a continuous forget score, enabling us to identify not only the worst-case forget set (identified by selecting \mathbf{w} with the top m largest magnitudes) but also the set of data points that are easiest to unlearn (identified by selecting \mathbf{w} with the top m smallest magnitudes). In practice, our alternating PGD and signSGD method (7)-(8) demonstrates good convergence properties. Typically, the upper-level optimization converges within 20 iterations, while for the lower-level problem, setting $K = 10$ epochs is found to be sufficient.

Extending to class-wise or prompt-wise forgetting. The previously proposed BLO problem (3) was conceived for identifying worst-case forget set in the context of data-wise forgetting. However, our approach can be easily extended to other MU scenarios, such as *class-wise forgetting* [25, 26, 33], and *prompt-wise forgetting* [20, 21]. For class-wise forgetting, the data selection variables \mathbf{w} in (2)-(3) can be reinterpreted as class selection variables. Here, $w_i = 1$ indicates the selection of the i th class for targeted worst-case unlearning. In the context of prompt-wise forgetting, we interpret \mathbf{w} as prompt selection variables. Here a prompt refers to a text condition used for text-to-image generation, known as a ‘concept’ within MU for generative models [21]. See Appendix A.2 for details.

5 Experiments

5.1 Experiment Setups

Unlearning tasks and setups. For *data-wise* forgetting, our primary experiments are conducted on the CIFAR-10 dataset [36] and the ResNet-18 model [28]. We further extend our evaluation to include CIFAR-100 [36], CelebA [42], and Tiny ImageNet datasets, alongside VGG [58] and ResNet-50 [28], detailed in Appendix B.1. For *class-wise* forgetting, we focus on the ImageNet [17] dataset with the ResNet-18 model. For *prompt-wise* forgetting, we consider the unlearning task of preventing the latent diffusion model [49] from generating artistic painting styles alongside image objects within the UnlearnCanvas dataset [76].

Unlearning methods and evaluation metrics. To validate the efficacy of MU evaluation on the worst-case forget set, we examine the exact unlearning method Retrain and 9 approximate unlearning methods as described in Table 1. During the evaluation, we mainly adopt 4 performance metrics as introduced in Sec. 3: *UA* and *MIA* are used to measure the unlearning effectiveness, *RA* and *TA* are used to assess the model utility post unlearning.

BLO implementation. In the upper-level optimization, we set the regularization parameter γ in (3) to 10^{-4} , and applied PGD by (7) with 20 iterations at the learning rate $\alpha = 10^{-3}$. In the lower-level optimization, SignSGD is performed with 10 epochs.

5.2 Experiment Results

Table 2: Performance of exact unlearning (Retrain) under random forget set and worst-case forget set at different forgetting data ratios on CIFAR-10 using ResNet-18. The result format is given by $a \pm b$, with mean a and standard deviation b over 10 independent trials. The performance difference is provided in Diff, represents the worst-case performance is **lower**▼, **equal**−, or **higher**▲ than random-case performance.

Metrics	1%-Data Forgetting			5%-Data Forgetting			10%-Data Forgetting			20%-Data Forgetting		
	Random	Worst-case	Diff	Random	Worst-case	Diff	Random	Worst-case	Diff	Random	Worst-case	Diff
UA	5.85 \pm 0.69	0.00 \pm 0.00	5.85▼	5.92 \pm 0.44	0.00 \pm 0.00	5.92▼	5.28 \pm 0.33	0.00 \pm 0.00	5.28▼	5.76 \pm 0.20	0.00 \pm 0.00	5.76▼
MIA	12.89 \pm 1.27	0.00 \pm 0.00	12.89▼	13.00 \pm 0.55	0.02 \pm 0.02	12.98▼	12.86 \pm 0.61	0.00 \pm 0.00	12.86▼	14.34 \pm 0.40	0.03 \pm 0.01	14.31▼
RA	99.96 \pm 0.00	99.95 \pm 0.02	0.01▼	100.00 \pm 0.00	100.00 \pm 0.00	0.00−	100.00 \pm 0.00	100.00 \pm 0.00	0.00−	100.00 \pm 0.00	100.00 \pm 0.00	0.00−
TA	93.17 \pm 0.15	93.45 \pm 0.17	0.28▲	94.51 \pm 0.07	94.67 \pm 0.08	0.16▲	94.38 \pm 0.15	94.66 \pm 0.09	0.28▲	94.04 \pm 0.08	94.60 \pm 0.08	0.56▲

