Factor Decorrelation Enhanced Data Removal from Deep Predictive Models

Wenhao Yang
Wuhan University of Technology, China
342471@whut.edu.cn

Lin Li[*]
Wuhan University of Technology, China
cathylilin@whut.edu.cn

Xiaohui Tao
University of Southern Queensland
Queensland, Australia
Xiaohui.Tao@unisq.edu.au

Kaize Shi
University of Southern Queensland
Queensland, Australia
Kaize.Shi@unisq.edu.au

Abstract

The imperative of user privacy protection and regulatory compliance necessitates sensitive data removal in model training, yet this process often induces distributional shifts that undermine model performance-particularly in out-of-distribution (OOD) scenarios. To address this issue we propose a novel data removal approach that enhances deep predictive models through factor decorrelation and loss perturbation. Our approach introduces: (1) a discriminative-preserving factor decorrelation module employing dynamic adaptive weight adjustment and iterative representation updating to reduce feature redundancy and minimize inter-feature correlations. (2) a smoothed data removal mechanism with loss perturbation that creates information-theoretic safeguards against data leakage during removal operations. Extensive experiments on five benchmark datasets show that our approach outperforms other baselines and consistently achieves high predictive accuracy and robustness even under significant distribution shifts. The results highlight its superior efficiency and adaptability in both in-distribution and out-of-distribution scenarios.

1 Introduction

Removing specific data in the machine learning model training process is crucial to protect user privacy and regulatory compliance [1, 2, 3]. For example, users of e-commerce platforms may invoke data deletion rights for product reviews that have been incorporated into the training corpus of recommendation models. Satisfying such requests entails removing the associated entries from front-end systems while also ensuring that the date's influence is purged from the model's internal representations and parameter space. In addition, financial clients can request the removal of transaction histories or loan application records that have contributed to the training of credit scoring models. These scenarios highlight that data removal requests are distributed in widely scenarios with domain-specific regulatory and operational constraints. Furthermore, such removals can induce shifts in the underlying data distribution, while retraining the overall model for each of the specific cases is impractical since the computational costs [4, 5]. Therefore, generalizable approaches to data removal are essential for adapting to varied and evolving deletion demands.

The primary challenge confronting data removal methodologies lies in the inadequate exploration and adaptation to out-of-distribution (OOD) data scenarios [6, 7]. Existing data removal mechanisms

[*] Corresponding author. Email: cathylilin@whut.edu.cn

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

predominantly rely on gradient-based updates and parameter fine-tuning, assuming that the data distribution is similar before and after removal, which inherently limits their robustness. As data distributions may evolve dynamically across temporal and contextual dimensions with continuous data removal requests, the intrinsic correlation between feature representations and corresponding labels transforms. These distributional shifts weaken the effectiveness of existing forgetting mechanisms, reduce removal accuracy, and decrease generalization ability to unseen data. Consequently, existing removal mechanisms struggle to maintain model performance under dynamic scenarios.

Feature dimensionality reduction serves as an effective decorrelation strategy in OOD scenarios and has been widely adopted in predictive modelling techniques, including principal component analysis (PCA) [8, 9], clustering-based approaches [10], and kernel-based mappings [11]. For instance, Stablenet [12] employs Random Fourier features to achieve spatial transformation for classification under OOD conditions. A core challenge lies in integrating dimensionality reduction with existing data removal strategies to design parameter update algorithms to maintain the balance between model accuracy and computational efficiency. While dimensionality reduction reshapes the representation space, it may also discard informative and discriminative features as dimensionality decreases. Loss functions without appropriate adaptation may lead to gradient directions that diverge from the true optimization objective, introducing training instability and degrading generalization performance.

To address the aforementioned challenge, we propose DecoRemoval, a data removal method that avoids retraining under OOD scenarios. In such settings, we introduce a discriminative maintenance factor decorrelation module and use a spatial mapping strategy to efficiently reduce feature dimensionality with linear computational complexity. This transformation is based on the Fourier transform of a kernel function, thereby reducing feature redundancy and promoting factor decorrelation. DecoRemoval maintains feature weights through iterative gradient updates, which accelerates convergence and enhances robustness without assuming a fixed data distribution. To further improve the safety and reliability of the removal process, we introduce a random linear perturbation module for smoothed data removal. This perturbation serves as a regularizer in the parameter space, smoothing the solution landscape of the objective function. As a result, it enables accurate approximation of retraining effects via localized parameter adjustments. Compared with several baselines, the proposed DecoRemoval achieves the best performance in balancing accuracy and efficiency in data removal scenarios. The contributions of this paper can be summarized as follows:

- We propose DecoRemoval, a discriminative-preserving factor decorrelation method that integrates feature dimensionality reduction with data removal, which dynamically adjusts feature weights to balance removal precision and computational efficiency.
- We design a smoothed data removal mechanism incorporating a Loss Perturbation module, which introduces linear interference to protect sensitive information while preserving model stability during the removal process.
- We conduct extensive experiments on standard benchmarks, showing that DecoRemoval achieves competitive predictive performance, robust generalization, and high efficiency under significant distributional shifts.

