Contrastive Unlearning: A Contrastive Approach to Machine Unlearning

Hong kyu Lee , **Qiuchen Zhang** , **Carl Yang** , **Jian Lou** and **Li Xiong**
Emory University
{hong.kyu.lee, qiuchen.zhang, j.carlyang, jian.lou, lxiong}@emory.edu

Abstract

Machine unlearning aims to eliminate the influence of a subset of training samples (i.e., unlearning samples) from a trained model. Effectively and efficiently removing the unlearning samples without negatively impacting the overall model performance is challenging. Existing works mainly exploit input and output space and classification loss, which can result in ineffective unlearning or performance loss. In addition, they utilize unlearning or remaining samples ineffectively, sacrificing either unlearning efficacy or efficiency. Our main insight is that the direct optimization on the representation space utilizing both unlearning and remaining samples can effectively remove influence of unlearning samples while maintaining representations learned from remaining samples. We propose a contrastive unlearning framework, leveraging the concept of representation learning for more effective unlearning. It removes the influence of unlearning samples by contrasting their embeddings against the remaining samples' embeddings so that their embeddings are closer to the embeddings of unseen samples. Experiments on a variety of datasets and models on both class unlearning and sample unlearning showed that contrastive unlearning achieves the best unlearning effects and efficiency with the lowest performance loss compared with the state-of-the-art algorithms. In addition, it is generalizable to different contrastive frameworks and other models such as vision-language models. Our main code is available on github.com/Emory-AIMS/Contrastive-Unlearning

1 Introduction

Machine unlearning [Cao and Yang, 2015] aims to remove a subset of data (i.e., unlearning samples) from a trained machine learning (ML) model without retraining the model from scratch and has received increasing attention due to various privacy regulations. Notably, "the right to be forgotten" from the General Data Protection Requirement (GDPR) gives individuals the right to request their data to be removed from databases, which extends to models trained on such data [Mantelero, 2024]. Since models can remember training data within their parameters [Arpit *et al.*, 2017], it is necessary to "unlearn" these data from a trained model. The goals and evaluation metrics for unlearning typically include: 1) unlearning efficacy, which measures how well the algorithm removes the influence of unlearning samples. This can be assessed by the model's performance on the unlearning samples, or by its robustness against membership inference attacks [Shokri *et al.*, 2017; Carlini *et al.*, 2022; Ye *et al.*, 2022; Sablayrolles *et al.*, 2019]; 2) model performance on its original tasks, which ensures that the unlearning does not significantly degrade its overall accuracy; and 3) computational efficiency, which assesses the time and resources required for the unlearning.

While many promising approaches are proposed, existing works present several limitations: 1) They exploit input and output space and classification loss. As a result, it may lead to significant shift in decision boundaries, affecting model utility. 2) They either utilizes unlearning or remaining samples alone or use both but in an ineffectively and hence sacrifice either unlearning efficacy or efficiency. For example, Gradient Ascent [Golatkar *et al.*, 2020] only uses unlearning samples and attempts to reverse their impact by applying gradient *ascent* using the classification loss. Finetune [Golatkar *et al.*, 2020] only uses remaining samples to iteratively retrain the model to gradually remove the influence of unlearning samples leveraging the catastrophic forgetting effect [Goodfellow *et al.*, 2013]. SCRUB [Kurmanji *et al.*, 2023] uses both unlearning and remaining samples for unlearning, but requires multiple iterations over the entire remaining samples, leading to excessive computations.

Our Contributions. To address these deficiencies, we present a novel contrastive approach for machine unlearning, or **contrastive unlearning**. We rethink the problem of machine unlearning in the perspective of representation space. We re-purpose the idea of supervised contrastive learning [Khosla *et al.*, 2020], a widely used representation learning approach, for more effective unlearning of general classification models.

The goal for unlearning is rooted in the fundamental difference between how a model perceives training and test samples. The model optimizes the representations of the training samples during the learning process, resulting in embeddings that are deeply aligned within the correct decision boundaries, often with high confidence. Test samples, in contrast, are unseen during training and typically produce embeddings within the correct decision boundary but closer to the boundary, reflecting the model's generalization to new data. This distinction is also the basis for privacy vulnerabilities, such as membership inference attacks [Shokri *et al.*, 2017; Yeom *et al.*, 2018], where adversaries exploit the model's higher confidence or distinctive embeddings for training samples to infer their membership in the training dataset.

Based on the rationale, our main idea is to simultaneously contrast an unlearning sample with 1) Positive samples (remaining samples from the same class as the unlearning sample) and push their embeddings apart from each other, and 2) Negative samples (remaining samples from different classes as the unlearning sample) and pull their embeddings close to each other. This results in embeddings of unlearning samples away from remaining training samples and closer to the decision boundaries and test samples' embeddings. It has two main insights. First, directly optimizing the embeddings of unlearning samples facilitates more effective unlearning. Second, by contrasting embeddings of unlearning samples, it can effectively unlearn while minimizing any change of the decision boundaries. Additionally we introduce an auxiliary classification loss on the contrasted remaining samples to further maintain model accuracy.

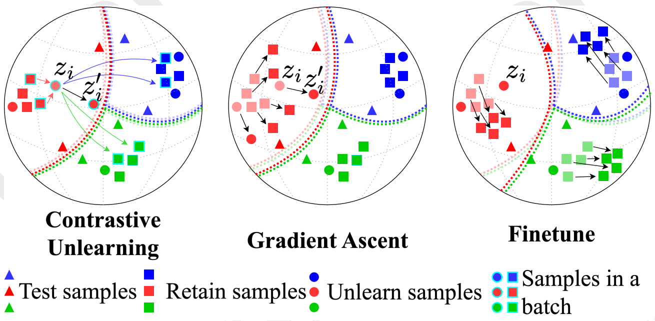

Figure 1: Visualization of Representation Spaces for Contrastive Unlearning, Gradient Ascent, and Finetuning

Figure 1 illustrate the intuition of contrastive unlearning compared to existing approaches in a normalized representation space. Circles, squares, and triangles are embeddings of unlearning, remaining samples, and test samples, respectively. Colors represent different classes. Dotted lines show decision boundaries. We assume the model has been trained, so the embeddings of training samples are clustered to their respective classes [Das and Chaudhuri, 2024].