Validating the worst-case forget set via Retrain. We begin by justifying the worst-case unlearning performance of the chosen forget set through the exact unlearning method, Retrain. In Table 2, we examine the performance disparities between the *worst-case forget set* and the *random forget set* in the task of MU for image classification on CIFAR-10, when employing Retrain at different forgetting data ratios including 1%, 5%, 10%, and 20%. In terms of unlearning effectiveness, the chosen worst-case forget set consistently poses the greatest challenge for unlearning in all scenarios tested, as indicated by a significant drop in UA and MIA to nearly 0% (see the ‘Worst-case’ and ‘Diff’ columns of Table 2). In addition, the variance in worst-case unlearning effectiveness performance (as measured by UA and MIA) remains significantly lower than that observed with random data forgetting at various forgetting data ratios. Furthermore, the utility

Table 3: Performance of approximate unlearning methods (including both relabeling-free and relabeling-based methods) under random forget sets and worst-case forget sets on CIFAR-10 using ResNet-18 with forgetting ratio 10%. The result format follows Table 2. Additionally, a performance gap against **Retrain** is provided in (•). The metric *averaging (avg.) gap* is calculated by averaging the performance gaps measured in all metrics. Note that the better performance of an MU method corresponds to the smaller performance gap with Retrain.

Methods	Random Forget Set					Worst-Case Forget Set				
	UA	MIA	RA	TA	Avg. Gap	UA	MIA	RA	TA	Avg. Gap
Retrain	5.28 \pm 0.33	12.80 \pm 0.61	100.00 \pm 0.00	94.38 \pm 0.15	0.00	0.00 \pm 0.00	0.00 \pm 0.00	100.00 \pm 0.00	94.66 \pm 0.09	0.00
Relabeling-free										
FT	5.08 \pm 0.39 (0.20)	10.96 \pm 0.38 (1.90)	97.46 \pm 0.52 (2.54)	91.02 \pm 0.36 (3.36)	2.00	0.00 \pm 0.00 (0.00)	0.02 \pm 0.03 (0.02)	97.63 \pm 0.46 (2.37)	91.58 \pm 0.40 (3.08)	1.37
EU- <i>k</i>	2.34 \pm 0.79 (2.94)	6.35 \pm 0.80 (6.51)	97.52 \pm 0.89 (2.48)	90.17 \pm 0.88 (4.21)	4.04	0.68 \pm 0.86 (0.68)	5.02 \pm 4.42 (5.02)	97.17 \pm 0.86 (2.83)	90.08 \pm 0.70 (4.58)	3.28
CF- <i>k</i>	0.02 \pm 0.02 (5.26)	0.76 \pm 0.02 (12.10)	99.98 \pm 0.00 (0.02)	91.45 \pm 0.02 (0.07)	4.36	0.00 \pm 0.00 (0.00)	0.00 \pm 0.00 (0.00)	99.98 \pm 0.00 (0.02)	94.34 \pm 0.00 (0.32)	0.08
SCRUB	12.42 \pm 19.82 (7.14)	22.43 \pm 24.44 (9.57)	88.31 \pm 19.78 (11.69)	83.15 \pm 17.94 (11.23)	9.91	0.01 \pm 0.01 (0.01)	0.04 \pm 0.03 (0.04)	98.65 \pm 0.33 (1.35)	92.78 \pm 0.36 (1.88)	0.82
ℓ_1 -sparse	4.34 \pm 0.73 (0.94)	9.82 \pm 1.04 (3.04)	97.70 \pm 0.72 (2.30)	91.41 \pm 0.68 (2.97)	2.31	0.02 \pm 0.03 (0.02)	0.11 \pm 0.11 (0.11)	96.93 \pm 0.73 (3.07)	90.96 \pm 0.82 (3.70)	1.72
Relabeling-based										
RL	3.59 \pm 0.24 (1.69)	28.02 \pm 2.47 (15.16)	99.97 \pm 0.01 (0.03)	93.74 \pm 0.12 (0.64)	4.38	1.93 \pm 1.11 (1.93)	96.70 \pm 0.66 (96.70)	99.96 \pm 0.01 (0.04)	93.83 \pm 0.24 (0.83)	24.88
BE	1.19 \pm 0.49 (4.09)	22.06 \pm 0.61 (9.20)	98.77 \pm 0.41 (1.23)	91.79 \pm 0.32 (2.59)	4.28	19.47 \pm 2.12 (19.47)	81.45 \pm 2.16 (81.45)	81.35 \pm 2.76 (18.65)	75.41 \pm 1.77 (19.25)	34.70
BS	5.72 \pm 1.42 (0.44)	27.15 \pm 1.41 (14.29)	94.29 \pm 1.06 (5.71)	87.45 \pm 1.06 (6.93)	6.84	29.75 \pm 2.39 (29.75)	74.88 \pm 3.13 (74.88)	78.34 \pm 0.96 (21.66)	72.07 \pm 1.25 (22.59)	37.22
SalUn	1.48 \pm 0.14 (3.80)	16.19 \pm 0.34 (3.33)	99.98 \pm 0.01 (0.02)	93.95 \pm 0.01 (0.43)	1.89	0.96 \pm 0.59 (0.96)	96.43 \pm 0.31 (96.43)	99.98 \pm 0.01 (0.02)	94.03 \pm 0.08 (0.63)	24.51