2 Related Work

2.1 Machine Unlearning

Machine unlearning has emerged as a pivotal area of research in response to growing privacy concerns and regulatory mandates [13, 14, 15, 16]. This field focuses on developing methodologies that enable machine learning models to effectively remove the influence of specific data points without necessitating complete retraining [17, 18].

As concerns about data privacy and regulatory compliance continue to grow, the ability of machine learning models to "forget" specific data points has emerged as a key area of research. Machine forgetting aims to remove the influence of individual data without retraining the entire model [19]. Early approaches include using gradient vectors or summary layers to isolate and mitigate data influence. Existing methods for forgetting in deep neural networks can be broadly categorized into two groups: retraining-based and retraining-free approaches. Retraining-based methods involve re-optimizing the model after data removal, while retraining-free methods avoid this by estimating the sensitivity of model parameters [20, 21]. These methods often rely on approximations using the Fisher information matrix or the Hessian matrix, as seen in early techniques such as Certified Removal and Optimal Brain Damage [22, 23].

A major challenge in this space has been adapting these forgetting methods to the complex, nonconvex landscape of deep neural networks [24, 25]. To address this, Zhang et al. (2024) extend certified unlearning to deep models by bounding the error introduced by a Newton update, enabling scalable and theoretically grounded forgetting in nonconvex settings [26]. Building on this direction, Foster et al. (2024) introduce Selective Synaptic Dampening, a method that identifies parameters most relevant to the forget set using Fisher information and proportionally reduces their impact [27]. This allows the model to unlearn specific data while maintaining performance on the remaining dataset.

However, in the dynamic environment, data distribution will evolve over time. The existing machine unlearning methods face serious limitations when dealing with scenarios where data distribution is changing. Our method introduces the feature decoupling module into deep neural networks, providing a balance between efficiency and adaptability in both in distribution and out of distribution settings.

2.2 Certified Data Removal

Certified data removal methods allow models to "forget" specific data points while maintaining statistical equivalence to models trained without the removed data [28, 29, 30, 31]. The requirement for data removal speed in practical application scenarios of machine learning cannot be ignored. Certified removal stands out for providing a favorable balance between removal speed and accuracy [13]. It can ensure removal accuracy to a certain extent while maintaining extremely high practical efficiency, making it a leading SOTA method in current research literature.

Certified data removal typically adjusts model parameters by removing residual influences of removed samples, often through gradient-based updates and calibrated noise injection [32, 31, 33]. Marchant et al. [14] pioneered a verification framework for unlearning by analyzing the Hessian matrix of training data and gradients associated with removable samples. Their method triggers retraining if theoretical error bounds exceed predefined thresholds. Subsequent work by Neel et al. [15] introduced regularized and distributed gradient descent variants, providing formal guarantees on model indistinguishability and accuracy for weakly convex loss functions. Guo et al. [13] advanced these principles for linear classifiers, delivering practical algorithms with theoretical rigor.

Our work restricts the scope to the more mature unlearning area of classification tasks. Based on certified removal, we introduce feature decoupling and loss perturbation modules to enable a good approximation of retraining prediction performance after sample removal through localized updates.

3 Factor Decorrelation Enhanced Data Removal

In this section, we will detail the design of our DecoRemoval framework illustrated in Figure 1. DecoRemoval include two main modules: (1) Discriminative-preserving factor decorrelation by using random Fourier features to achieve spatial mapping and perform dimensionality reduction on input features (Section 3.2); (2) Smoothed data removal by integrating the random linear perturbation loss into unlearning training process to ensure privacy and security (Section 3.3). Moreover, we will integrate the core steps of 3.2 and 3.3 and introduce the main process of the DecoRemoval algorithm (Section 3.4). Next, we will explain them one by one.

3.1 Definitions

Definition: ***Factor Decorrelation*** [6, 12]: *Let $X \in \mathbb{R}^{n \times d}$ denote a dataset with n samples and d features, and let $\mathcal{A}$ be a learning algorithm trained on X. Factor Decorrelation refers to the process of reducing statistical dependencies (e.g., correlation) among features in X, with the objective of transforming it into a decorrelated representation X' that preserves essential information for learning. As the correlations between features affect or even impair the model prediction, several works have focused on remove such correlation in the training process such as Random Fourier Features(RFF) [34]. RFF is used to approximate kernel functions and induce decorrelation by mapping the data to a higher-dimensional space. Given a kernel $k(x, y)$, RFF provides a feature*