Given an embedding of sample z_i, contrastive unlearning pushes z_i away from its own class (positive pairs) and pulls z_i towards the samples with different classes (negative pairs). This results in the unlearned embedding z_i' to be distant from remaining samples and closer to the decision boundaries, where test samples' embeddings (triangles) are located. In comparison, Gradient ascent [Golatkar *et al.*, 2020] pushes z_i away in the representation space from its own class but may either apply insufficient change (ineffective unlearning), or significantly affect embeddings of remaining samples of the same class and the decision boundary (model utility loss). Finetune indirectly pushes the unlearning samples away from its class (ineffective unlearning) and is susceptible to overfitting to the remaining samples (model utility loss).

Our contrastive *unlearning* is fundamentally different from contrastive learning. The goal of contrastive learning is to learn representations to distinguish different samples, while our goal is to modify embeddings of unlearning samples and maintain model's general classification performance. It features several novel algorithm designs and new findings: 1) we construct contrasting pairs different from conventional contrastive learning to serve the unlearning purpose and design new contrastive unlearning losses for both sample unlearning (unlearning randomly selected training samples) and single class unlearning (unlearning every sample of a class); 2) while it is common to add a classification loss on the remaining samples to maintain the performance of the unlearning model, we find that the classification loss helps keep the embeddings of the remaining samples in place and reciprocally improves unlearning effectiveness, validated by our empirical analysis followed by in-depth analysis.

In addition, contrastive unlearning is highly scalable as it can leverage other existing contrastive learning algorithms as a backbone. While our main experiments and analysis utilize supervised contrastive learning (SupCon) [Khosla *et al.*, 2020], we also demonstrate the scalability using MoCo [He *et al.*, 2020]-based contrastive framework. Finally, while existing approaches focus on standard classifiers, contrastive unlearning is highly generalizable, capable of unlearning a variety of models. We empirically demonstrate its effectiveness in unlearning a class from a finetuned vision-language model CLIP [Radford *et al.*, 2021].

In summary, our contributions are as follows:

(1) We propose contrastive unlearning, a novel unlearning framework utilizing the concept of the representation learning and contrastive loss. Instead of analyzing inputs and outputs of the model, we formulate the unlearning framework as modifying embeddings of unlearning samples to be similar to the embeddings of test samples (unseen samples) without directly using them, hence effectively removing the influence of unlearning samples.

(2) To achieve the unlearning goal, we customize the contrastive unlearning loss for two different unlearning tasks: single class unlearning and random sample unlearning. We design an effective termination condition for each task, achieving effective and efficient unlearning.

(3) We conduct comprehensive experiments comparing contrastive unlearning with various state-of-the-art methods on two unlearning tasks, single class and sample unlearning, to demonstrate the effectiveness, efficiency, and versatility of our approach. We also conduct a membership inference attack to verify the unlearning efficacy of sample unlearning. The results show that contrastive unlearning has the best efficacy while maintaining model utility with high computational efficiency. In addition, we demonstrate generalizability of contrastive unlearning across various models by unlearning an vision-language model. We show scalability of our approach by leveraging more advanced contrastive learning algorithms.

2 Related Works

Machine unlearning was introduced by [Cao and Yang, 2015] with three goals: 1) completeness, suggesting an unlearning algorithm should reverse the influence of unlearning samples and the unlearned model should be consistent with a retrained model with the remaining samples; and 2) timeliness, the unlearning algorithm should be be faster than retraining; and 3) The unlearned model should maintain high utility after unlearning. Exact unlearning ensures the completeness of unlearning. SISA is an exact unlearning framework that splits the dataset into partitions and retrains submodels whose shard has the unlearning sample [Bourtoule *et al.*, 2021]. These require partitioned training and still costly retraining, and model performance is highly dependent on partitioning strategy [Koch and Soll, 2023].

Approximate unlearning allows approximate completeness. Certified unlearning provides a mathematical guarantee on unlearning. [Guo *et al.*, 2020] proposed unlearning using newton-type hessian update with (ε, δ)-indistinguishability. [Neel *et al.*, 2024] utilized projected gradient descent on the partitioned dataset with a probabilistic bound. [Gupta *et al.*, 2021] proposed adaptive unlearning streams. Fisher unlearning uses fisher information matrix [Golatkar *et al.*, 2020] to identify optimal noise to remove the unlearning samples. Drawbacks of certified unlearning algorithms include the difficulty to scale, and most of them require convexity for the mathematical guarantee. Moreover, [Thudi *et al.*, 2022] questioned validity of certified unlearning. Recently, some works tried to address limitations of certified unlearning, including LCODEC [Mehta *et al.*, 2022], which reduced the computation cost by selectively generating hessian matrices and certified unlearning for non-convex setting [Zhang *et al.*, 2024]. While both are promising, experimental results show suboptimal unlearn efficacy.

Another body of approximate unlearning shows the unlearning effect through empirical evaluations. Usually, these works target class unlearning, which is to unlearn every sample of a class. UNSIR [Tarun *et al.*, 2023] conducts noisy gradient updates. Boundary unlearning unlearns an entire class [Chen *et al.*, 2023] by changing decision boundaries. ERM-KTP uses a special model architecture known as an entanglement reduce mask [Lin *et al.*, 2023]. SCRUB [Kurmanji *et al.*, 2023] is based on the knowledge distillation, where the teacher or the original model transfers knowledge to the unlearned model in every class except the unlearning class. [Bui *et al.*, 2024] proposed more robust second-order unlearning. The authors proposed cubic-regularizer to prevent hessian degeneration. [Nguyen *et al.*, 2022] proposed markov-chain monte carlo algorithm for unlearning, and [Nguyen *et al.*, 2020] proposed unlearning for bayesian models. Recently, [Cha *et al.*, 2024] proposed instance-wise unlearning with analysis on decision boundaries. However, it assumes that remaining samples are unavailable, and defined the unlearning goal as to incorrectly classify all unlearning samples, which are different from most unlearning works. As most works, we assume that remaining samples are available and our goal of unlearning is to make the model to perceive unlearning samples as unseen samples, not to completely misclassify them. We do not compare with [Cha *et al.*, 2024] due to its different assumption and unlearning goal.