of the unlearned model, as indicated by RA and TA, shows no loss when comparing unlearning on worst-case forget sets to random forget sets. Intriguingly, the TA of models unlearned with the worst-case forget set may even surpass those unlearned with random sets, hinting at a connection to coreset selection that will be further explored later. In Appendix C.4, we broaden our evaluation of the worst-case forget set across additional datasets and model architectures. The findings align with those presented in Table 2.

Reassessing approximate unlearning under worst-case forget set. Next, we examine the performance of approximate unlearning methods (FT, EU-*k*, CF-*k*, SCRUB, ℓ_1 -sparse, RL, BE, BS, and SalUn) when applied to the worst-case forget set scenario. Ideally, an effective and robust approximate unlearning method should mirror the trend of Retrain, *i.e.*, maintaining a minimal performance discrepancy with exact unlearning. However, evaluations using worst-case forget sets can reveal performance disparities in approximate unlearning (vs. Retrain), which differ from those observed with random forget sets.

In Table 3, we present the performance of approximate unlearning methods under both random and worst-case forget sets, with the forgetting data ratio 10%. For comparison, we also include the performance of Retrain and analyze the performance gap between approximate unlearning methods and Retrain (see the column ‘Avg. Gap’). Note that an ideal approximate unlearning method is expected to have a smaller performance gap with Retrain. Following Sec. 3, we categorize approximate unlearning methods into two groups: relabeling-free and relabeling-based, respectively. We highlight two main observations from Table 3.

First, relabeling-free approximate unlearning methods (FT, EU-*k*, CF-*k*, SCRUB, ℓ_1 -sparse) generally follow the trend of Retrain when evaluated on worst-case forget sets (*i.e.*, inadequately erasure the influence of the worst-case forget set). This is evidenced by a significant decrease in UA and MIA, along with a narrowing performance gap with Retrain. Moreover, consistent with Table 2, evaluations using the worst-case forget set reduce the high variance encountered in random data forgetting, as demonstrated by the SCRUB method.

Second, different from relabeling-free methods, relabeling-based approximate unlearning (RL, BE, BS, SalUn) exhibit a significant performance gap compared

to Retrain. This gap is highlighted by an increase in UA for methods like BE and BS when applied to worst-case forget sets, diverging from the performance of Retrain. A similar discrepancy is noted in MIA. Crucially, all relabeling-based approaches seem to offer a *false* sense of unlearning effectiveness under worst-case forget set, as evidenced by a marked rise in MIA, suggesting the inefficacy of relabeling-based strategies in this context. Recall that relabeling-based methods achieve unlearning by explicitly negating the forget set through relabeling. Therefore, this approach can cause significant changes in model behavior, especially when the forget set mainly consists of common-case data points.