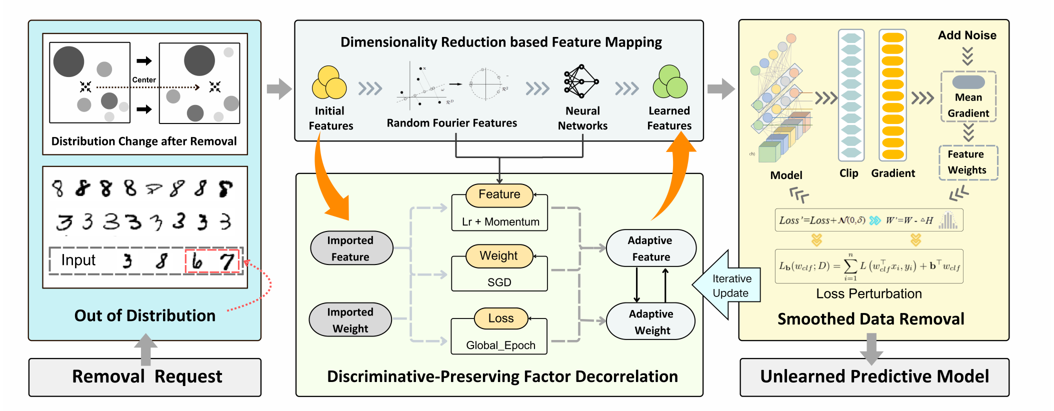

Figure 1: Factor decorrelation enhanced data removal. Overview of the DecoRemoval framework. It consists of two main modules: (1) Discriminative-Preserving Factor Decorrelation based on Random Fourier Features for spatial mapping and dimensionality reduction; (2) Smoothed Data removal through random linear perturbation loss integrated into the unlearning training process.

mapping $\phi(x)$ such that the dot product $\langle \phi(x), \phi(y) \rangle \approx k(x, y)$. The transformation is given by:

$$\phi(x) = \sqrt{\frac{2}{d}} \left[\cos(Wx + b) \right], \tag{1}$$

where $W \in \mathbb{R}^{d \times d}$ is a random matrix, $b \in \mathbb{R}^{d}$ is a bias term, and d is the number of random Fourier features. This transformation is designed to decorrelate the original features by embedding them in a higher-dimensional space.

Definition: ***Certified Removal [13]****: Let D be a training dataset and A be a learning algorithm trained on D. $Range(A)$ is the value range of A. A data-removal mechanism M is applied to $A(D)$, and we say that the removal mechanism M performs ϵ-certified removal (ϵ-CR) for learning algorithm A if, for all $S \subseteq Range(A)$ and $x \in D$, the following condition holds:*

$$e^{-\epsilon} \leq \frac{P(M(A(D), D, x) \in S)}{P(A(D \setminus x) \in S)} \leq e^{\epsilon} \tag{2}$$

The definition ensures that removing a single data point x from the dataset D will not affect the model's predictions by more than an exponential factor of ϵ, preserving the model's stability.

3.2 Discriminative-Preserving Factor Decorrelation

To deal with OOD, we propose a discriminative-preserving factor decorrelation module that integrates Random Fourier Features (RFF) [12] into the neural network architecture. This module aims to decorrelate input features while preserving their class-discriminative structure, thereby promoting stable and generalizable learning. Specifically, input features are first projected into a higher-dimensional randomized feature space via an RFF-based transformation. This mapping reduces redundancy and statistical dependency among features, resulting in a smoother optimization landscape for subsequent layers.

While RFF aids in feature decorrelation, it may also disperse discriminative information across dimensions. Applying standard dimensionality reduction or naive loss functions without adapting to the transformed structure risks misaligning gradient directions with the true task objective. Such mismatch can destabilize training and impair generalization. To address this, we explicitly balances feature decorrelation with discriminative preservation. By aligning the transformation process with task-aware loss design, our approach maintains effective learning dynamics and avoids representation collapse.

Random Fourier Feature Mapping: Let the input feature vector for the i-th sample be denoted by $X_i \in \mathbb{R}^{m_X}$. The goal is to map X_i into a high-dimensional feature space using the Random Fourier

Feature transformation. This transformation is based on the Fourier transform of a kernel function and is defined as follows:

$$Z_i = \sqrt{2} \cdot \cos(\omega X_i + \phi), \quad \omega \sim \mathcal{N}(0, I), \quad \phi \sim \text{Uniform}(0, 2\pi), \tag{3}$$

where $\omega \in \mathbb{R}^{mz}$ is sampled from a standard normal distribution, and $\phi \in [0, 2\pi]$ is sampled uniformly. The resulting vector $Z_i \in \mathbb{R}^{mz}$ is the transformed feature for the i-th sample. By utilizing this RFF mapping, we can approximate the kernel function $k(X, X')$ in a feature space without directly computing it, enabling the use of linear models in a high-dimensional feature space.

Feature Decorrelation via Sample Weighting: To further eliminate the correlation between the transformed features, we employ a sample weighting strategy that minimizes the dependence between features. Let $Z_{:,i}$ and $Z_{:,j}$ represent the i-th and j-th feature vectors of the transformed input. The goal is to reduce the statistical dependence between all pairs of features in the transformed space.

