Unlearning From Weakly Supervised Learning

Yi Tang[1] , **Yi Gao**[2 *] , **Yong-Gang Luo**[3] , **Ju-Cheng Yang**[3] , **Miao Xu**[4] , **Min-Ling Zhang**[5]
[1]School of Automation, Southeast University, China
[2]School of Cyber Science and Engineering, Southeast University, China
[3]AI LAB, Chongqing Changan Automobile Co. Ltd. [4]University of Queensland, Australia
[5]School of Computer Science and Engineering,Southeast University, China
{tangy, gao_yi, zhangml}@seu.edu.cn, {luoyg3, yangjc3}@changan.com.cn, miao.xu@uq.edu.au

Abstract

Machine unlearning provides users with the right to remove their privacy data from a well-trained model. Existing approaches of machine unlearning mainly focus on exploring data removing within *supervised learning* (SL) tasks. However, *weakly supervised learning* (WSL) is more applicable to real-world scenarios since collecting WSL data is less laborious than collecting fully supervised data. In this paper, we first propose a machine unlearning approach for WSL by updating the model parameters. Motivated by the uniform distribution of untrained model predictions, we derive a formulated target to force the model's predictions of removed data to be indistinguishable. This encourages the model to forget its ability to recognize features of data slated for unlearning. Moreover, we employ formulated targets to transform the classification unlearning into the convex regression, which can significantly reduce computational cost and avoid extra information storage during the training process. Additionally, we discuss how to design a target to ensure the models' predictions of removed data being indistinguishable in different learning scenarios, e.g., SL or WSL. As the flexibility in formulating targets, the proposed approach effectively deals with the WSL problem while still excels in SL models. Empirical studies show the superiority of the proposed approach.

1 Introduction

Organizations and companies extensively leverage user data for training machine learning models across various applications, involving movie recommendations or healthcare, etc [Sekhari *et al.*, 2021a]. As increasing concerns about the misuse of privacy data, especially sensitive information like personal emails and medical records [Bourtoule *et al.*, 2021], more and more countries place a greater emphasis on privacy data protection [Mantelero, 2013; Proserpio *et al.*, 2014; Shokri *et al.*, 2017]. Corresponding regulations empower individuals to revoke their authorization for the use of their data in data analysis and *machine learning* (ML) model training [State of California Department of Justice, 2023]. This arouses new discussions on data privacy and ownership [Shintre *et al.*, 2019], and promotes the emergence of machine unlearning that focuses on forgetting data [Bourtoule *et al.*, 2021; Nguyen *et al.*, 2022].

Existing approaches of machine unlearning can be roughly divided into two categories: exact unlearning [Golatkar *et al.*, 2020] and approximate unlearning [Thudi *et al.*, 2022]. Among them, exact unlearning typically expects that the distribution of the unlearned model is exactly the same as the retraining one. Intuitively, the straightforward approach is to retrain the model from scratch using the data that excludes the information to be unlearned [Yan *et al.*, 2022]. However, the retraining way comes with a substantial computational cost that is prohibitively expensive [Bourtoule *et al.*, 2021; Xu *et al.*, 2024]. To alleviate this problem, many researchers turn to explore alternative machine unlearning strategy: approximate unlearning [Wu *et al.*, 2022; Guo *et al.*, 2020]. Approximate unlearning aims to achieve a similar effect of retraining by modifying model parameters to remove the influence of specific data on parameter updates [Yan *et al.*, 2022; Wu *et al.*, 2022]. Compared to approaches belonging to exact unlearning, approaches of approximate unlearning are less computationally expensive and more efficient for data unlearning [Guo *et al.*, 2020; Golatkar *et al.*, 2020; Huang *et al.*, 2021].

Despite machine unlearning brings a novel inspiration to address the challenge of data unlearning [Sekhari *et al.*, 2021b], existing machine unlearning approaches both explore the data removal within the *supervised learning* (SL) tasks [Xu *et al.*, 2024; Yan *et al.*, 2022; Wu *et al.*, 2022]. In SL tasks, collecting extensive data with high-quality labels is typically required, while annotating large-scale datasets accurately is costly and time-consuming [Gao and Zhang, 2021; Tang *et al.*, 2024]. Due to collecting *weakly supervised learning* (WSL) data is less laborious than SL one, the WSL paradigm may be more applicable to real-world scenarios, which allows the model to learn from the data with imprecise or incomplete supervised information [Zhou, 2018; Gao *et al.*, 2023]. The goal of WSL is similar to that of SL, but approaches in WSL are greater flexibility and unrestricted by the quality of data annotation. Since ground-truth labels are unavailable for the training data in the WSL

[*] Corresponding author

paradigm, previous machine unlearning approaches cannot hold the weakly supervised data.

In this paper, we first propose an approach called *Uniform Distribution-guided Regression Unlearning* (UDRU) for the WSL problem to achieve data unlearning by updating the model parameters, which can effectively deal with the uncertainty labeling information of WSL data and also excel in the unlearning of supervised learning models. Naturally, when a trained model is asked to forget specific data, the unlearned model's predictions of these data should become indistinguishable or incorrect. Observed by that an untrained model tends to make predictions following uniform distributions. Therefore, we expect the model predictions of removal data to lie in intermediate after unlearning, since the model may still exhibit a memory of removal data when these prediction probabilities are either excessively high or low. To modify the learned output, we design a flexible unlearning target to force predictions of removal data to become indistinguishable by minimizing the divergence between probability distributions of model predictions and a uniform distribution. Different from some previous approaches relying on training data, the proposed approach applies formulated targets to replace classification unlearning as convex regression, which makes a model implement data unlearning without additional training data. Furthermore, the convex regression contributes to reducing computational cost and smoother convergence. UDRU can effectively tackle the WSL problem and still excel in SL models as the flexibility in formulating targets of our approach. The main contributions are summarized as follows:

- We propose a machine unlearning approach called UDRU for the WSL problem by formulating a flexible target based on a uniform distribution. To the best of our knowledge, moreover, our work explores machine unlearning in the WSL problem for the first time.
- UDRU transforms the process of classification unlearning into convex regression by the formulated targets, which significantly reduces computational cost and eliminates the need for storing extra training data or information during the training process.
- As the flexibility in formulating targets, UDRU is not only suitable for WSL but also well-adapted on the SL tasks. Empirical studies on various learning paradigms across different model scales show the effectiveness of the proposed approach.

The rest of this paper is organized as follows. In Section 2, we review related work and then introduce the proposed approach in Section 3. Experimental results and conclusion are presented in Section 4 and Section 5, respectively.

2 Related Work

In this section, we will give a brief review of related work of machine unlearning and the WSL paradigm, including *partial label learning* (PLL) and *noisy label learning* (NLL).