Analyzing the selected worst-case forget set through a coreset lens.

Our prior observation from Table 2 indicates that TA of models unlearned with worst-case forget sets could surpass those subjected to random forget sets. This prompts us to investigate the characteristics of data typically selected within the worst-case forget sets. In what follows, we explore this question through the lens of *coreset*, given by a subset of the training data deemed sufficient for model training [5, 43, 75]. We hypothesize that the data points included in the worst-case forget set likely do *not* constitute the

coreset, as forgetting them poses challenges, possibly owing to their strong inherent correlation with data not selected for forgetting. Motivated by this hypothesis, we explore whether the *complement* of the worst-case forget set corresponds to the *coreset*. **Fig. 3** contrasts the testing accuracy of an image classifier (ResNet-18) trained on the complement of the worst-case forget set (termed ‘Worst’) against that trained on coresets identified through other coreset selection methods, including **EL2N** and **GraNd** [47]. For comparison, we also present the performance using the model trained on the original full dataset (termed ‘Original’) and the random set (termed ‘Random’). Training models on the complement of the worst-case forget set achieve TA comparable to the state-of-the-art coreset selection methods across various data selection ratios for both CIFAR-10 and CIFAR-100 datasets. Furthermore, TA achieved by using the complement of the worst-case forget set on CIFAR-100 can even exceed the performance of the original model trained on the entire dataset at 90% and 95% selection ratios (*i.e.*, 10% and 5% forgetting ratios). These observations suggest that the chosen worst-case forget set indeed does *not* constitute a coreset, whereas its complement serves as a coreset. From the coreset analysis perspective, identifying the worst-case forget set not only addresses the most challenging data to forget but also offers a method to *attribute data influence* in model training based on their ‘unlearning difficulty’ levels.

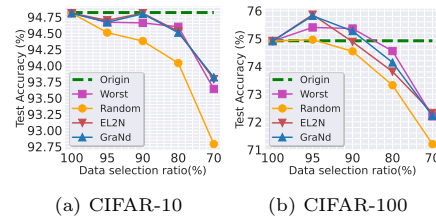


Fig. 3: Performance of ResNet-18 trained on coresets of (a) CIFAR-10 and (b) CIFAR-100, determined by different approaches, including the complement of worst-case forget set (Worst), random select (Random), EL2N and GraNd, at varying coreset ratios. The dashed line represents the model’s performance trained on the full dataset (Origin).

Case study: Selecting the worst-case forget set in a biased dataset.

The previous experiment results suggest that the identified worst-case forget set corresponds to the complement of the coreset (*i.e.*, the non-coreset). We further explore this intriguing finding through a case study, identifying the worst-case forget set in a biased dataset created from CelebA [42]. We consider this dataset for hair color prediction (Blond vs. Non-Blond), with a spurious correlation with the ‘gender’ attribute (Male vs. Female) [51]. **Fig. 4** presents the composition of the selected worst-case forget set (with the data forgetting ratio 10%), categorized into four groups based on the combination of the label and the ‘gender’ attribute. As we can see, within the chosen worst-case forget set, there exists a large portion of data points associated with (Blond + Female). It’s worth noting that in CelebA, blonde hair is commonly correlated with females, making data points in the (Blond + Female) group relatively *easy to learn*, acting as a non-coreset if forgetting part of the data points from this group.

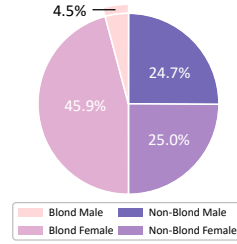


Fig. 4: Composition of the worst-case forget set under CelebA.