To achieve this, we utilize hypothesis testing statistics based on the cross-covariance between random variables. Let us define the cross-covariance operator Σ_{AB} between two random variables A and B, with corresponding kernel functions k_A and k_B, as follows:

$$\langle h_A, \Sigma_{AB} h_B \rangle = \mathbb{E}_{AB}[h_A(A) h_B(B)] - \mathbb{E}_A[h_A(A)] \mathbb{E}_B[h_B(B)], \tag{4}$$

where $h_A \in \mathcal{H}_A$ and $h_B \in \mathcal{H}_B$ are elements of the Reproducing Kernel Hilbert Spaces (RKHS) corresponding to the random variables A and B. The independence of the random variables A and B is indicated by the condition:

$$\Sigma_{AB} = 0 \iff A \perp B. \tag{5}$$

In our case, we use the cross-covariance between the transformed features $Z_{:,i}$ and $Z_{:,j}$ to measure their dependence. The partial cross-covariance matrix $\hat{\Sigma}_{AB}$ can be estimated as follows:

$$\hat{\Sigma}_{AB} = \frac{1}{n-1} \sum_{i=1}^{n} \left[\left(u(Z_i) - \frac{1}{n} \sum_{j=1}^{n} u(Z_j) \right)^T \cdot \left(v(Z_i) - \frac{1}{n} \sum_{j=1}^{n} v(Z_j) \right) \right], \tag{6}$$

where u and v are the RFF transformations applied to the features Z_i and Z_j, respectively. The Frobenius norm of this matrix is used as a measure of the dependence between features:

$$I_{AB} = \|\hat{\Sigma}_{AB}\|_F^2. \tag{7}$$

To further reduce feature dependence, we apply sample weighting. Let w_i denote the sample weight for the i-th sample. The weighted partial cross-covariance matrix is computed as:

$$\hat{\Sigma}_{AB;w} = \frac{1}{n-1} \sum_{i=1}^{n} \left[\left(w_i u(Z_i) - \frac{1}{n} \sum_{j=1}^{n} w_j u(Z_j) \right)^T \cdot \left(w_i v(Z_i) - \frac{1}{n} \sum_{j=1}^{n} w_j v(Z_j) \right) \right]. \tag{8}$$

Optimization of Sample Weights: The optimal sample weights w^* are determined by minimizing the total dependence between all pairs of features. The optimization problem for the weights is formulated as:

$$w^* = \arg\min_{w \in \Delta_n} \sum_{1 \le i < j \le m_Z} \|\hat{\Sigma}_{Z_{:,i} Z_{:,j}; w}\|_F^2, \tag{9}$$

where $\Delta_n = \{w \in \mathbb{R}^{n+} \mid \sum_{i=1}^{n} w_i = n\}$ ensures that the sample weights are positive and sum to n.

3.3 Smoothed Data Removal

To enhance the data removal mechanism with generalization across distributions, we propose a smoothed data removal module based on random linear perturbations. Specifically, we inject randomized noise into the training loss, which obfuscates gradient signals associated with removed or irrelevant samples. This smoothing effect suppresses sharp updates caused by individual data points, minimizing their influence on model predictions. As a result, the model becomes less sensitive to the removed data, reducing the risk of information leakage while maintaining stable learning behavior.

Loss Perturbation for Data Removal: To ensure that the removal of data does not inadvertently leak information about the removed samples, we begin by applying a loss perturbation technique at the training stage. This involves perturbing the loss function by adding a random linear term:

$$L_{\mathbf{p}}(w_{clf}; D) = \sum_{i=1}^{n} L\left(w_{clf}^{\top} x_i, y_i\right) + \mathbf{b}^{\top} w_{clf} \tag{10}$$

where $w_{clf} \in \mathbb{R}^d$ denotes the weight vector of the linear classifier (distinct from the sample weights w used for decorrelation), and $\mathbf{b} \in \mathbb{R}^d$ is a random vector sampled from a prescribed distribution (e.g., Gaussian or uniform). The addition of $\mathbf{b}^{\top} w_{clf}$ serves to inject controlled stochasticity into the optimization process, thereby obscuring potential gradient signals associated with specific training instances. This perturbation mitigates the risk of overfitting and strengthens the model's robustness to sample removal under removal guarantees.

Linear Authentication Removal: After the loss perturbation, we proceed to the linear authentication removal step. To perform linear authentication removal, the deep learning network is split into two parts: the feature extraction layer parameters w_{extr} and the linear classification layer parameters w_{clf}. This separation allows us to rewrite the loss function in terms of the linear classifier:

$$L(w_{clf}; D) = \sum_{i=0}^{n} L\left(w_{clf}^{\top} f(w_{extr}; f(w^0; x_i)), y_i\right) \tag{11}$$

where $w_{clf}^* = A(D) = \arg\min_{w_{clf}} L(w_{clf}; D)$. We assume that we aim to remove the last training sample (x_n, y_n) from the dataset D, resulting in a modified dataset $D' = D \setminus (x_n, y_n)$.

To remove the sample (x_n, y_n), we first compute the gradient of the loss function at (x_n, y_n) and the Hessian of $L(\cdot; D')$ at w_{clf}^*:

$$\Delta = \nabla L(w_{clf}; (x_n, y_n)) \quad H_{w_{clf}^*} = \nabla^2 L(w_{clf}^*; D') \tag{12}$$

We then apply the Newton update removal mechanism M as follows:

$$w_{clf}^- = M(w_{clf}^*, D, (x_n, y_n)) := w_{clf}^* + H_{w_{clf}^*}^{-1} \Delta \tag{13}$$

This update $H_{w_{clf}^*}^{-1} \Delta$ represents the influence function of the removed training sample on the vector w_{clf}^*.The training process of our DecorRemoval is described in Appendix.A.