2.1 Machine Unlearning

Before proceeding, we provide a brief overview of notations and existing approaches in machine unlearning. To formulate machine unlearning, suppose $\mathcal{X} \subset \mathbb{R}^d$ represents the feature space with d dimensions, and $\mathcal{Y} = \{1, 2, \ldots, K\}$ is the label space with K possible labels. Let $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_t}$ be the training dataset with N_t instances, where $y_i \in \mathcal{Y}$ is the ground-truth label of $\boldsymbol{x}_i \in \mathcal{X}$. We define $\mathcal{D}_u \subset \mathcal{D}_t$ as the unlearning dataset and $\mathcal{D}_r = \mathcal{D}_t \setminus \mathcal{D}_u$ as the remaining dataset. Given a mapping model $\boldsymbol{f} : \mathcal{X} \rightarrow \mathbb{R}^K$ and $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$, the model parameters $\boldsymbol{\theta}^*$ are trained using a learning algorithm $\mathcal{A}(\cdot)$, denoted as $\boldsymbol{\theta}^* = \mathcal{A}(\mathcal{D}_t)$. To normalize $\boldsymbol{z}$ into a probability distribution $\boldsymbol{p}$ that satisfies $\sum_{i=1}^{K} p_i = 1$, we apply a softmax function $\boldsymbol{\sigma}$. The i-th element of $\boldsymbol{p}$ is expressed as:

$$p_i = \boldsymbol{\sigma}_i(\boldsymbol{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{1}$$

where z_i denotes the i-th element of $\boldsymbol{z}$. As recent line of work in machine unlearning can be divided into two categories: exact unlearning [Ginart *et al.*, 2019; Brophy and Lowd, 2021] and approximate unlearning [Guo *et al.*, 2020; Wu *et al.*, 2022]. Exact unlearning approaches expect that the parameter distribution of the unlearning model is identical to that of the retraining model. This can be formulated as:

$$\mathcal{K}(P(\mathcal{U}(\mathcal{A}(\mathcal{D}_t), \mathcal{D}_t, \mathcal{D}_u) \in \mathcal{T}), P(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T})) = 0, \tag{2}$$

where $\mathcal{U}(\cdot)$ refers to an exact unlearning process, and $\mathcal{T} \subseteq \mathcal{H}$ denotes the hypothesis space $\mathcal{H}$ after unlearning parameters $\boldsymbol{\theta}$. $P(\cdot)$ and $\mathcal{K}(\cdot)$ denote the distribution of parameters and a distribution measurement of KL-divergence, respectively. One representative approach for exact unlearning is ARCANE [Yan *et al.*, 2022], which divided the training data according to labels and trained a one-class classifier for each subset. Here, unlearning specific data only required retraining the one-class classifier corresponding to the labels of unlearning data. However, exact unlearning approaches face challenges in efficiently handling with massive data due to the reliance of the entire training data.

Consequently, many researchers have turned their attention to approximate unlearning to reduce the computational cost of data unlearning. Unlike exact unlearning, approximate unlearning tolerates the divergence between $P(\mathcal{U}(\mathcal{A}(\mathcal{D}_t), \mathcal{D}_t, \mathcal{D}_u) \in \mathcal{T})$ and $P(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T})$ within specified thresholds, i.e.,

$$e^{-\epsilon} \leq \frac{P(\mathcal{U}(\mathcal{A}(\mathcal{D}_t), \mathcal{D}_t, \mathcal{D}_u) \in \boldsymbol{\mathcal{T}})}{P(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T})} \leq e^{\epsilon}. \tag{3}$$

Eq. (3) forms the basis for a series of approximate unlearning approaches, including but not limited to, *model-agnostic algorithms* [Chundawat *et al.*, 2023], *model-intrinsic algorithms* [Baumhauer *et al.*, 2022; Izzo *et al.*, 2021] and *data-driven algorithms* [Huang *et al.*, 2021; Shen *et al.*, 2024b; Shen *et al.*, 2024a]. It is important to note that existing machine unlearning approaches cannot work well on WSL data as they are designed for fully supervised tasks.

2.2 Weakly Supervised Learning

Differing from fully SL tasks that require massive data with accurate supervision information, WSL provides the flexibility of learning with weak supervision. In this paper, we mainly explore machine unlearning for PLL and NLL in WSL. We proceed to introduce these two learning problems as follows.

Partial Label Learning. In the case of PLL, each training instance is associated with a candidate label set, only one of which is the ground-truth label. Let $\mathcal{D}_p = \{(\boldsymbol{x}_i, s_i)\}_{i=1}^{N_p}$ be the training dataset including N_p instances for PLL, where $s_i \in \{2^{\mathcal{Y}} \setminus \emptyset\}$ is the set of candidate labels for $\boldsymbol{x}_i$. PLL research follows two main strategies: the *average-based strategy* (ABS) [Zhou and Gu, 2018] and the *identification-based strategy* (IBS) [Feng and An, 2019a; Wen *et al.*, 2021]. ABS-based approaches treat all labels within the candidate label set equally [Cour *et al.*, 2011]. However, they may suffer from low accuracy since their outputs often overwhelm the ground-truth label [Shi *et al.*, 2023]. Hence, many IBS-based approaches have emerged, which aim to purify each candidate label set and heuristically explore the ground-truth label during the learning process [Cour *et al.*, 2011; Feng and An, 2019b].

Noisy Label Learning. With the advent of highly-curated datasets, deep neural networks have demonstrated promising performance on various classification tasks [Krizhevsky *et al.*, 2012; Noh *et al.*, 2015]. In fact, real-world data is inherently imperfect, which inevitably introduces corruptions commonly known as noise [Zhang and Sabuncu, 2018; Wei *et al.*, 2022]. The training dataset for NLL is denoted as $\mathcal{D}_n = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{N_n}$, where an instance $(\boldsymbol{x}, \tilde{y})$ has the label $\tilde{y}$ that does not align with the ground-truth label y. Previous studies have demonstrated that noisy labels pose challenges for overparameterized neural networks, which results in overfitting and performance degradation [Arpit *et al.*, 2017; Zhang *et al.*, 2017]. Therefore, noise-robust algorithms are proposed, which develop loss functions that can tolerate noisy labels [Zhang and Sabuncu, 2018]. Additionally, certain approaches focus on explicitly correcting the loss function by estimating the noise transition matrix to solve the NLL problem [Patrini *et al.*, 2017]. In the following section, we will illustrate UDRU how to implement machine unlearning on WSL and SL paradigms.

3 The Proposed Approach

In this section, we introduce a novel approach called UDRU, which solves machine unlearning by formulating an unlearning target adhering to a uniform distribution. Through analyzing the outputs of different learning paradigms, we formulate distinct unlearning targets tailored to each learning paradigm. We theoretically prove the feasibility of these unlearning targets. Moreover, UDRU employs regression to solve classification unlearning tasks, which significantly reduces computational cost.

3.1 Unlearning in Supervised Learning

Given a SL model $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}^*)$ learned with a learning algorithm $\mathcal{A}(\cdot)$ on the training dataset $\mathcal{D}_t$, the goal of machine unlearning is to remove the unlearning dataset $\mathcal{D}_u \subset \mathcal{D}_t$ from the model through updating the model's parameters from $\boldsymbol{\theta}^*$ to $\boldsymbol{\theta}_u$ without seriously degrading its prediction performance. Evaluating the efficacy of the unlearning process is crucial for an unlearning approach, which reflects the expected effect of an unlearning approach. Unfortunately, we find that using Eq. (2) or Eq. (3) for a consistent evaluation of unlearning is challenging, because the training dataset may not be available in many cases. Hence, we are eager to design an evaluation measure without the restriction on the availability of the training dataset.