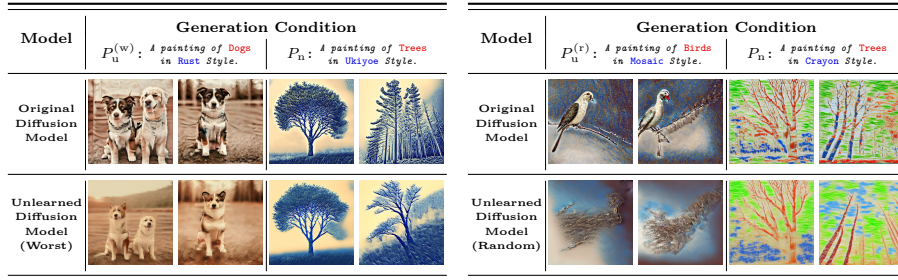


Fig. 5: Examples of image generation using the original SD model (w/o unlearning), the unlearned SD over the worst-case forgetting prompt set (Worst), and the unlearned SD over the random forget set (Random). For each diffusion model, images are generated based on two conditions, an unlearned prompt ($P_u^{(w)}$ or $P_u^{(r)}$) and an unlearning-irrelevant normal prompt (P_n). Here $P_u^{(w)}$ and $P_u^{(r)}$ indicate the prompt drawn from the worst-case forget set and the random forget set, respectively.

An extended study: Worst-case forget set on prompt-wise unlearning.

Extending from data-wise forgetting, we further demonstrate the efficacy of our approach in prompt-wise forgetting for text-to-image generation. We utilize the stable diffusion (SD) model [49] on the UnlearnCanvas dataset, a benchmark dataset designed to evaluate the unlearning of painting styles and objects [76]. In UnlearnCanvas, a text prompt used as the condition of image generation is given by ‘A painting of [Object Name] in [Style Name] Style.’ We considered 10 objects and 10 styles (100 combinations in total) for prompts and selected 10% of them to unlearn. For designated prompts targeted for unlearning, we apply the Erased Stable Diffusion (ESD) [21] technique; See Appendix B.2 for further implementation details. **Fig. 5** presents examples of images generated using the pre-trained SD model, the unlearned model by forgetting a random prompt set, and the unlearned model by forgetting the identified worst-case for-

get set. For each model, images are generated under two conditions, given from (1) the unlearned prompt set $P_u^{(w/r)}$ where (w) or (r) indicates the worst-case or the random forget set, and (2) the normal prompt set P_n irrelevant to forget sets. As we can see, the unlearned diffusion model is *unable to prevent* image generation based on prompts from the worst-case forget set ($P_u^{(w)}$), resulting in similar image outputs to those of the original diffusion model. In contrast, the diffusion model unlearned through random forgetting can avoid generating accurate images based on the unlearning-targeted prompts ($P_u^{(r)}$) from the random forget set, displaying a significant deviation from the original model’s outputs, indicating successful unlearning. Furthermore, when conditioned on the normal, forgetting-irrelevant prompts (P_n), both worst-case and random forgetting-oriented diffusion models perform well, generating the requested images. The above results indicate that erasing the influence of prompts from the worst-case forget set introduces new challenges of MU for image generation. We refer readers to Appendix A.2 for more visualizations.

Additional results. In Appendix C.7, we examine the uniqueness of the worst-case forget set by mixing it with other randomly selected data points for unlearning. We also demonstrate the effectiveness of worst-case forget set in the scenario of *class-wise* forgetting on ImageNet [17] in Appendix C.5.

6 Conclusion and Discussion

In this study, we delved into the challenge of pinpointing the worst-case forget set in MU, introducing a fresh perspective that broadens the scope and enhances the effectiveness of MU beyond conventional methods like random data forgetting. By employing BLO, we developed a structured approach to accurately identify these pivotal sets. Through extensive experiments, we demonstrated the effectiveness of our proposed method in different data-model setups, showcasing its significance for improved reliability in MU evaluations.

Although our worst-case performance assessment was inspired by the lack of robustness in random data forgetting, it also deepens the understanding of when MU becomes ‘easy’ or ‘difficult’ and the underlying reasons from a data selection-based interpretability perspective. Our results further encourage rethinking the role of data difficulty in unlearning. For example, incorporating a curriculum based on these difficulty levels may significantly impact unlearning performance. We term the incorporation of curriculum learning [3] into MU as *curriculum unlearning*, which may show promise in improving unlearning effectiveness. Additionally, the process of identifying the worst-case forget set offers a way to attribute data influence by evaluating their unlearning difficulty. In this work, coreset selection emerges as a byproduct of this data attribution process, based on the assessment of unlearning difficulty levels. Furthermore, the inability to retrain from scratch to unlearn the identified challenging forget set prompts a reevaluation of its appropriateness in defining ‘exact’ unlearning.