3 Problem Definition

We define a classification model $\mathcal{F} = \mathcal{H}\left(\mathcal{E}_\theta\left(\cdot\right)\right)$ where $\mathcal{E}_\theta\left(\cdot\right)$ is a neural network based encoder parameterized by θ and $\mathcal{H}\left(\cdot\right)$ is a classification head. $\mathcal{E}_\theta$ produces embeddings z given a sample x. $\mathcal{H}$ receives z and yields a prediction. Let $\mathcal{F}$ be trained using dataset $\mathcal{D}_{tr} = \{(x_1, y_1) \cdots (x_n, y_n)\}$, where each data point is a tuple (x_i, y_i) including feature set x_i and label $y_i \in \{0 \cdots C\}$ where C is the number of classes. We suppose $\mathcal{F}$ was trained with cross-entropy loss. Let $\mathcal{D}_{ts}$ be a test dataset sampled from an analogous distribution with $\mathcal{D}_{tr}$, satisfying $\mathcal{D}_{ts} \cap \mathcal{D}_{tr} = \emptyset$.

Let $\mathcal{D}_{tr}^u \subseteq \mathcal{D}_{tr}$ be a set of samples to be forgotten (i.e., unlearning samples). The remaining set is $\mathcal{D}_{tr}^r = \mathcal{D}_{tr} \setminus \mathcal{D}_{tr}^u$. Let a retrained model $\mathcal{F}^R$ be trained only with $\mathcal{D}_{tr}^r$. An unlearning algorithm M receives $\mathcal{D}_{tr}^r, \mathcal{D}_{tr}^u, \theta$ and produces θ'. An unlearned model $\mathcal{F}' = \mathcal{H}\left(\mathcal{E}_{\theta'}\right)$ should resemble $\mathcal{F}^R$.

3.1 Single Class Unlearning

For single class unlearning, D_{tr}^u consists of all samples of an unlearning class c. The test set $\mathcal{D}_{ts}$ can be split into $\mathcal{D}_{ts}^u$ and $\mathcal{D}_{ts}^r$, where $\mathcal{D}_{ts}^u$ includes all test samples of class c, and $\mathcal{D}_{ts}^r = \mathcal{D}_{ts} \setminus \mathcal{D}_{ts}^u$ includes all test samples of remaining classes. A retrained model $\mathcal{F}^R$ will have zero accuracy on $\mathcal{D}_{tr}^u$ and $\mathcal{D}_{ts}^u$, the training and test samples of class c, since it was retrained without class c. So given an accuracy function Acc, the goal of single class unlearning is for the unlearned model $\mathcal{F}'$ to achieve near-zero accuracy on both training and test samples of class c (unlearning efficacy) and similar accuracy as the retrained model $\mathcal{F}^R$ for remaining classes (model utility).

$$\mathrm{Acc}\left(\mathcal{F}', \mathcal{D}_{tr}^u\right) \approx 0, \quad \mathrm{Acc}\left(\mathcal{F}', \mathcal{D}_{ts}^u\right) \approx 0, \tag{1}$$

$$\mathrm{Acc}\left(\mathcal{F}', \mathcal{D}_{ts}^r\right) \approx Acc\left(\mathcal{F}^R, \mathcal{D}_{ts}^r\right). \tag{2}$$

Single-class unlearning can be potentially implemented using simple rules such as assigning random labels from remaining classes to the samples classified as the unlearning class. However, such rule-based unlearning has fundamental flaws: (1) Insufficient Unlearning: the patterns or influence of samples from the unlearning class remain within the model (weights). If the model is released or leaked, an adversary can potentially recover knowledge of the unlearning class. (2) Model Utility: the random class assignment can degrade the performance of all remaining classes. Hence our goal is to unlearn the model itself to remove the influence of the class.

3.2 Sample Unlearning

For sample unlearning, the unlearning samples $\mathcal{D}_{tr}^u$ can belong to different classes. A retrained model $\mathcal{F}^R$ has similar accuracy on unlearning samples $\mathcal{D}_{tr}^u$ and test samples $\mathcal{D}_{ts}$. So the goal of sample unlearning is for the unlearned model $\mathcal{F}'$ to achieve similar accuracy as the retrained model $\mathcal{F}^R$ on both unlearning samples (unlearning efficacy) and test samples (model utility).

$$\text{Acc}\left(\mathcal{F}', \mathcal{D}_{tr}^{u}\right) \approx Acc\left(\mathcal{F}^{R}, \mathcal{D}_{ts}\right), \tag{3}$$

$$\text{Acc}\left(\mathcal{F}', \mathcal{D}_{ts}\right) \approx Acc\left(\mathcal{F}^{R}, \mathcal{D}_{ts}\right). \tag{4}$$

As we discussed earlier, a model's generalization capability is intrinsically related to unlearning. A model with stronger generalization can be easier for sample unlearning because it relies on broader patterns rather than memorizing individual data points, and its test and train accuracy is already similar. (Equation 3). However, generalization alone is insufficient and even a generalized model can still memorize unique pattern of training samples and requires full unlearning [Long *et al.*, 2018].