Robustness of Removal Under Perturbation. To ensure the proposed removal mechanism remains valid under the perturbed loss, we analyze its impact on the gradient and Hessian. The perturbed loss is given by:

$$L_{\mathbf{p}}(w_{clf}; D) = \sum_{i=1}^{n} L(w_{clf}^{\top} x_i, y_i) + \mathbf{b}^{\top} w_{clf}, \tag{14}$$

where $\mathbf{b} \in \mathbb{R}^d$ is a random vector independent of individual samples. This linear term introduces a constant shift in the gradient:

$$\nabla L_{\mathbf{p}}(w_{clf}) = \nabla L(w_{clf}) + \mathbf{b}, \tag{15}$$

but leaves the Hessian unchanged:

$$\nabla^2 L_{\mathbf{p}}(w_{clf}) = \nabla^2 L(w_{clf}). \tag{16}$$

As the removal update depends on the Hessian and the gradient of the sample to be removed, which are both unaffected by $\mathbf{b}$, the update

$$w_{clf}^- = w_{clf}^* + H^{-1} \nabla L(w_{clf}^*; (x_n, y_n)) \tag{17}$$

remains valid. Hence, our removal mechanism is robust to the proposed linear perturbation, and the specific proof process is detailed in Appendix B.

4 Experiments

4.1 Datasets and Evaluation Metrics

Our approach are evaluated on five widely used datasets spanning multiple data modalities, including image, text, and structured features. Following the setup in Certified Removal [13], we consider three public datasets for classification tasks: MNIST [35], CIFAR-10 [36], and SST-2 [37]. MNIST consists of grayscale images of handwritten digits, where digits 3 and 8 are selected as in-distribution classes and the remaining digits are treated as out-of-distribution (OOD) data. CIFAR-10 contains 60,000 color images (32×32) evenly distributed across 10 object categories and serves as a standard benchmark for evaluating image classification methods. SST-2 is a binary sentiment analysis dataset derived from movie reviews, commonly used in text classification and language understanding tasks.

To assess the algorithm's applicability to privacy-sensitive structured data, we also include two social survey datasets: the 2015 China General Social Survey (CGSS) and the 2018 European Social Survey (ESS). Both datasets contain multi-label annotations related to self-reported happiness levels on a five-point ordinal scale [38, 39].

In the context of unlearning, it is essential to evaluate performance from three key aspects: utility, efficiency, and privacy protection. Given that certified unlearning methods provide theoretical guarantees on privacy, our experiments primarily focus on reporting utility and efficiency. We report accuracy (ACC) and weighted average F1 score as evaluation metrics to capture both overall classification performance and class imbalance sensitivity across diverse data types.

4.2 Experimental Setting

Following certified removal, we split the dataset into training, validation, and testing sets with a ratio of 7:1:1. Both training and validation sets consist solely of correctly labeled data to ensure standard supervised learning conditions. We train models independently on each category (e.g., training with data from class A), and then sample 10% of instances from other categories (e.g., class B). For testing, if these testy samples are predicted as "Not A", it means the unlearned model is correct. This creates an extreme testing scenario where semantic content mismatches the assigned label, enabling evaluation of the model's robustness against mislabeling or conceptual confusion.

In the experiment, the data unlearning baselines include Certified Removal(CR) [13], SISA (5 shards) [3], DP-SGD [40], Certified Unlearning (CU) [26]and SSD [27]. Both CR and DP-SGD provide strong theoretical guarantees under the framework of differential privacy. CU offers model-agnostic approximate unlearning strategies with soundness certificates and can be extended to deep neural networks [26]. SSD leverages the Fisher information matrix for parameter updates and represents the current state-of-the-art in certified removal [27]. In this paper, the source code provided by the baselines is used to fine-tune the parameters and obtain the optimal values to represent the experimental results of the baselines. For baselines which source code was not provided, this study reproduced the model design based on PyTorch framework. All methods set MLP as the backbone for all deep prediction models, ans are trained using a batch size of $d = 50$ and a total of $T = 20$ training epochs. For fair comparison, we fix the standard deviation parameters in DecoRemoval to $\delta = 10^{-3}$ and $std = 1$, and use a consistent optimization schedule with $num_steps = 100$ across all experiments. Under these conditions, we subsequently applied the removal mechanism to each group separately. The related experiments in this study are conducted on four NVIDIA 4090D GPUs.

The time consumption for our DecoRemoval algorithm (DR) to remove data mainly comes from the computation of adaptive weights and the update operation of different features. We evaluate the performance of DecoRemoval algorithm across multiple datasets and varying data remove scales (1000, 3000, and 10000 samples). The full retraining from scratch (Retrain) is treated as the upper bound for accuracy, while Certified Removal serves as the baseline for efficiency comparison [13]. Our main codes and datasets are available at https://github.com/WUT-IDEA.