Acknowledgement

This research is supported by the ARO Award W911NF2310343. Additionally, the work of C. Fan, J. Liu, and S. Liu is partially supported by the NSF Grant IIS-2207052, and the work of Alfred Hero is partially supported by NSF-2246157.

References

1. Achille, A., Kearns, M., Klingenberg, C., Soatto, S.: Ai model disgorgement: Methods and choices. arXiv preprint arXiv:2304.03545 (2023)
2. Becker, A., Liebig, T.: Evaluating machine unlearning via epistemic uncertainty. arXiv preprint arXiv:2208.10836 (2022)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 41–48 (2009)
4. Bernstein, J., Wang, Y.X., Azizzadenesheli, K., Anandkumar, A.: signsgd: Compressed optimisation for non-convex problems. In: International Conference on Machine Learning. pp. 560–569. PMLR (2018)
5. Borsos, Z., Mutny, M., Krause, A.: Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems* **33**, 14879–14890 (2020)
6. Bourtole, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 141–159. IEEE (2021)
7. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: 2015 IEEE Symposium on Security and Privacy. pp. 463–480. IEEE (2015)
8. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramer, F.: Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1897–1914. IEEE (2022)
9. Che, T., Zhou, Y., Zhang, Z., Lyu, L., Liu, J., Yan, D., Dou, D., Huan, J.: Fast federated machine unlearning with nonlinear functional theory (2023)
10. Chen, M., Gao, W., Liu, G., Peng, K., Wang, C.: Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7766–7775 (2023)
11. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., Zhang, Y.: Graph unlearning. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 499–513 (2022)
12. Chen, R., Yang, J., Xiong, H., Bai, J., Hu, T., Hao, J., Feng, Y., Zhou, J.T., Wu, J., Liu, Z.: Fast model debias with machine unlearning. *Advances in Neural Information Processing Systems* **36** (2024)
13. Cheng, J., Dasoulas, G., He, H., Agarwal, C., Zitnik, M.: Gnndelete: A general strategy for unlearning in graph neural networks. arXiv preprint arXiv:2302.13406 (2023)
14. Chien, E., Pan, C., Milenkovic, O.: Certified graph unlearning. arXiv preprint arXiv:2206.09140 (2022)
15. Coleman, C., Yeh, C., Musmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., Zaharia, M.: Selection via proxy: Efficient data selection for deep learning. arXiv preprint arXiv:1906.11829 (2019)

16. Cotogni, M., Bonato, J., Sabetta, L., Pelosin, F., Nicolosi, A.: Duck: Distance-based unlearning via centroid kinematics. *arXiv preprint arXiv:2312.02052* (2023)
17. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
18. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 486–503. Springer (2006)
19. Eldan, R., Russinovich, M.: Who’s harry potter? approximate unlearning in llms (2023)
20. Fan, C., Liu, J., Zhang, Y., Wei, D., Wong, E., Liu, S.: Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508* (2023)
21. Gandikota, R., Materzynska, J., Fiotto-Kaufman, J., Bau, D.: Erasing concepts from diffusion models. *arXiv preprint arXiv:2303.07345* (2023)
22. Gandikota, R., Orgad, H., Belinkov, Y., Materzyńska, J., Bau, D.: Unified concept editing in diffusion models. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5111–5120 (2024)
23. Ginart, A., Guan, M., Valiant, G., Zou, J.Y.: Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems* **32** (2019)
24. Goel, S., Prabhu, A., Sanyal, A., Lim, S.N., Torr, P., Kumaraguru, P.: Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640* (2022)
25. Golatkar, A., Achille, A., Soatto, S.: Eternal sunshine of the spotless net: Selective forgetting in deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9304–9312 (2020)
26. Graves, L., Nagisetty, V., Ganesh, V.: Amnesiac machine learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11516–11524 (2021)
27. Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L.: Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030* (2019)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
29. Heng, A., Soh, H.: Selective amnesia: A continual learning approach to forgetting in deep generative models (2023)
30. Hoofnagle, C.J., van der Sloot, B., Borgesius, F.Z.: The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law* **28**(1), 65–98 (2019)
31. Huggins, J., Campbell, T., Broderick, T.: Coresets for scalable bayesian logistic regression. *Advances in neural information processing systems* **29** (2016)
32. Izzo, Z., Smart, M.A., Chaudhuri, K., Zou, J.: Approximate data deletion from machine learning models. In: International Conference on Artificial Intelligence and Statistics. pp. 2008–2016. PMLR (2021)
33. Jia, J., Liu, J., Ram, P., Yao, Y., Liu, G., Liu, Y., Sharma, P., Liu, S.: Model sparsity can simplify machine unlearning. *Advances in neural information processing systems* **36** (2023)
34. Kim, S., Bae, S., Yun, S.Y.: Coreset sampling from open-set for fine-grained self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7537–7547 (2023)