In experiments, we can observe that predictions from an untrained model tend to follow uniform distributions when it is used to assign the ground-truth labels for unseen instances. Conversely, a well-trained model provides distinct probability predictions for the features of input instances. This implies that a trained model, after removing specific data, is expected to exhibit similar predictions for these specific data as an untrained model. Motivated by this, we design an evaluation measure that gets rid of the reliance on the availability of the training dataset. Here, we will investigate the evaluation measure for machine unlearning from the definition of a uniform distribution as follows.

Definition 1 (Uniform Distribution)**.** *Let $\bar{\boldsymbol{p}} \in \mathbb{R}^K$ be a uniform distribution. When the number of labels is K, the i-th element of $\bar{\boldsymbol{p}}$ satisfies*

$$\bar{p}_i = \frac{1}{K}, \quad s.t. \;\; i \in \{1, 2, \ldots, K\}.$$

With only an unlearning dataset $\mathcal{D}_u$ available, we can substitute Eq. (2) or Eq. (3) with the divergence from an untrained model which has no bias for the features of unlearning data. The evaluation measure is implemented by minimizing the divergence between the uniform distribution $\bar{\boldsymbol{p}}$ and the output probability distribution $\boldsymbol{p}$. KL-Divergence has the ability to measure the difference between two distributions, which is adopted to facilitate the achievement of the evaluation measure. The evaluation measure is expressed as:

$$D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{p}) = \sum_{i=1}^{K} \bar{p}_i \ln \frac{\bar{p}_i}{p_i} = \sum_{i=1}^{K} \frac{1}{K} \ln \frac{\frac{1}{K}}{p_i}. \tag{4}$$

Next, we will introduce how to formulate an unlearning target in SL with the evaluation measure. In the process of unlearning specific data, our goal is to diminish the model's bias towards a particular class label while maintaining the performance of the output for other class labels. For this reason, we expect to derive an unlearning target whose prediction has the minimal divergence with predictions of an untrained model and does not influence the outputs of other classes. Due to the structures of neural networks, there are numerous and separated parameters that are only sensitive to relevant features. Therefore, we can assume that the unlearning process has little effect on the i-th output z_i $(i \neq y)$ of the model when the scale of the model is sufficient and relatively irrelevant parameters are not responsible for the unlearning modification. According to this assumption, we only need to keep the outputs corresponding to all labels except the ground-truth one unchanged and assign a moderate value to the output associated with the ground-truth label. Obviously, the unlearning target $\boldsymbol{t} \in \mathbb{R}^K$ should satisfy the conditions shown as follows:

$$\begin{cases} t_i \approx z_i, & i \neq y \\ t_i = \operatorname*{arg\,min}_{z_i} D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{\sigma}(\boldsymbol{z})), & i = y \\ \min \boldsymbol{z} \leq t_i \leq \max \boldsymbol{z}, & i = y \end{cases} \tag{5}$$

where t_i denotes the i-th element of $\boldsymbol{t}$. To construct the unlearning target $\boldsymbol{t}$ under the conditions of Eq. (5), we derive a function $\boldsymbol{T}(\boldsymbol{z}, y)$ to automatically formulate $\boldsymbol{t}$ according to each instance $(\boldsymbol{x}, y)$, i.e., $\boldsymbol{t} = \boldsymbol{T}(\boldsymbol{z}, y)$. Theorem 2 shows the unlearning target function $\boldsymbol{T}(\boldsymbol{z}, y)$ that satisfies the above conditions.

Theorem 2. *To satisfy the conditions shown in Eq.* (5)*,* $\boldsymbol{T}(\boldsymbol{z}, y)$ *in SL is derived as:*

$$\boldsymbol{T}_i(\boldsymbol{z}, y) = \begin{cases} z_i, & i \neq y \\ \ln(\frac{1}{K-1} \sum_{j \neq y} e^{z_j}), & i = y \end{cases}$$

where $T_i(\boldsymbol{z}, y)$ *presents the* i*-th element of* $T(\boldsymbol{z}, y)$*.*

The proof is provided in Appendix A. With this formulated target, we can ensure that the modified predictions attain minimal divergence from the uniform distribution and maintain performance on the outputs except the ground-truth one. After obtaining this target, we derive a convex objective function by introducing a regularization term, which is defined as:

$$\begin{aligned} \mathcal{J}(\boldsymbol{\theta}, \boldsymbol{\theta}^*, \mathcal{D}_u) =& \frac{1}{|\mathcal{D}_u|} \sum_{i=1}^{|\mathcal{D}_u|} \|\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\theta}) - \boldsymbol{T}(\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\theta}^*), y_i)\|^2 \\ &+ \delta \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2, \end{aligned} \tag{6}$$

where $\boldsymbol{\theta}$ denotes the parameter variable that needs optimization, and δ is a trade-off parameter used to balance the strength of $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2$. Then, the unlearned parameters $\boldsymbol{\theta}_u$ can be obtained by optimizing the following equation:

$$\boldsymbol{\theta}_u = \arg\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}, \boldsymbol{\theta}^*, \mathcal{D}_u). \tag{7}$$

Building on this regression, UDRU effectively facilitates the unlearning of specific data while preserving performance on the remaining data. Moreover, the utilization of specific targets substantially reduces computational and time cost, leading to smooth and rapid convergence. δ is the only hyperparameter, which falls within a certain range to avoid a significant impact on the final convergence results.

3.2 Unlearning in Partial Label Learning

Suppose $\boldsymbol{\theta}^*$ is the parameters of a PLL model $\boldsymbol{f}_{PL}(\boldsymbol{x}, \boldsymbol{\theta}^*)$ learned from the PLL training dataset $\mathcal{D}_p$. The solution to the PLL problem is similar to that of the SL problem, which aims to derive $\boldsymbol{\theta}_u$ through constructing an unlearning target. Given an unlearning dataset $\mathcal{D}_u = \{(\boldsymbol{x}_i, s_i)\}_{i=1}^{N_u}$ and $\mathcal{D}_u \subset \mathcal{D}_p$, each candidate set s_i consists of a ground-truth label of $\boldsymbol{x}_i$ and incorrect labels. Due to the supervision information in PLL differing from SL, the formulated unlearning target and the corresponding conditions of PLL have also changed according to the weak supervision information. Theorem 3 shows the unlearning target of PLL and the associated conditions.