4.3 Unlearning Performance

Evaluated on five diverse datasets, DecoRemoval consistently achieves near-retraining performance in both accuracy and F1 score across varying removal scales. Unlike prior methods that focus primarily on privacy, our approach addresses feature correlation shifts via spatial mapping and randomized loss

Table 1: Comparison of ACC (%) and F1 scores across different methods and removed sample sizes (The closer to Retrain, the better)

Dataset	Samples	Retrain		CR [13]		SISA [3]		DP-SGD [40]		SSD [27]		CU [26]		DR (Ours)	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
MNIST	1000	51.753	0.505	43.132	0.394	43.635	0.405	45.783	0.440	45.452	0.458	47.345	0.458	**48.973**	**0.482**
	3000	51.351	0.498	42.213	0.391	43.455	0.401	45.342	0.438	45.241	0.455	46.943	0.455	**48.653**	**0.478**
	10000	51.016	0.495	41.872	0.390	42.955	0.398	44.873	0.432	45.031	0.450	46.532	0.450	**48.340**	**0.473**
CIFAR-10	1000	50.762	0.501	43.086	0.392	43.214	0.401	45.301	0.436	45.062	0.452	46.842	0.452	**48.563**	**0.478**
	3000	50.459	0.496	42.293	0.391	42.942	0.396	44.839	0.433	44.723	0.448	46.521	0.449	**48.141**	**0.473**
	10000	50.011	0.491	41.763	0.389	42.512	0.392	44.371	0.428	44.513	0.443	46.106	0.444	**47.832**	**0.469**
SST-2	1000	91.764	0.843	89.705	0.808	89.975	0.817	90.452	0.825	89.983	0.818	89.942	0.813	**90.451**	**0.827**
	3000	91.545	0.840	89.651	0.801	89.760	0.814	90.356	0.820	89.865	0.816	89.765	0.808	**90.387**	**0.825**
	10000	91.142	0.839	89.478	0.796	89.653	0.809	90.101	0.816	89.873	0.816	89.673	0.805	**90.375**	**0.821**
ESS	1000	55.432	0.540	48.608	0.410	48.635	0.420	50.473	0.450	50.147	0.486	51.341	0.490	**54.973**	**0.520**
	3000	55.351	0.540	48.412	0.400	48.455	0.410	50.012	0.440	50.007	0.479	51.153	0.480	**54.852**	**0.510**
	10000	55.236	0.530	48.402	0.390	48.435	0.410	49.673	0.430	49.186	0.478	50.871	0.470	**54.640**	**0.510**
CGSS	1000	51.602	0.515	41.239	0.465	43.516	0.472	46.756	0.487	47.765	0.475	48.765	0.485	**50.824**	**0.501**
	3000	51.324	0.506	40.738	0.458	43.016	0.469	46.252	0.482	47.313	0.474	48.210	0.480	**50.482**	**0.496**
	10000	50.975	0.498	40.145	0.434	42.745	0.465	45.954	0.473	47.152	0.471	47.851	0.477	**50.104**	**0.495**

perturbation, ensuring both utility and robustness. Compared to recent baselines such as SSD and certified unlearning, DecoRemoval demonstrates stronger generalization, especially under large-scale deletions and noisy, high-dimensional data.

DecoRemoval achieved the best performance in out of distribution scenarios across five datasets. As shown in Table ref Table: merged, DecoRemoval consistently achieved near-retraining accuracy and F1 score across five datasets and removal sizes, surpassing all existing Sota data removal models under out-of-distribution settings. On the ESS and CGSS datasets, which feature noisy and highly correlated survey data, our method achieves 54.9% and 50.8% accuracy after removing 1000 samples, with minimal degradation compared to full retraining (55.4% and 51.6%). In SST-2, DecoRemoval maintains over 90.3% accuracy across all remove scales, outperforming DecoRemoval by approximately 1 percentage point on both ACC and F1 metrics. Especially in image dataset scenes, DecoRemoval achieved significant improvement under out of distribution settings. This scenario is the main research and application goal of the current data removal mechanism.

Compared to traditional data removal models that mainly focus on privacy processing, DecoRemoval performs better. The biggest problem with data removal under out of distribution settings is the correlation between features that affects the distribution of data. The existing traditional data removal models mainly focus on privacy protection issues during the data removal process, using methods such as differential privacy to ensure model stability and security. However, when there are out of distribution changes in the data scene, these methods lack processing of the correlation between features, resulting in poor model performance. DecoRemoval identified the complexity of features in this scenario, achieved feature dimensionality reduction through spatial mapping, and ensured the privacy of the removal process by adding random loss perturbations, thus achieving optimal performance in ACC and F1 scores in OOD scenarios.

Greater robustness compared to SOTA. Compared with the latest methods such as SSD updated with Fisher matrix and optimized Certified Unlearning, DecoRemoval utilizes the advantage of spatial dimensionality reduction in feature correlation processing and exhibits stronger robustness in large-scale data deletion. For example, on the CGSS dataset with 10000 removed samples, our method achieved an F1 score of 0.495, while Zhang and Foster's methods only scored 0.477 and 0.471, respectively. This pattern is applicable to all datasets, indicating that our method has stronger generalization ability and stability.