4 Contrastive Unlearning

Contrastive unlearning utilizes representation space for unlearning purposes and leverages the contrast between remaining and unlearning samples. If a sample x had been used as a training example, information extracted from x by $\mathcal{E}_\theta$ would be geometrically expressed in the representation space. Specifically, we hypothesize that samples of a class have similar embeddings and samples from different classes have dissimilar embeddings even when the model was not explicitly trained with representation learning. Existing literature supports this by mathematically and empirically showing that a model optimized with cross-entropy loss produces higher geometric similarity among embeddings of samples of the same class and lower similarity among different classes [Das and Chaudhuri, 2024; Graf *et al.*, 2021].

From this intuition, we aim to isolate the representations or embeddings of unlearning samples away from remaining samples up to the point where the model perceives them as unseen samples. To effectively achieve this, we contrast each unlearning sample with 1) remaining samples from the same class (positive pairs) and push their representations apart from each other, and 2) remaining samples from different classes (negative pairs) and pull their representations closer to each other. To this end, the embeddings of unlearning samples approach to the decision boundaries of the classes. This has some relation with existing literature of contrastive learning, however, our approach is fundamentally different as it contrasts pairs of unlearning and remaining samples while contrastive learning contrasts samples simply by their classes.

Contrastive Unlearning Loss: Sample Unlearning. Contrastive unlearning uses a batched process. In each round, an unlearning batch $X^u = \{x_1^u, \cdots x_B^u\}$ with size B is sampled from the unlearning data $\mathcal{D}_{tr}^u$, and a remaining batch $X^r = \{x_1^r \cdots x_B^r\}$ is sampled from the remaining set $\mathcal{D}_{tr}^r$. We denote x_i, the i-th sample of X^u, as an anchor. Based on x_i, positives and negatives are chosen from X^r. Positives are $P_{\mathbf{x}}(x_i) = \{x_j | x_j \in X^r, y_j = y_i\}$, or remaining samples with the same class as x_i; negatives are $N_{\mathbf{x}}(x_i) = \{x_j | x_j \in X^r, y_j \neq y_i\}$, or remaining samples with different class as x_i. Correspondingly, let embeddings of positives and negatives be $P_{\mathbf{z}}(x_i) = \{z_j | z_j = \mathcal{E}_\theta(x_j), x_j \in P_{\mathbf{x}}(x_i)\}$ and $N_{\mathbf{z}}(x_i) = \{z_j | z_j = \mathcal{E}_\theta(x_j), x_j \in N_{\mathbf{x}}(x_i)\}$. The contrastive unlearning loss aims to minimize the similarity of positive pairs and maximizes the similarity of negative pairs (the opposite of contrastive learning).

$$\mathcal{L}_{UL} = \sum_{x_i \in X^u} \frac{-1}{|N_{\mathbf{z}}(x_i)|} \sum_{z_a \in N_z} \log \frac{\exp(z_i \cdot z_a / \tau)}{\sum\limits_{z_p \in P_{\mathbf{z}}(x_i)} \exp(z_i \cdot z_p / \tau)}. \tag{5}$$

where $\tau \in \mathcal{R}^+$ is a scalar temperature parameter. In our final algorithm, we contrast each X^u, with ω randomly sampled batches of X^r. Thus within a single unlearning round, our algorithm computes every batch of $\mathcal{D}_{tr}^u$ for ω times. Refer to appendix B for more details.

Contrastive Unlearning Loss: Single Class Unlearning. For single class unlearning, the unlearning set $\mathcal{D}_{tr}^u = \{(x_i, y_i) | y_i = c\}$ and remaining set $\mathcal{D}_{tr}^r = \{(x_i, y_i) | y_i \neq c\}$. This makes the positive set $P_{\mathbf{z}} = \emptyset$ as none of remaining samples belong to class c. In short, there are no positive remaining samples to push away the unlearning samples. Thus we change equation 5 as follows.

$$\mathcal{L}_{UL} = \sum_{x_i \in X^u} \frac{-1}{|N_{\mathbf{z}}(x_i)|} \sum_{z_a \in N_z} \log \frac{\exp(z_i \cdot z_a / \tau)}{|N_z(x_i)|}. \tag{6}$$

We replaced the previous denominator to $|N_{\mathbf{z}}(x_i)|$. This is because equation 5 requires both directions to push and pull unlearning samples. Lacking one of the directions increases the instability, as it can lead to representation collapse [Chen and He, 2021]. Since $P_z = \emptyset$, we replace the denominator to $|N_{\mathbf{z}}(x_i)|$ to introduce damping effects against excessively pulling unlearning samples to negative samples.

Classification Loss of Remaining Samples. A novel challenge of contrastive unlearning is to preserve embeddings of remaining samples. Optimizing equation 5 not only alters embeddings of the anchor unlearning sample but also reciprocally alters embeddings of all samples in $P_{\mathbf{x}}$ and $N_{\mathbf{x}}$. All positive samples are slightly pushed away from and all negatives are slightly pulled toward the anchor. A similar effect arises in contrastive learning, but it is not problematic as it reinforces the consolidation of embeddings of the same class. However, for unlearning purposes, embeddings of X^r have to be preserved, because: 1) not preserving them directly leads to a loss in model performance, and 2) it also reciprocally affects unlearning effectiveness as magnitude of pulling and pushing decreases. In short, embeddings of X^r are also modified as a byproduct of optimization and it is necessary to restore them back. We utilize cross-entropy loss for restoring embeddings of X^r, because it derives maximum likelihood independently to each sample [Shore and Johnson, 1981]. This ensures obtaining directions very close to the original embeddings. Combining the unlearning loss, the final loss for our proposed contrastive unlearning is as follows,

$$\mathcal{L} = \lambda_{UL}\mathcal{L}_{UL} + \lambda_{CE}\mathcal{L}_{CE}\left(\mathcal{F}(X^r), Y^r\right) \tag{7}$$

where X^r and Y^r are the sampled batches of remaining samples and their corresponding labels. λ_{CE} and λ_{UL} are hyperparameters to determine influence of the two loss terms. The full algorithm is in Appendix B.