Theorem 3. *Let* $\boldsymbol{z} = \boldsymbol{f}_{PL}(\boldsymbol{x}, \boldsymbol{\theta})$ *and* z_i *denotes the* i*-th element of* $\boldsymbol{z}$*. The* i*-th element of the unlearning target* $\boldsymbol{T}(\boldsymbol{z}, s)$ *in PLL is expressed as:*

$$\boldsymbol{T}_i(\boldsymbol{z}, s) = \begin{cases} z_i, & i \notin s \\ \ln(\frac{1}{K-|s|} \sum_{j \notin s} e^{z_j}), & i \in s \end{cases}$$

which satisfies the following conditions:

$$\begin{cases} t_i \approx z_i, & i \notin s \\ t_i = \arg\min_{z_i, i \in s} D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{\sigma}(\boldsymbol{z})), & i \in s \\ \min \boldsymbol{z} \leq t_i \leq \max \boldsymbol{z}, & i \in s \end{cases}$$

The proof is stated in Appendix B. Theorem 3 displays the unlearning target of PLL, which is theoretically inferred by Definition 1 and satisfies certain conditions. Then, the optimized objective for machine unlearning in PLL is defined as:

$$\begin{aligned} \mathcal{J}_p(\boldsymbol{\theta}, \boldsymbol{\theta}^*, \mathcal{D}_u) = \frac{1}{|\mathcal{D}_u|} \sum_{i=1}^{|\mathcal{D}_u|} \|\boldsymbol{f}_{PL}(\boldsymbol{x}_i, \boldsymbol{\theta}) - \boldsymbol{T}(\boldsymbol{f}_{PL}(\boldsymbol{x}_i, \boldsymbol{\theta}^*), s_i)\|^2 \\ + \delta \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2 \end{aligned} \tag{8}$$

The unlearned parameters $\boldsymbol{\theta}_u$ can be acquired by minimizing the objective function:

$$\boldsymbol{\theta}_u = \arg\min_{\boldsymbol{\theta}} \mathcal{J}_p(\boldsymbol{\theta}, \boldsymbol{\theta}^*, \mathcal{D}_u) \tag{9}$$

These updated parameters will effectively remove the influence of the unlearning data from PLL models, while maintaining performance on the remaining data. Applying this unlearning approach enables effective handling of situations where the labels contain inaccurate information.

3.3 Unlearning in Noisy Label Learning

Machine unlearning in NLL shares similarities with the aforementioned learning paradigms, which aims to find $\boldsymbol{\theta}_u$ that achieves the ability of unlearning specific data. Given $\boldsymbol{\theta}^*$ as the parameters of a trained NLL model $\boldsymbol{f}_{NL}(\boldsymbol{x}, \boldsymbol{\theta})$, the unlearning dataset is denoted as $\mathcal{D}_u = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{N_u}$. In the NLL problem, the given label $\tilde{y}$ of an instance $\boldsymbol{x}$ has a chance of being corrupted. This means that $\tilde{y}$ may represent the ground-truth label of $\boldsymbol{x}$, or it may be a noisy label. Hence, we can encompass all scenarios when using the model to predict a label $\hat{y}$ for an instance, which are summarized as follows:

(1) $\hat{y} = \tilde{y}$, and $\tilde{y}$ is the ground-truth label of $\boldsymbol{x}$;

(2) $\hat{y} = \tilde{y}$, and $\tilde{y}$ is an incorrect label of $\boldsymbol{x}$;

(3) $\hat{y} \neq \tilde{y}$, and one of them is the ground-truth label of $\boldsymbol{x}$;

(4) $\hat{y} \neq \tilde{y}$, and are both incorrect.

For scenario (1), it resembles the case in SL. In scenario (2), the model fails to learn the correct information from this particular instance. Scenario (3) reveals that the prediction label may be the ground-truth label, which requires to be unlearned. Scenario (4) shows that the model does not learn correct information from this data. To ensure the model erases any potential information learned from unlearning data, the labeling information of NLL should cover all scenarios. Therefore, we employ a way to construct a set $\tilde{s}$ that guarantees the unlearning instance $\boldsymbol{x}$ associated with all potential information. If $\hat{y} = \tilde{y}$, $\tilde{s} = \{\tilde{y}\}$. On the other hand, if $\hat{y} \neq \tilde{y}$, we combine $\hat{y}$ and $\tilde{y}$ into $\tilde{s}$. The computation of $\tilde{s}$ is defined as:

$$\tilde{s} = \begin{cases} \{\tilde{y}\}, & \hat{y} = \tilde{y} \\ \{\tilde{y}, \hat{y}\}, & \hat{y} \neq \tilde{y} \end{cases}. \tag{10}$$

Table 1: Experimental results on 3 datasets over the multi-class classification in SL. Value shows accuracy (in %). The best performance is shown in boldface.

Dataset	Original	Approach	unlearning data ↓					remaining data ↑				
			20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
MNIST	98.53	Retrain	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	97.45	96.95	97.22	97.06	96.85
		Amnesiac ML	5.1	3.27	0.15	**0.00**	**0.00**	45.71	32.16	8.81	0.39	0.09
		K-priors	**0**	0.52	0.87	2.61	2.95	97.26	96.95	96.64	96.46	96.34
		UDRU	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**97.63**	**97.64**	**97.63**	**97.62**	**97.61**
Fashion	98.05	Retrain	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	94.8	94.7	94.7	94.8	94.7
		K-priors	0.42	1.07	1.72	6.48	9.43	94.74	94.61	94.12	95.15	96.35
		UDRU	0.54	1.04	2.41	3.29	3.64	**96.72**	**96.53**	**96.47**	**96.82**	**96.98**
CIFAR10	99.31	Retrain	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	91.19	91.12	91.10	91.24	91.15
		K-priors	0.01	3.14	6.09	8.98	11.94	89.99	90.19	89.98	90.14	90.07
		UDRU	0.02	0.04	0.03	0.01	0.01	**98.52**	**97.26**	**97.16**	**96.01**	**95.51**

Table 2: Experimental results on MNIST and Fashion-MNIST over the multi-class classification in PLL. Value shows accuracy (in %). The best performance is shown in boldface.

Dataset	Original	Approach	unlearning data↓					remaining data↑				
			20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
MNIST	98.68	Retrain	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	95.25	95.41	95.69	95.20	95.51
		Amnesiac ML	2.99	1.21	**0.00**	**0.00**	**0.00**	36.36	14.36	8.59	0.00	0.00
		K-priors	94.99	94.43	92.21	92.30	90.57	93.51	94.02	93.87	95.84	94.53
		UDRU	0.81	0.82	0.84	0.86	0.89	**96.21**	**96.57**	**96.19**	**96.77**	**96.49**
Fashion	93.20	Retrain	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	92.81	91.71	91.68	91.75	91.31
		K-priors	91.02	89.02	90.57	88.77	87.75	91.19	90.24	90.34	90.04	88.39
		UDRU	1.21	1.35	1.54	1.75	1.76	**93.65**	**93.24**	**92.91**	**92.35**	**93.15**

With Eq. (10), we can observe that the format of supervision information in NLL is the same as that in PLL. Therefore, the unlearning target $\boldsymbol{T}(\boldsymbol{z}, \tilde{s})$ of NLL can be defined as:

$$\boldsymbol{T}_i(\boldsymbol{z}, \tilde{s}) = \begin{cases} z_i, & i \notin \tilde{s} \\ \ln(\frac{1}{K - |\tilde{s}|} \sum_{j \notin \tilde{s}} e^{z_j}), & i \in \tilde{s}, \end{cases} \tag{11}$$

where $\boldsymbol{T}_i(\boldsymbol{z}, \tilde{s})$ refers to the i-th element of $\boldsymbol{T}(\boldsymbol{z}, \tilde{s})$. Finally, the objective function of NLL for machine unlearning is defined as:

$$\begin{aligned} \mathcal{J}_n(\boldsymbol{\theta}, \boldsymbol{\theta}^*, \mathcal{D}) = & \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \| \boldsymbol{f}_{NL}(\boldsymbol{x}_i, \boldsymbol{\theta}) - \boldsymbol{T}(\boldsymbol{f}_{NL}(\boldsymbol{x}_i, \boldsymbol{\theta}^*), \tilde{s}_i) \|^2 \\ & + \delta \| \boldsymbol{\theta} - \boldsymbol{\theta}^* \|^2 \end{aligned} \tag{12}$$