4.4 Efficiency Analysis

In addition to unlearning fidelity, computational efficiency is a critical factor for practical deployment. While full retraining (Retrain) achieves optimal activity performance, it requires model reinitialization

Table 2: Comparison of running Time(s) for different removal methods (the closer to Certified Removal, the better)

Samples	Dataset	Retrain	CR [13]	SISA [3]	DP-SGD [40]	CU [26]	SSD [27]	DR
1000	MNIST	4643.100	**21.312**	1923.430	38.710	33.134	32.541	30.430
	SST-2	61.500	**0.074**	24.670	0.124	0.105	0.095	0.097
	ESS	1539.100	**7.420**	648.400	12.800	12.458	12.127	11.500
	CGSS	615.690	**8.450**	362.400	15.346	14.541	14.377	14.353
3000	MNIST	11432.500	**62.425**	4321.200	102.510	91.329	90.786	81.420
	SST-2	178.100	**0.204**	78.430	0.401	0.325	0.309	0.315
	ESS	3142.200	**22.120**	1448.200	40.500	38.718	37.812	35.200
	CGSS	1923.500	**25.630**	983.200	47.545	45.341	44.749	43.345
10000	MNIST	43123.500	**190.342**	13214.400	310.120	287.490	276.710	268.420
	SST-2	598.400	**0.715**	236.400	1.030	0.904	0.891	0.894
	ESS	14532.500	**78.400**	6534.400	131.100	127.760	123.710	121.400
	CGSS	5893.400	**81.423**	3418.900	164.600	158.860	147.230	141.700

and complete retraining on the original dataset, making it computationally impractical for frequent or large-scale data removal scenarios. In contrast, DecoRemoval delivers strong unlearning performance at significantly lower computational cost. Unlike Retrain, which revisits the entire training set, DecoRemoval performs a lightweight fine-tuning procedure that specifically targets the removal-induced dominant feature directions, enabling it to efficiently handle removal scales ranging from 1,000 to 10,000 samples without full model retraining. Furthermore, compared to existing methods, DecoRemoval achieves better performance with lower overhead. Certified Removal and SISA rely on ensemble models or shard-based training pipelines, which incur significant complexity and computational burden [3, 13]. While DP-SGD offers built-in privacy guarantees, it injects substantial noise during training, resulting in lower post-removal accuracy despite its relative efficiency [40]. Finally, DecoRemoval strikes a more favorable balance between efficiency and unlearning quality than recent approaches proposed by Certified Unlearning (2024) and SSD (2024) [26, 27]. Across all settings, it consistently maintains performance close to the retraining upper bound while requiring far fewer computational resources, making it particularly suitable for real-world, large-scale, or streaming environments where fast and effective unlearning is essential.

4.5 Empirical Evaluation of Privacy Protection via Membership Inference Attacks

To further validate the privacy-preserving capabilities of DecoRemoval in the context of compliant machine learning (e.g., GDPR, CCPA), we evaluate its robustness against *Membership Inference Attacks* (MIAs)—a canonical threat model where an adversary attempts to determine whether a specific data sample was used in model training. Strong resistance to MIAs indicates effective data removal and reduced risk of information leakage.

Our evaluation is conducted under a realistic forgetting scenario with a privacy budget of $\epsilon = 1$. We compare DecoRemoval against several baselines. We assess MIA success rates across three model architectures (MLP, LSTM, Transformer) and two real-world datasets (ESS, CGSS), while also reporting accuracy and F1-score to evaluate utility preservation.

The results are summarized in Table 3. Key observations include:

- DecoRemoval consistently achieves lower MIA success rates than No-DP, demonstrating effective privacy protection after data removal.
- It outperforms SSD and CR across most settings, especially on Transformers, with better utility.
- On CGSS with LSTM, it achieves the lowest attack rate (50.2%) while maintaining higher accuracy than CR.
- Compared to DP-SGD, DecoRemoval offers significantly better utility at comparable privacy levels.

Table 3: Performance comparison of DecoRemoval and baselines on ESS and CGSS datasets across different architectures. Metrics: Accuracy (%), F1-score (%), and MIA success rate (%). Lower attack rate is better.

Backbone	Method	ESS ACC ↑	ESS F1 ↑	ESS Attack ↓	CGSS ACC ↑	CGSS F1 ↑	CGSS Attack ↓
MLP	Retrain	**63.9**	**62.5**	66.1	**60.4**	**48.2**	61.4
	DP-SGD	56.8	54.4	56.8	51.7	43.5	**51.8**
	SSD	59.0	56.9	57.3	53.2	45.2	52.5
	CR	60.7	59.8	**56.7**	55.6	45.9	52.4
	DecoRemoval	61.5	59.6	56.8	56.3	45.7	52.1
LSTM	Retrain	**65.8**	**65.2**	65.3	**62.5**	**56.4**	59.8
	DP-SGD	52.5	48.5	56.5	52.1	49.1	51.6
	SSD	55.6	53.2	56.8	53.8	53.2	50.9
	CR	57.1	53.5	**55.9**	55.4	54.1	50.4
	DecoRemoval	58.9	55.4	56.7	56.8	55.1	**50.2**
Transformer	Retrain	**65.6**	**65.0**	69.2	**61.5**	**55.6**	63.3
	DP-SGD	54.1	52.5	62.0	53.9	51.6	59.1
	SSD	57.3	55.8	62.3	54.8	52.7	58.8
	CR	57.7	56.1	**60.5**	55.4	52.9	**57.4**
	DecoRemoval	58.4	56.3	60.9	56.2	52.8	58.1

These results confirm that DecoRemoval not only stabilizes model distribution after data removal but also provides strong empirical privacy guarantees, making it suitable in privacy-sensitive applications.