35. Krantz, S.G., Parks, H.R.: The implicit function theorem: history, theory, and applications. Springer Science & Business Media (2002)
36. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
37. Kumari, N., Zhang, B., Wang, S.Y., Shechtman, E., Zhang, R., Zhu, J.Y.: Ablating concepts in text-to-image diffusion models (2023)
38. Kurmanji, M., Triantafillou, P., Triantafillou, E.: Towards unbounded machine unlearning. arXiv preprint arXiv:2302.09880 (2023)
39. Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Xu, X., Yao, Y., Li, H., Varshney, K.R., et al.: Rethinking machine unlearning for large language models. arXiv preprint arXiv:2402.08787 (2024)
40. Liu, Y., Fan, M., Chen, C., Liu, X., Ma, Z., Wang, L., Ma, J.: Backdoor defense with machine unlearning. arXiv preprint arXiv:2201.09538 (2022)
41. Liu, Y., Xu, L., Yuan, X., Wang, C., Li, B.: The right to be forgotten in federated learning: An efficient realization with rapid retraining. arXiv preprint arXiv:2203.07320 (2022)
42. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
43. Mirzasoleiman, B., Bilmes, J., Leskovec, J.: Coresets for data-efficient training of machine learning models. In: International Conference on Machine Learning. pp. 6950–6960. PMLR (2020)
44. Neel, S., Roth, A., Sharifi-Malvajerdi, S.: Descent-to-delete: Gradient-based methods for machine unlearning. In: Algorithmic Learning Theory. pp. 931–962. PMLR (2021)
45. Nguyen, T.T., Huynh, T.T., Nguyen, P.L., Liew, A.W.C., Yin, H., Nguyen, Q.V.H.: A survey of machine unlearning. arXiv preprint arXiv:2209.02299 (2022)
46. Oesterling, A., Ma, J., Calmon, F.P., Lakkaraju, H.: Fair machine unlearning: Data removal while mitigating disparities. arXiv preprint arXiv:2307.14754 (2023)
47. Paul, M., Ganguli, S., Dziugaite, G.K.: Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems* **34**, 20596–20607 (2021)
48. Pruthi, G., Liu, F., Kale, S., Sundararajan, M.: Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems* **33**, 19920–19930 (2020)
49. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
50. Rosen, J.: The right to be forgotten. *Stan. L. Rev. Online* **64**, 88 (2011)
51. Sagawa, S., Raghunathan, A., Koh, P.W., Liang, P.: An investigation of why overparameterization exacerbates spurious correlations. In: International Conference on Machine Learning. pp. 8346–8356. PMLR (2020)
52. Sattigeri, P., Ghosh, S., Padhi, I., Dognin, P., Varshney, K.R.: Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. In: *Advances in Neural Information Processing Systems* (2022)
53. Schioppa, A., Zablotskaia, P., Vilar, D., Sokolov, A.: Scaling up influence functions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8179–8186 (2022)
54. Sekhari, A., Acharya, J., Kamath, G., Suresh, A.T.: Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems* **34**, 18075–18086 (2021)

55. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489 (2017)
56. Shaban, A., Cheng, C.A., Hatch, N., Boots, B.: Truncated back-propagation for bilevel optimization. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1723–1732. PMLR (2019)
57. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
58. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
59. Song, L., Mittal, P.: Systematic evaluation of privacy risks of machine learning models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2615–2632 (2021)
60. Thudi, A., Deza, G., Chandrasekaran, V., Papernot, N.: Unrolling sgd: Understanding factors influencing machine unlearning. In: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P). pp. 303–319. IEEE (2022)
61. Thudi, A., Jia, H., Shumailov, I., Papernot, N.: On the necessity of auditable algorithmic definitions for machine unlearning. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 4007–4022 (2022)
62. Ullah, E., Mai, T., Rao, A., Rossi, R.A., Arora, R.: Machine unlearning via algorithmic stability. In: Conference on Learning Theory. pp. 4126–4142. PMLR (2021)
63. Wang, J., Guo, S., Xie, X., Qi, H.: Federated unlearning via class-discriminative pruning. In: Proceedings of the ACM Web Conference 2022. pp. 622–632 (2022)
64. Warnecke, A., Pirch, L., Wressnegger, C., Rieck, K.: Machine unlearning of features and labels. arXiv preprint arXiv:2108.11577 (2021)
65. Wu, L., Guo, S., Wang, J., Hong, Z., Zhang, J., Ding, Y.: Federated unlearning: Guarantee the right of clients to forget. IEEE Network **36**(5), 129–135 (2022)
66. Wu, X., Li, J., Xu, M., Dong, W., Wu, S., Bian, C., Xiong, D.: Depn: Detecting and editing privacy neurons in pretrained language models. arXiv preprint arXiv:2310.20138 (2023)
67. Xia, X., Liu, J., Yu, J., Shen, X., Han, B., Liu, T.: Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In: The Eleventh International Conference on Learning Representations (2022)
68. Yang, S., Xie, Z., Peng, H., Xu, M., Sun, M., Li, P.: Dataset pruning: Reducing training data by examining generalization influence. arXiv preprint arXiv:2205.09329 (2022)
69. Yao, Y., Xu, X., Liu, Y.: Large language model unlearning. arXiv preprint arXiv:2310.10683 (2023)
70. Yu, C., Jeoung, S., Kasi, A., Yu, P., Ji, H.: Unlearning bias in language models by partitioning gradients. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 6032–6048 (2023)
71. Zeng, Y., Pan, M., Jahagirdar, H., Jin, M., Lyu, L., Jia, R.: How to sift out a clean data subset in the presence of data poisoning? arXiv preprint arXiv:2210.06516 (2022)
72. Zhang, E., Wang, K., Xu, X., Wang, Z., Shi, H.: Forget-me-not: Learning to forget in text-to-image diffusion models. arXiv preprint arXiv:2303.17591 (2023)
73. Zhang, J., Chen, S., Liu, J., He, J.: Composing parameter-efficient modules with arithmetic operations. arXiv preprint arXiv:2306.14870 (2023)
74. Zhang, Y., Khanduri, P., Tsaknakis, I., Yao, Y., Hong, M., Liu, S.: An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. arXiv preprint arXiv:2308.00788 (2023)

- 75. Zhang, Y., Zhang, Y., Chen, A., Liu, J., Liu, G., Hong, M., Chang, S., Liu, S., et al.: Selectivity drives productivity: Efficient dataset pruning for enhanced transfer learning. *Advances in Neural Information Processing Systems* **36** (2024)
- 76. Zhang, Y., Zhang, Y., Yao, Y., Jia, J., Liu, J., Liu, X., Liu, S.: Unlearncanvas: A stylized image dataset to benchmark machine unlearning for diffusion models. *arXiv preprint arXiv:2402.11846* (2024)
- 77. Zhang, Y., Jia, J., Chen, X., Chen, A., Zhang, Y., Liu, J., Ding, K., Liu, S.: To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. *arXiv preprint arXiv:2310.11868* (2023)