The unlearned parameters $\boldsymbol{\theta}_u$ can be obtained by minimizing the objective function:

$$\boldsymbol{\theta}_u = \arg\min_{\boldsymbol{\theta}} \mathcal{J}_n(\boldsymbol{\theta}, \boldsymbol{\theta}^*, \mathcal{D}_u). \tag{13}$$

Minimizing the objective function $\mathcal{J}_n$ enables successful unlearning of the data, regardless of the correctness of the label or whether the model has previously learned correct information from the training data. This allows us to effectively remove the influence of unlearning data, and ensure the model's predictions from being influenced by noisy information.

4 Experiments

In this section, we conduct experiments on three learning paradigms: SL, PLL, and NLL, to evaluate the performance of the proposed approach, UDRU. The experiments mainly focus on evaluating unlearning approaches in terms of efficiency in unlearning, the ability of preserving performance, and time consumption. We implement our experiments using PyTorch on NVIDIA RTX 4090. Our code is released at https://github.com/Ehwartz/udru.

4.1 Experimental Settings

Datasets & Pre-processing. We conduct experimental studies on three widely-used datasets: MNIST, Fashion-MNIST (Fashion), and CIFAR10. The MNIST dataset comprises handwritten digits distributed across 10 classes, while the Fashion dataset includes standardized images of fashion items with 10 classes. The CIFAR10 dataset consists of color images grouped into 10 classes. The partial rate of PLL is set

Table 3: Experimental results on 2 datasets over the multi-class classification in NLL. Value shows accuracy (in %). The best performance is shown in boldface.

Dataset	Approach	Original	unlearning data↓					remaining data↑				
			20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
MNIST	Retrain	97.33	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	96.92	96.11	**96.89**	**96.99**	95.20
	Amnesiac ML		2.94	1.15	0.43	0.28	0.07	36.08	14.18	8.49	3.20	1.72
	K-priors		94.90	93.95	90.61	92.12	89.95	92.33	93.10	93.63	94.18	93.46
	UDRU		0.58	0.56	0.59	0.54	0.56	**96.95**	**96.70**	96.34	95.72	**96.00**
Fashion	Retrain	91.37	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	90.25	90.86	90.75	90.20	90.90
	K-priors		93.76	92.52	92.44	87.91	87.47	87.73	87.30	87.33	87.51	86.53
	UDRU		0.92	1.26	1.38	1.38	1.43	**91.82**	**91.59**	**91.68**	**91.99**	**91.30**

Table 4: Execution time (s) comparison among the data unlearning approaches on SL, PLL, and NLL over 3 datasets. Lower is better, and the best performance is shown in boldface.

Approach	SL			PLL			NLL		
	MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10
Retrain	293.79	947.48	7537.11	301.66	980.72	7668.40	306.83	948.71	7739.66
K-priors	0.83	2.73	434.80	0.85	2.84	437.59	0.87	2.77	455.86
UDRU	**0.68**	**2.26**	**388.43**	**0.71**	**2.29**	**402.49**	**0.69**	**2.39**	**393.18**

as 0.2, and the noise rate of NLL is 0.2. MNIST and Fashion datasets are utilized for studying the unlearning performance of all approaches in PLL and NLL.

Baselines. We select retrain [Bourtoule *et al.*, 2021], Amnesiac ML [Graves *et al.*, 2021] and K-priors [Khan and Swaroop, 2021] as comparison approaches. Retrain, Amnesiac ML and K-priors belong to strategies representing the most basic unlearning strategy, updating parameters by storing gradients during training, and a knowledge adaption strategy, respectively. In the training process, we store gradients for each instance in datasets to apply Amnesiac ML by updating parameters with accumulated gradients. The implementation of K-priors depends on a loss function, which allows us to apply it for experiments by changing suitable loss functions according to different learning paradigms. It is worth noting that Amnesiac ML cannot handle the cases involving Fashion and CIFAR10 due to constraints in storing gradients and hidden layers of CNN and ResNet50 for all training data.

Setup. For machine unlearning tasks, we need to obtain parameters ($\boldsymbol{\theta}^*$) of the trained model before using the unlearning approaches to experiment. Therefore, we should train models for different learning paradigms firstly. The selection of appropriate classification models is based on the scales of the datasets and the difficulty of identification. Specifically, MLP, CNN, and ResNet50 are used to identify MNIST, Fashion and CIFAR10, respectively. For the selection of loss functions, Cross-Entropy loss is used to SL, and Classifier-Consistent Loss [Feng *et al.*, 2020] and Generalized Cross Entropy Loss [Zhang and Sabuncu, 2018] are used to train models in PLL and NLL, respectively. We train models using SGD with a learning rate of 10^{-4} and a weight decay of 10^{-3}. The batch size and epoch are set as 64 and 256, respectively. Subsequently, we introduce the settings for machine unlearning in three learning paradigms. Batch size for Fashion and CIFAR10 is set as 64, while we do not divide MNIST into batches. Learning rate for three datasets is set as 10^{-4}, δ in UDRU is set as 1 for MNIST and Fashion, and 10^{-2} for CIFAR10.

4.2 Empirical Results

As shown in Table 1-3, the unlearning data for each dataset shares the same label, where this label is randomly selected from the label space. In our systematic sampling, we consider 20%, 40%, 60%, 80%, and 100% of the data belonging to the selected label from the training dataset for unlearning. In the tables, ”Original” denotes the accuracy of the original training dataset. “Unlearning data” in the tables shows the accuracy of the models on the unlearning data after removing that data, while “remaining data” represents the accuracy of these unlearned models on the remaining data except the data in the same label as the unlearning data. ↓ indicates that the accuracy of approaches is smaller for the unlearning data, indicating better performance. ↑ displays that the accuracy of approaches is higher for the remaining data, the performance is better. The experiments for convergence and the selection of δ are shown in Appendix C.

Efficiency in Unlearning. Analyzing the accuracy reported in Table 1, Table 2, and Table 3, we find that the results of UDRU in “unlearning data” are consistently close to the comparison approaches across all datasets and different unlearning percentages. This demonstrates the effectiveness of UDRU in thoroughly unlearning sampled data, which ex-

Table 5: Comparison of membership inference attack after data removal operation. Value shows percentage of unlearning data that is identified as training data, where we display the results according to each label. Lower values show better performance of unlearning. The best performance is shown in boldface.