Termination Condition. Pinpointing the right moment to terminate the unlearning process is crucial, as terminating too early or too late will lead to insufficient unlearning or poor model utility. None of existing works explicitly discuss the termination condition. We design explicit termination conditions for both class and sample unlearning based on our unlearning goals. We assume a small dataset $\mathcal{D}_{\text{eval}}$ is available for determining the termination condition. We evaluate the conditions every unlearning round.

For class unlearning, recall our problem definition in 3.1 and the goal in equation 1. We can set $\mathcal{D}_{\text{eval}} = D^u_{ts}$, the test data of the unlearning class. Ideally, we want $\mathcal{F}'$ to have close to 0 accuracy for the unlearning class. However, this can be too strict for termination. We loosen the condition and terminate the algorithm when the accuracy of $\mathcal{F}'$ on the unlearning class falls below a threshold. We set the threshold to be $1/C$ where C is the total number of classes in the training data and $1/C$ corresponds to the accuracy of a random guess, which suggests knowledge about the unlearning class is sufficiently removed from the model.

$$\text{Acc}\left(\mathcal{F}', \mathcal{D}_{\text{eval}}\right) \leq \frac{1}{C}. \tag{8}$$

For sample unlearning, recall our problem definition of 3.2 and the goal in equation 3. Ideally, we want the accuracy of unlearning samples by the unlearned model to be similar to the accuracy of the test samples by the retrained model. Since we do not have access to the retrained model, we use a proxy criteria which requires the accuracy of the unlearning samples to be similar to the test samples by the same unlearned model. Specifically, we set $\mathcal{D}_{\text{eval}} = \{\mathcal{D}^u_{eval}, \mathcal{D}^{ts}_{eval}\}$ where $\mathcal{D}^u_{eval} \subseteq \mathcal{D}^u_{tr}$ and $\mathcal{D}^{ts}_{eval} \subseteq \mathcal{D}_{ts}$. The algorithm terminates when the accuracy of $\mathcal{F}'$ on $\mathcal{D}^u_{eval}$ drops below the accuracy on $\mathcal{D}^{ts}_{eval}$.

$$\text{Acc}\left(\mathcal{F}', \mathcal{D}^u_{eval}\right) \leq \text{Acc}\left(\mathcal{F}', \mathcal{D}^{ts}_{eval}\right). \tag{9}$$

Intuitively, it is not desired to terminate the algorithm before satisfying the condition in 9 because it implies that the model still retains information regarding $\mathcal{D}^u_{tr}$. It is also not desired to continue running the algorithm to further reduce accuracy on $\mathcal{D}^u_{tr}$ much lower than $\mathcal{D}_{ts}$ because it is negatively injecting information regarding $\mathcal{D}^u_{tr}$ into θ'. This results in $\mathcal{F}'$ to deliberately make incorrect classification on $\mathcal{D}^u_{tr}$, which is not aligned with our goal of sample unlearning.

5 Experiments

5.1 Experiment Setup

Datasets and Models. We use three benchmark datasets: CIFAR-10, SVHN, and Mini-Imagenet [Cao, 2022], and employ ResNet (RN)-18, 34, 50, and 101 models [He *et al.*, 2016] and ViT-small [Dosovitskiy *et al.*, 2021]. We report the results of CIFAR-10 and Mini-Imagenet in the main paper. Please refer to the appendix for details on the models, implementations (code), results of SVHN, additional experiments on CIFAR-10 and Mini-Imagenet, and parameter studies. We use CLIP model [Radford *et al.*, 2021], and a different contrastive framework MOCO [He *et al.*, 2020] to show the generalizability and scalability.

Comparison Methods. For class unlearning, we remove all samples belonging to class 5 by default. For sample unlearning, we remove randomly selected 500 samples by default. We also evaluate class unlearning on other classes and sample unlearning of varying number of samples. Please refer to Appendix D.3 and D.5 for results. To assure the robustness, we repeat sample unlearning with a random seed for five times and report the average and standard deviation of the results. We provide **Retrain**, a retrained model using the training data excluding the unlearning data, as a reference.

We include four state-of-the-art (SOTA) methods specifically designed for **single class unlearning**: 1) **Boundary Expansion** [Chen *et al.*, 2023] trains the model using all unlearning samples as a temporary class and then discards the temporary class. 2) **Boundary Shrink** [Chen *et al.*, 2023] modifies the decision boundary of unlearning class to prevent unlearning samples from being classified into the unlearning class. 3) **SCRUB** [Kurmanji *et al.*, 2023] is based on the knowledge distillation, selectively transfers knowledge from the original model to the unlearned model (all information except that of the unlearning class). 4) **UNSIR** [Tarun *et al.*, 2023] uses an iterative process of generating noise that maximizes error in the unlearning class and repairing the classification performance for the other classes.

We include four SOTA methods designed for **sample unlearning**: 1) **Finetune** [Golatkar *et al.*, 2020] iteratively trains the original model using only the remaining samples. 2) **Gradient Ascent** [Golatkar *et al.*, 2020] conducts gradient ascent using unlearning samples. 3) **Fisher** [Golatkar *et al.*, 2020] is a certified unlearning algorithm using randomization with Fisher information matrix. 4) **LCODEC** [Mehta *et al.*, 2022] is also a certified unlearning method with a fast and effective way of obtaining Hessian by importance-based parameter selection.

We note that sample unlearning methods may be used for class unlearning. However, our class unlearning baselines already demonstrated their superiority over the sample unlearning baselines. Hence we do not include them in comparison.