	Approach	Label										
		0	1	2	3	4	5	6	7	8	9	Average
SL	Original	76.01	71.48	64.25	76.36	79.51	65.49	71.52	76.37	76.41	65.91	72.33
	Retrain	52.77	53.93	**35.90**	**40.32**	32.18	51.65	50.06	41.20	26.21	38.54	42.28
	Amnesiac	44.83	45.24	50.85	49.49	54.15	50.04	30.95	50.85	70.14	60.90	50.75
	K-priors	62.14	42.26	64.30	69.19	67.76	42.33	62.25	44.97	68.45	50.58	57.42
	UDRU	**27.36**	**26.04**	61.21	55.77	**17.55**	**16.05**	**22.06**	**35.75**	**16.58**	**38.18**	**31.66**
PLL	Original	68.65	66.96	65.32	58.19	62.76	64.10	71.26	67.72	86.50	81.73	69.32
	Retrain	42.87	43.15	44.19	63.72	**50.60**	44.07	31.63	28.11	46.28	34.74	42.94
	Amnesiac	57.72	45.25	41.33	51.76	70.56	59.42	44.84	36.14	46.70	53.19	50.69
	K-priors	47.19	61.20	64.80	64.75	55.69	62.42	30.90	78.95	78.45	**23.46**	56.78
	UDRU	**35.84**	**26.15**	**19.64**	**29.62**	65.89	**37.70**	**17.34**	**25.36**	70.96	42.08	**37.06**
NLL	Original	70.52	76.94	89.95	82.21	69.18	61.90	49.61	77.32	80.03	85.01	74.27
	Retrain	**16.33**	23.62	52.31	57.50	**36.38**	49.55	**43.20**	48.05	**27.98**	57.34	41.22
	Amnesiac	47.98	70.83	57.65	37.27	65.94	46.62	70.38	55.42	43.80	60.37	55.63
	K-priors	69.46	77.87	28.36	67.25	46.29	76.64	57.36	38.74	53.57	77.09	59.26
	UDRU	51.66	**20.59**	**25.11**	**35.14**	46.51	**30.48**	58.64	**37.35**	44.42	**36.93**	**38.68**

hibits close performance to the retraining approach. Although UDRU may not completely erase all unlearning data in some cases, leading to a slight reduction in performance, these findings provide the potential of UDRU as a robust approach for efficient unlearning in the SL, PLL, and NLL scenarios.

The Ability of Preserving Performance. In Table 1, Table 2, and Table 3, we observe that the accuracy of "remaining data" in UDRU exhibits comparable performance across the most cases. This proves that UDRU can effectively preserve models' performance regardless of the number of instances being unlearned. Contrary to Retrain, which struggles to maintain the original performance in larger datasets because of unbalanced training datasets, and Amnesiac ML, which completely erodes models' performance, UDRU successfully maintains an impressive accuracy on the remaining data.

4.3 Additional Experiments

Time Consumption. To evaluate the time consumption of each unlearning approach, we conduct a comparative analysis of the experiment durations presented in Table 1-3 for unlearning 20% of the data. The results, as depicted in Table 4, indicate that UDRU outperforms other approaches, which demonstrates a faster unlearning process across various datasets. This observation highlights the computational efficiency of UDRU, positioning it as a compelling choice for practical applications due to its lower computational cost.

Privacy Protection Against Membership Inference Attack. A membership inference attack is a privacy attack that aims to determine whether a specific data was used during the process of training a machine learning model [Shokri *et al.*, 2017]. To evaluate the robustness of different unlearning approaches, including UDRU, against membership inference attacks, we conduct experiments using the MNIST dataset. Our experimental setup involves a white-box attack, where we concatenate all hidden layers and gradients in an MLP to train an attacker model. Then, we compare the accuracy when models are required to unlearn 20% data of each class separately. The results presented in Table 5 demonstrate that UDRU can lead to relatively lower precision of the attacker, which implies the effectiveness of UDRU on unlearning and thwarting membership inference attacks.

5 Conclusion

In this paper, we propose a novel approach called UDRU to explore machine unlearning across scenarios ranging from SL to WSL. Motivated by that untrained model's predictions follow a uniform distribution, we formulate an unlearning target for model outputs by minimizing the divergence between the model's prediction distribution and a uniform distribution. This erases the ability of the model to distinguish features from the training data. Recognizing that real-world applications often involve weak supervised information, UDRU can successfully address uncertainty in unlearning by analyzing possible situations in PLL and NLL. Furthermore, formulating targets only from outputs of unlearned data that gets rid of redundant training data, enhancing the flexibility of our approach, particularly when unlearning large models and big datasets. With formulated targets, we apply a regularization to derive an objective function, which converts unlearning task of classification models into convex regression. This contributes to faster convergence and reduced computational cost. Empirical studies show the superiority and robustness of our approach in unlearning across SL, PLL, and NLL tasks while preserving model performance. Additionally, UDRU proves effective against membership inference attacks.

References

[Arpit *et al.*, 2017] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 233–242, Sydney, Australia, 2017.

[Baumhauer *et al.*, 2022] Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: linear filtration for logit-based classifiers. *Mach. Learn.*, 111(9):3203–3226, 2022.

[Bourtoule *et al.*, 2021] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *Proceedings of 42nd IEEE Symposium on Security and Privacy*, pages 141–159, San Francisco, CA, 2021.

[Brophy and Lowd, 2021] Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 1092–1104, Virtual Event, 2021.

[Chundawat *et al.*, 2023] Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan S. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of 37th AAAI Conference on Artificial Intelligence*, pages 7210–7217, Washington, DC, 2023.

[Cour *et al.*, 2011] Timothée Cour, Benjamin Sapp, and Ben Taskar. Learning from partial labels. *J. Mach. Learn. Res.*, 12:1501–1536, 2011.

[Feng and An, 2019a] Lei Feng and Bo An. Partial label learning by semantic difference maximization. In Sarit Kraus, editor, *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2294–2300, Macao, China, 2019.

[Feng and An, 2019b] Lei Feng and Bo An. Partial label learning with self-guided retraining. In *Proceedings of 33rd AAAI Conference on Artificial Intelligence*, pages 3542–3549, Honolulu, HA, 2019.

[Feng *et al.*, 2020] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems 33*, Virtual Event, 2020.

[Gao and Zhang, 2021] Yi Gao and Min-Ling Zhang. Discriminative complementary-label learning with weighted loss. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139, pages 3587–3597, Virtual Event, 2021.

[Gao *et al.*, 2023] Yi Gao, Miao Xu, and Min-Ling Zhang. Unbiased risk estimator to multi-labeled complementary label learning. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 3732–3740, Macao, China, 2023.

[Ginart *et al.*, 2019] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems 32*, pages 3513–3526, Vancouver, BC, 2019.

[Golatkar *et al.*, 2020] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9301–9309, Seattle, WA, 2020.

[Graves *et al.*, 2021] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of 35th AAAI Conference on Artificial Intelligence*, pages 11516–11524, Virtual Event, 2021.

[Guo *et al.*, 2020] Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119, pages 3832–3842, Virtual Event, 2020.

[Huang *et al.*, 2021] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *Proceedings of 9th International Conference on Learning Representations*, Virtual Event, 2021.

[Izzo *et al.*, 2021] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *Proceedings of 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 2008–2016, Virtual Event, 2021.

[Khan and Swaroop, 2021] Mohammad Emtiyaz Khan and Siddharth Swaroop. Knowledge-adaptation priors. In *Advances in Neural Information Processing Systems 34*, pages 19757–19770, Virtual Event, 2021.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, Lake Tahoe, NV, 2012.

[Mantelero, 2013] Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[Nguyen *et al.*, 2022] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *CoRR*, abs/2209.02299, 2022.

[Noh *et al.*, 2015] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of 2015 IEEE In-*

ternational Conference on Computer Vision, pages 1520–1528, Santiago, Chile, 2015.

[Patrini *et al.*, 2017] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2233–2241, Honolulu, HI, 2017.

[Proserpio *et al.*, 2014] Davide Proserpio, Sharon Goldberg, and Frank McSherry. Calibrating data to sensitivity in private data analysis. *Proc. VLDB Endow.*, 7(8):637–648, 2014.

[Sekhari *et al.*, 2021a] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems 34*, pages 18075–18086, Virtual Event, 2021.

[Sekhari *et al.*, 2021b] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems 34*, pages 18075–18086, Virtual Event, 2021.

[Shen *et al.*, 2024a] Shaofei Shen, Chenhao Zhang, Alina Bialkowski, Weitong Chen, and Miao Xu. Camu: Disentangling causal effects in deep model unlearning. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, Virtual Event, 2024.

[Shen *et al.*, 2024b] Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Tony Chen, and Miao Xu. Label-agnostic forgetting: A supervision-free unlearning in deep models. In *The Twelfth International Conference on Learning Representations*, Virtual Event, 2024.

[Shi *et al.*, 2023] Yu Shi, Ning Xu, Hua Yuan, and Xin Geng. Unreliable partial label learning with recursive separation. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 4208–4216, Macao, China, 2023.

[Shintre *et al.*, 2019] Saurabh Shintre, Kevin A. Roundy, and Jasjeet Dhaliwal. Making machine learning forget. In *Annual Privacy Forum*, volume 11498, pages 72–83, Rome, Italy, 2019.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of 2017 IEEE Symposium on Security and Privacy*, pages 3–18, San Jose, CA, 2017.

[State of California Department of Justice, 2023] State of California Department of Justice. California consumer privacy act. https://oag.ca.gov/privacy/ccpa, 2023.

[Tang *et al.*, 2024] Wei Tang, Weijia Zhang, and Min-Ling Zhang. Multi-instance partial-label learning: Towards exploiting dual inexact supervision. *Science China Information Sciences*, 67(3):1–14, 2024.

[Thudi *et al.*, 2022] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: understanding factors influencing machine unlearning. In *Proceedings of 7th IEEE European Symposium on Security and Privacy*, pages 303–319, Genoa, Italy, 2022.

[Wei *et al.*, 2022] Yi Wei, Mei Xue, Xin Liu, and Pengxiang Xu. Data fusing and joint training for learning with noisy labels. *Frontiers of Computer Science*, 16(6):166338, 2022.

[Wen *et al.*, 2021] Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 11091–11100, Virtual Event, 2021.

[Wu *et al.*, 2022] Ga Wu, Masoud Hashemi, and Christopher Srinivasa. PUMA: performance unchanged model augmentation for training data removal. In *Proceedings of 36th AAAI Conference on Artificial Intelligence*, pages 8675–8682, Virtual Event, 2022.

[Xu *et al.*, 2024] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1):9:1–9:36, 2024.

[Yan *et al.*, 2022] Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. ARCANE: an efficient architecture for exact machine unlearning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pages 4006–4013, Vienna, Austria, 2022.

[Zhang and Sabuncu, 2018] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31*, pages 8792–8802, Montréal, Canada, 2018.

[Zhang *et al.*, 2017] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of 5th International Conference on Learning Representations*, Toulon, France, 2017.

[Zhou and Gu, 2018] Yu Zhou and Hong Gu. Geometric mean metric learning for partial label data. *Neurocomputing*, 275:394–402, 2018.

[Zhou, 2018] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

A The Proof of Theorem 2

Theorem 2. *To satisfy the conditions shown in Eq.* (5), $\boldsymbol{T}(\boldsymbol{z}, y)$ *is derived as:*

$$\boldsymbol{T}_i(\boldsymbol{z}, y) = \begin{cases} z_i, & i \neq y \\ \ln(\frac{1}{K-1}\sum_{j\neq y} e^{z_j}), & i = y \end{cases}$$

where $T_i(\boldsymbol{z}, y)$ *presents the* i*-th element of* $T(\boldsymbol{z}, y)$*.*

Proof. According to Eq. (5), we start with KL-Divergence:

$$\begin{aligned} D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p}) &= \sum_{i=1}^{K} \frac{1}{K} \ln \frac{\frac{1}{K}}{\sigma_i(\boldsymbol{t})} \\ &= \sum_{i=1}^{K} \frac{1}{K} \ln \frac{\frac{1}{K}}{\frac{\boldsymbol{e^{t_i}}}{\sum_{j=1}^{K} \boldsymbol{e^{t_j}}}}. \end{aligned} \tag{14}$$

To simplify the formula, let $a_i = e^{t_i}$, so we can obtain

$$\begin{aligned} D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p}) &= \sum_{i=1}^{K} \frac{1}{K} \ln \frac{\frac{1}{K}}{\sigma_i(\boldsymbol{t})} \\ &= \sum_{i=1}^{K} \frac{1}{K} \ln \frac{\frac{1}{K}}{\frac{\boldsymbol{a_i}}{\sum_{j=1}^{K} \boldsymbol{a_j}}} \\ &= -\ln K - \sum_{i=1}^{K} \frac{1}{K} \ln \frac{a_i}{\sum_{j=1}^{K} a_j} \\ &= -\ln K - \sum_{i\neq y} \frac{1}{K} \ln \frac{a_i}{\sum_{j=1}^{K} a_j} - \frac{1}{K} \ln \frac{a_y}{\sum_{j=1}^{K} a_j}. \end{aligned} \tag{15}$$

Proceed with the derivative of D_{KL} with respect to a_y:

$$\begin{aligned} \frac{\partial}{\partial a_y} D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p}) &= \frac{1}{K} \frac{K-1}{\sum_{j=1}^{K} a_j} - \frac{1}{K} \frac{1}{a_y} \frac{\sum_{j=1}^{K} a_j - a_y}{\sum_{j=1}^{K} a_j} \\ &= \frac{1}{K} \frac{K-1}{\sum_{j=1}^{K} a_j} - \frac{1}{K} \frac{1}{a_y} \frac{\sum_{j\neq y} a_j}{\sum_{j=1}^{K} a_j}. \end{aligned} \tag{16}$$

Now, let $\frac{\partial}{\partial a_y} D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p}) = 0$, we can have $a_y^* = \frac{1}{K-1} \sum_{j\neq y} a_j$. When $a_y < a_y^*$, $\frac{\partial}{\partial a_y} D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p}) < 0$. When $a_y > a_y^*$, $\frac{\partial}{\partial a_y} D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p}) > 0$. So, when $a_y = a_y^*$, $D_{KL}(\bar{\boldsymbol{p}}\|\boldsymbol{p})$ reaches the minimal divergence. Additionally, it satisfies all requirements in Eq. (5). Then we can get:

$$\boldsymbol{T}_y(\boldsymbol{z}, y) = \ln(\frac{1}{K-1}\sum_{j\neq y} e^{z_j}) \tag{17}$$