Evaluation Metrics. 1) **Model performance**. For class unlearning, we assess the accuracy of the unlearned model on $\mathcal{D}^r_{ts}$ (test data of remaining classes). For sample unlearning, we evaluate $\mathcal{D}_{ts}$ (test data). 2) **Unlearning efficacy**. For class unlearning, we assess accuracy of the unlearned model on $\mathcal{D}^u_{tr}$ and $\mathcal{D}^u_{ts}$ (training and test data of unlearning class). Successful class unlearning should achieve zero for both. For sample unlearning, we assess accuracy of $\mathcal{D}^u_{tr}$ (unlearning samples). We provide an additional metric of **unlearn score**. It is the absolute difference between the accuracy of test and unlearn samples. A successful sample unlearning should achieve a low unlearn score which means the model perceives unlearning samples and test samples (unseen samples) similarly. The statistical reliability is dependent on the test and unlearn accuracy. 3) **Efficiency** is measured by the runtime of the unlearning algorithm.

Unlearning Verification via MIA. We conduct a membership inference attack (MIA) [Shokri *et al.*, 2017] to verify effectiveness of sample unlearning. Although more robust MIA frameworks are available such as LiRA [Carlini *et al.*, 2022], we used the MIA framework from [Shokri *et al.*, 2017] as our main goal is to fairly compare our contrastive unlearning and other baseline unlearning algorithms and to obtain a generalizable comparison on unlearning efficacy. Refer to appendix C.1 for details of MIA.

We report the **Member prediction rate** defined as num-

	Remain test ↑				Unlearn train ↓				Unlearn test ↓			
Method	RN18	RN34	RN50	RN101	RN18	RN34	RN50	RN101	RN18	RN34	RN50	RN101
Retrain (Reference)	65.62	67.64	70.57	71.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Contrastive	**60.69**	**57.61**	**58.81**	**58.53**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Boundary Shrink	10.17	14.88	-	-	0.00	0.00	-	-	0.00	0.00	-	-
Boundary Expansion	51.26	26.89	-	-	0.00	0.00	-	-	0.95	0.00	-	-
SCRUB	50.20	26.57	22.03	12.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
UNSIR	17.05	12.32	12.74	8.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 1: Performance evaluation for single class unlearning on Mini-Imagenet

Model	Retrain (Reference)	**Contrastive**	Boundary Shrink	Boundary Expansion	SCRUB	UNSIR
RN18	329.23	**4.51**	7.99	8.38	12.54	5.74
RN34	468.89	**8.25**	14.76	12.82	21.54	10.05
RN50	911.55	**16.01**	-	-	47.50	20.95
RN101	1473.07	**26.94**	-	-	76.17	31.01

Table 2: Running time of class unlearning on Mini-Imagenet (Minutes).

ber of positive (member) predictions by the MIA divided by total number of tests. It can be considered as false positive rate (FPR) for unlearning samples (considering them as non-members) and true positive rate (TPR) for members. An effective unlearning algorithm should have a low member prediction rate on unlearning samples and high member prediction rate on member samples. Our metric is consistent with existing literature [Jia *et al.*, 2023] utilizing true negative rate (TNR) for unlearning samples and test non-member samples.

5.2 Results on Single Class Unlearning

Unlearning Efficacy and Model Performance. Table 1 shows the accuracy of unlearned models on Mini-Imagenet. Results of other classes are consistent. Readers may refer to Appendix D.3. We only report the average and omit standard deviation since all of them are very small (<0.01). The retrain shows an ideal unlearning with good performance and zero accuracy for both unlearn train and unlearn test sets. Contrastive unlearning achieves zero unlearn accuracy across all models with the smallest performance loss, showing completeness in unlearning while preserving model utility. Compare to the experiments on CIFAR-10 and SVHN datasets, the utility loss is bigger on unlearning Mini-Imagenet dataset. We presume that it is due to the large number of classes as model could exhibit more intricate decision boundaries. We did not report experiments of ViT models, Boundary Shrink and Boundary Expansion for ResNet50 and ResNet101 because they required excessive computational resources and resulted out-of-memory error.

Efficiency. Table 2 depicts the elapsed time for each unlearning algorithm of the single class unlearnings. Contrastive unlearning is the fastest compared to all baselines, requiring the smallest number of passes over the entire unlearning samples.

5.3 Results on Sample Unlearning

Model Performance. Table 3 shows the performance metrics of unlearned models. Like the results of the retrained model, successful sample unlearning should achieve high test accuracy (utility) and low unlearn score (effective unlearning). Contrastive unlearning achieved best utility and good unlearn score. LCODEC achieved higher test accuracy on ViT, however, its unlearn score implies unlearning was not complete, resulting in higher test accuracy. Although fine-tuning resulted lower unlearning scores, the difference is not significant and we demonstrate in the following paragraph that contrastive unlearning actually achieves effective unlearning.

Unlearning Efficacy via MIA. Table 4 shows the member prediction rate of the MIA on unlearning samples and test member samples against each unlearned model. An ideal attack model against the retrain model should have zero member prediction rate for unlearning samples and 100% for member samples (since the unlearning samples are non-members). However, the attack model in our experiment shows around 60% for unlearning samples on the retrain model, which is due to the attack power of the attack model. The high rate on member samples suggests it has reasonable attack power in recognizing members. We expect stronger attack methods [Carlini *et al.*, 2022] can better differentiate members and non-members but the comparison of the methods should stay the same. An unlearning algorithm is more effective if it exhibits 1) lower member prediction rate on unlearning samples, and 2) bigger difference in member prediction rate on unlearning samples and member samples. For gradient ascent, Fisher, and LCODEC, the member prediction rate for member samples and unlearning samples are similar, showing ineffective unlearning. For finetune and contrastive unlearning, the member prediction rate for unlearning samples is lower than member samples. However, the difference is significantly bigger in contrastive unlearning, suggesting stronger discrimination between unlearning samples and member samples and more effective unlearning.