□

B The Proof of Theorem 3

Theorem 3. *Let $\boldsymbol{z} = \boldsymbol{f}_{PL}(\boldsymbol{x}, \boldsymbol{\theta})$ and z_i denotes the i-th element of $\boldsymbol{z}$. The i-th element of the unlearning target $\boldsymbol{T}(\boldsymbol{z}, s)$ in PLL is expressed as:*

$$\boldsymbol{T}_i(\boldsymbol{z}, s) = \begin{cases} z_i, & i \notin s \\ \ln(\frac{1}{K - |s|} \sum\limits_{j \notin s} e^{z_j}), & i \in s \end{cases}$$

Proof. According to Eq. (15):

$$D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{p}) = -\ln K - \sum_{i \notin s} \frac{1}{K} \ln \frac{a_i}{\sum\limits_{j=1}^{K} a_j} - \sum_{i \in s} \frac{1}{K} \ln \frac{a_i}{\sum\limits_{j=1}^{K} a_j}. \tag{18}$$

Calculate partial derivatives of D_{KL} with respect to a_i, for $i \in s$:

$$\begin{aligned} \frac{\partial}{\partial a_i} D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{p}) &= \frac{1}{K} \frac{K-1}{\sum\limits_{j=1}^{K} a_j} - \frac{1}{K} \frac{1}{a_i} \frac{\sum\limits_{j=1}^{K} a_j - a_i}{\sum\limits_{j=1}^{K} a_j} \\ &= \frac{1}{K} \frac{K-1}{\sum\limits_{j=1}^{K} a_j} - \frac{1}{K} \frac{1}{a_i} \frac{\sum\limits_{j \neq y} a_j}{\sum\limits_{j=1}^{K} a_j}. \end{aligned} \tag{19}$$

If we have confidence that the model has learned comprehensively from the dataset and can consistently make correct predictions on the training set, we can directly use Theorem 2. However, in most situations, it is difficult to determine whether the prediction is correct. To be on the safe side, we should consider all labels in the candidate set s. To achieve this, we define:

$$a_i = a_p, \ i \in s. \tag{20}$$

Then with Eq. (19) and Eq. (20), we can derive

$$\frac{1}{K} \frac{K-1}{\sum\limits_{j=1}^{K} a_j} - \frac{1}{K} \frac{1}{a_i} \frac{\sum\limits_{j \neq y} a_j}{\sum\limits_{j=1}^{K} a_j} = \frac{1}{K} \frac{K-1}{\sum\limits_{j=1}^{K} a_j} - \frac{1}{K} \frac{1}{a_p} \frac{(|s|-1)a_p + \sum\limits_{j \notin s} a_j}{\sum\limits_{j=1}^{K} a_j} = 0, \tag{21}$$

$$a_p^* = \frac{1}{K - |s|} \sum_{j \notin s} a_j. \tag{22}$$

When $a_p < a_p^*$, $\frac{\partial}{\partial a_y} D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{p}) < 0$. When $a_p > a_p^*$, $\frac{\partial}{\partial a_y} D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{p}) > 0$. So, when $a_p = a_p^*$, $D_{KL}(\bar{\boldsymbol{p}} \| \boldsymbol{p})$ reaches the minimal divergence and requirements. Therefore, we can formulate the target as:

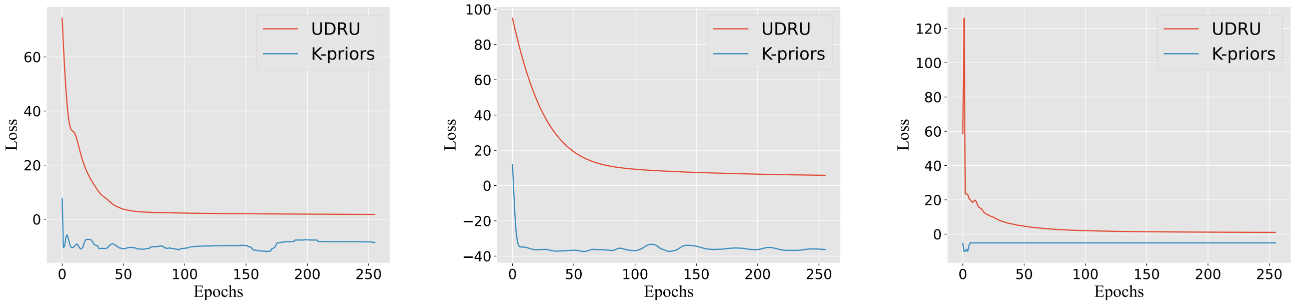
$$\boldsymbol{T}_i(\boldsymbol{z}, s) = \ln(\frac{1}{K - |s|} \sum_{j \notin s} e^{z_j}), \ i \in s. \tag{23}$$

□

C Experiments

The Experiments of Convergence. We conducted experiments to compare the convergence progress of UDRU with K-priors in SL. Both approaches only rely on loss calculated from unlearning data, while the loss of retraining models is based on all remaining data. Amnesiac ML, on the other hand, does not produce loss. Figure 1 illustrates the results of loss convergence on three datasets. The findings indicate that UDRU exhibits smoother convergence, which is faster than K-priors to reach a stable value. In CIFAR10, the starting increase in UDRU's loss curve is attributed to the larger parameter scale of ResNet50, contributing to a sharp rise in regularization at the beginning.

Effect of Different δ. To assess the parameter sensitivity of δ, we assign different orders of magnitude to δ, ranging from 1 to 10^{-4}, to evaluate its effect in SL. Observing the results in Table 6, it is evident that smaller datasets are more sensitive to changes in δ, with a significant drop in accuracy on remaining data when δ is set as 10^{-2} on MNIST. Meanwhile, in Fashion and CIFAR10, accuracy on unlearning data and remaining data needs to be balanced to consider both unlearning and preserving performance. Therefore, we set δ as 1 for MNIST and as 10^{-2} for Fashion and CIFAR10 to achieve more comprehensive performance.



(a) Unlearning loss on MNIST of 256 epochs (b) Unlearning loss on Fashion of 256 epochs (c) Unlearning loss on CIFAR10 of 256 epochs

Figure 1: Comparison of loss convergence between UDRU and K-priors on 3 datasets when models are required to unlearn 20% data of one class.

Table 6: Experimental results of different δ ranging from 1 to 10^{-4} on 3 datasets in the SL paradigm.

Dataset	Original	δ				
		1	10^{-1}	10^{-2}	10^{-3}	10^{-4}
		unlearning data↓				
MNIST	98.46	0.00	0.00	0.00	0.00	0.00
Fashion	97.98	0.87	0.84	0.54	0.33	0.21
CIFAR10	99.61	1.90	0.09	0.02	0.02	0.00
		remaining data↑				
MNIST	98.44	97.63	97.09	89.97	89.29	89.38
Fashion	98.01	97.78	97.47	96.72	96.59	96.33
CIFAR10	99.55	99.21	99.14	98.52	96.38	94.53