Efficiency. Table 5 shows the runtime of different algorithms. It shows contrastive unlearning is the fastest to reach the termination condition. On average, it needed less than 15 unlearning rounds, which is computation equivalent to at most $15 \times \omega$ passess on unlearning dataset. While gradient ascent also iterates only on unlearning dataset, it requires more than 40 passess to achieve unlearning effects, and requires a smaller batch size for the better results. Finetune, Fisher, and LCODEC need longer runtime as they iterate over the entire set of remaining samples. Moreover, Fisher and LCODEC are even slower for bigger models as their computation is proportional to model parameters and hardly parallelizable.

Embeddings visualization. Figure 2 is the visualization of

	Test accuracy ↑					Unlearn accuracy					Unlearn score ↓				
Method	RN18	RN34	RN50	RN101	ViT	RN18	RN34	RN50	RN101	ViT	RN18	RN34	RN50	RN101	ViT
Retrain	84.68±0.23	85.48±0.14	86.44±0.57	85.98±0.13	73.28±0.52	85.30±0.6	85.12±0.21	86.86±0.52	86.11±0.27	73.40±0.82	0.62	0.08	0.42	0.31	0.12
Contrastive	**81.86±0.33**	**83.53±0.54**	**84.80±0.34**	**86.75±0.87**	62.02±0.49	81.69±0.24	81.50±1.4	83.20±0.00	85.34±0.87	59.67±0.90	0.17	2.03	1.6	1.41	2.35
Finetune	81.68±0.29	82.38±0.80	82.60±0.51	83.76±1.16	73.08±2.35	83.65±2.5	82.7±0.89	82.46±1.59	82.23±1.58	96.43±3.23	1.97	0.32	0.14	0.53	23.35
Gradient	67.64±3.41	67.54±3.41	67.70±5.22	76.76±6.71	69.25±3.17	88.65±3.86	88.65±3.86	91.80±1.12	94.18±3.34	95.93±2.59	21.01	12.11	24.10	17.42	26.68
Fisher	76.54±2.34	76.54±2.34	72.03±8.00	82.81±0.83	20.66±3.10	92.83±2.71	92.85±2.73	85.15±12.1	98.30±0.93	24.98±3.30	16.29	16.31	13.12	15.49	4.32
LCODEC	76.20±1.37	81.22±0.85	78.14±1.04	78.62±1.11	**84.54±0.78**	99.65±0.24	99.53±0.23	99.31±0.45	99.08±0.78	89.23±0.97	23.45	18.31	21.17	20.46	4.69

Table 3: Performance evaluation on sample unlearning on CIFAR-10.

Model	Unlearning Samples ↓						Member-test Samples (Reference)					
	Retrain (Ref.)	**Contrastive**	Finetune	Gradient Ascent	Fisher	LCODEC	Retrain (Ref.)	**Contrastive**	Finetune	Gradient Ascent	Fisher	LCODEC
RN18	63.28±0.48	**60.88±0.78**	63.87±0.98	79.85±1.13	85.91±1.26	92.18±1.41	96.08±0.52	91.05±0.59	85.81±1.01	84.62±1.12	89.23±1.31	92.98±0.89
RN34	63.81±0.55	**53.51±0.58**	66.65±0.87	83.08±0.99	82.59±1.10	95.49±1.13	94.82±0.32	86.44±0.46	86.99±0.84	84.01±1.18	83.74±0.98	97.21±1.21
RN50	63.04±0.29	**60.87±0.64**	68.47±0.89	85.87±1.08	74.46±1.42	93.98±1.35	97.43±0.47	91.13±0.54	84.03±0.93	89.29±1.29	77.15±1.68	93.59±1.56
RN101	62.49±0.51	60.79±0.78	**54.89±0.99**	91.98±1.14	84.20±1.86	94.93±1.53	95.74±0.62	86.45±0.92	62.39±1.05	90.47±0.89	84.90±1.77	95.10±1.68
ViT	53.57±0.38	**55.49±0.74**	84.97±1.04	56.58±1.23	56.18±1.59	83.99±1.48	89.29±0.76	72.87±0.69	85.92±1.18	57.49±1.44	59.86±0.88	87.12±1.43

Table 4: Member prediction rate on unlearning samples and member-test samples (memorized train samples) of MIA on CIFAR-10 dataset.

Method	RN18	RN34	RN50	RN101	ViT
Retrain	43.05±2.18	73.22±3.44	134.42±4.72	215.84±4.57	402.15±3.73
Contrastive	**2.68±0.64**	**3.64±0.72**	**8.46±0.98**	**12.63±1.02**	**3.10±0.45**
Finetune	16.93±2.24	31.51±2.21	42.93±3.52	103.74±3.05	79.24±3.61
GA	4.89±0.82	7.52±1.21	14.16±1.46	20.21±1.41	35.65±1.19
Fisher	72.31±1.52	115.51±1.98	219.49±1.95	398.87±1.66	218.93±1.48
LCODEC	34.87±1.87	55.50±1.15	152.28±1.64	449.11±1.31	1719.60±3.41

Table 5: Running time of sample unlearning on CIFAR-10 (minutes)

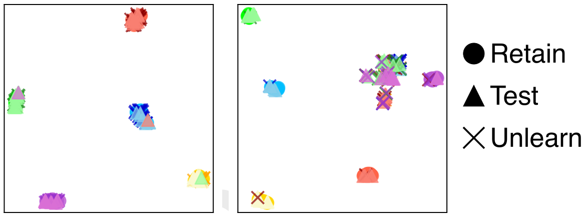

Figure 2: Visualization of the representation space

representation space with Uniform Manifold Approximation and Projection (UMAP). The colors represent each class. The circles, crosses and triangles represent embeddings of remaining, unlearning and test samples respectively. For simplicity, we only show five classes and 128 unlearning, remaining and test samples. The left figure shows the embeddings before unlearning, both unlearning and remaining samples are clustered to their classes. The right figure shows the embeddings after termination condition is satisfied. It clearly shows that representations of unlearning samples are pushed away from their original clusters and closer to test samples while remaining samples are intact.

5.4 Generalizability & Scalability

Generalizability. Contrastive unlearning is highly generalizable across different models due to its nature of optimizing embeddings to achieve unlearning. It can unlearn models beyond the standard classifiers such as vision language models. To demonstrate, we finetune CLIP [Radford *et al.*, 2021] with CIFAR-100 (top-1 accuracy of 82.3%) and unlearn entire class of 1 using contrastive unlearning. We compare results only with gradient ascent and finetune as other baselines are strictly designed for unlearning standard classifiers. Table 6 shows that contrastive unlearning completely removes knowledge of the class from CLIP with small utility loss. Meanwhile, baselines experienced ineffective unlearning or utility loss. Refer to Appendix D.7 for the details.

Method	Remain test ↑	Unlearn train↓	Unlearn test↓
Contrastive	**76.20**	**0.0**	**0.0**
Gradient Ascent	12.42	**0.0**	**0.0**
Finetune	79.87	87.00	68.32

Table 6: Performance evaluation on class unlearning on CLIP

Scalability. Contrastive unlearning is a general framework and can leverage more advanced contrastive "learning" algorithms for enhanced scalability and reduced batch size dependence. To demonstrate, we utilize MoCo [He *et al.*, 2020], a batch-agnostic contrastive learning algorithm as a backbone and compared the unlearning efficacy and model utility with the standard one. The results showed that MoCo-based unlearning significantly outperformed the standard method with small batch sizes. Refer to Appendix D.8 for the details.

6 Conclusion

In this paper, we proposed a novel contrastive approach for machine unlearning. It achieves unlearning by effectively optimizing embedding space and contrasting unlearning samples and remaining samples. Through extensive experiments, we demonstrated that it outperforms state-of-the-art unlearning algorithms in model performance, unlearning efficacy, efficiency and scalability. In future work, we will examine the efficacy of contrastive unlearning in different model architectures and different unlearning scenarios such as graph unlearning and correlated sequence unlearning.

Acknowledgements

This research is partially supported by NSF grants CNS-2124104, CNS-2125530, IIS-2302968, and NIH grants R01LM013712, R01ES033241.

References

[Arpit *et al.*, 2017] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron 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*, pages 233–242. PMLR, 2017. ISSN: 2640-3498.

[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 *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[Bui *et al.*, 2024] Nhung Bui, Xinyang Lu, Rachael Hwee Ling Sim, See-Kiong Ng, and Bryan Kian Hsiang Low. On newton's method to unlearn neural networks. *arXiv preprint arXiv:2406.14507*, 2024.

[Cao and Yang, 2015] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE, 2015.

[Cao, 2022] Xin Cao. MLclf: The Project Machine Learning CLassiFication for Utilizing Mini-imagenet and Tiny-imagenet. Zenodo, October 2022.

[Carlini *et al.*, 2022] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[Cha *et al.*, 2024] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11186–11194, Mar. 2024.

[Chen and He, 2021] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[Chen *et al.*, 2023] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. 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*, pages 7766–7775, 2023.

[Cotogni *et al.*, 2023] Marco Cotogni, Jacopo Bonato, Luigi Sabetta, Francesco Pelosin, and Alessandro Nicolosi. DUCK: Distance-based Unlearning via Centroid Kinematics, December 2023. arXiv:2312.02052 [cs].

[Das and Chaudhuri, 2024] Rudrajit Das and Subhasis Chaudhuri. On the separability of classes with the cross-entropy loss function, 2024.

[Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

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

[Goodfellow *et al.*, 2013] I. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. *CoRR*, December 2013.

[Graf *et al.*, 2021] Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised contrastive learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 3821–3830. PMLR, 2021. ISSN: 2640-3498.

[Guo *et al.*, 2020] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pages 3832–3842. JMLR.org, 2020.

[Gupta *et al.*, 2021] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Advances in Neural Information Processing Systems*, volume 34, pages 16319–16330. Curran Associates, Inc., 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[Jia *et al.*, 2023] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *Neural Information Processing Systems*, 2023.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[Koch and Soll, 2023] Korbinian Koch and Marcus Soll. No matter how you slice it: Machine unlearning with SISA comes at the expense of minority classes. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 622–637, 2023.

[Kurmanji *et al.*, 2023] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning, 2023.

[Lin *et al.*, 2023] Shen Lin, Xiaoyu Zhang, Chenyang Chen, Xiaofeng Chen, and Willy Susilo. Erm-ktp: Knowledge-level machine unlearning via knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20147–20155, 2023.

[Liu *et al.*, 2021] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34, 2021.

[Long *et al.*, 2018] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[Mantelero, 2024] 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, 2024.

[McCandlish *et al.*, 2018] Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[Mehta *et al.*, 2022] Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10422–10431, 2022.

[Neel *et al.*, 2024] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, pages 931–962. PMLR, 2024. ISSN: 2640-3498.

[Nguyen *et al.*, 2020] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33:16025–16036, 2020.

[Nguyen *et al.*, 2022] Quoc Phong Nguyen, Ryutaro Oikawa, Dinil Mon Divakaran, Mun Choon Chan, and Bryan Kian Hsiang Low. Markov chain monte carlo-based machine unlearning: Unlearning what needs to be forgotten. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 351–363, 2022.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Sablayrolles *et al.*, 2019] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, May 2017. ISSN: 2375-1207.

[Shore and Johnson, 1981] J. Shore and R. Johnson. Properties of cross-entropy minimization. *IEEE Transactions on Information Theory*, 27(4):472–482, July 1981. Conference Name: IEEE Transactions on Information Theory.

[Tarun *et al.*, 2023] Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–10, 2023.

[Thudi *et al.*, 2022] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022.

[Ye *et al.*, 2022] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

[Yeom *et al.*, 2018] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[Zhang *et al.*, 2024] Binchi Zhang, Yushun Dong, Tianhao Wang, and Jundong Li. Towards certified unlearning for deep neural networks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 58800–58818. PMLR, 21–27 Jul 2024.