4.6 Key Parameter Study

In Figure 2, we map the MINST dataset to a two-dimensional space for visualization after dimensionality reduction. It can be observed that after data removal, the distribution of the dataset and the position of the center point have undergone significant changes, which is consistent with our initial hypothesis about the impact of data removal. Appendix C shows DecoRemoval's performance depends on hidden layer size and RFF dimension, with optimal capacity ensuring good generalization.

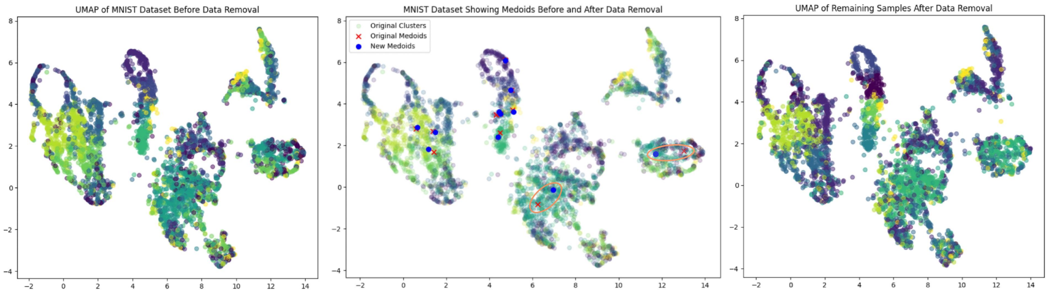

Figure 2: Results of MNIST dataset showing medoids before and after data removal

5 Conclusions and Future work

To tackle privacy preservation and out-of-distribution (OOD) challenges in predictive tasks, we integrate discriminative-preserving factor decorrelation with smoothed data removal. Our DecoRemoval mechanism enables efficient data unlearning while maintaining compliance with privacy regulations. The proposed method alleviates accuracy degradation commonly seen in traditional unlearning approaches under distribution shifts. Empirical results confirm its effectiveness and robustness in OOD scenarios. Future work will focus on applying this algorithm to various application domains that require robust machine unlearning capabilities, and how to work with current popular large models.

6 Acknowledge

This work is partially supported by NSFC, China (No.62276196). The authors also gratefully appreciate the anonymous reviewers for their valuable comments and constructive suggestions that greatly helped to improve the quality of the paper.

References

[1] Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan S. Kankanhalli. Fast yet effective machine unlearning. *IEEE Trans. Neural Networks Learn. Syst.*, 35(9):13046–13055, 2024.

[2] 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.

[3] Youyang Qu, Xin Yuan, Ming Ding, Wei Ni, Thierry Rakotoarivelo, and David B. Smith. Learn to unlearn: Insights into machine unlearning. *Computer*, 57(3):79–90, 2024.

[4] Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. ARCANE: an efficient architecture for exact machine unlearning. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4006–4013. ijcai.org, 2022.

[5] 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 2023, Raleigh, NC, USA, February 8-10, 2023*, pages 622–637. IEEE, 2023.

[6] Daniel Gedon, Antônio H. Ribeiro, Niklas Wahlström, and Thomas B. Schön. Invertible kernel PCA with random fourier features. *IEEE Signal Process. Lett.*, 30:563–567, 2023.

[7] Kun Fang, Qinghua Tao, Kexin Lv, Mingzhen He, Xiaolin Huang, and Jie Yang. Kernel PCA for out-of-distribution detection. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[8] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014.

[9] HanQin Cai, Jialin Liu, and Wotao Yin. Learned robust PCA: A scalable deep unfolding approach for high-dimensional outlier detection. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 16977–16989, 2021.

[10] Yusuke Endo and Koujin Takeda. L1-regularized ica: A novel method for analysis of task-related fmri data. *Neural Computation*, 36(11):2540–2570, 2024.

[11] Youtian Du, Xue Wang, Yunbo Cui, Hang Wang, and Chang Su. Kernel-based mixture mapping for image and text association. *IEEE Trans. Multim.*, 22(2):365–379, 2020.

[12] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5372–5382. Computer Vision Foundation / IEEE, 2021.

[13] 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 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR, 2020.

[14] Neil G. Marchant, Benjamin I. P. Rubinstein, and Scott Alfeld. Hard to forget: Poisoning attacks on certified machine unlearning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7691–7700. AAAI Press, 2022.

[15] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Algorithmic Learning Theory, 16-19 March 2021, Virtual Conference, Worldwide*, volume 132 of *Proceedings of Machine Learning Research*, pages 931–962. PMLR, 2021.

[16] Guangzhen Yao, Long Zhang, Sandong Zhu, and Miao Qi. Dp-prune: Global optimal strategy for retraining-free pruning of transformer models. In *IEEE International Performance, Computing, and Communications Conference, IPCCC 2024, Orlando, FL, USA, November 22-24, 2024*, pages 1–6. IEEE, 2024.

[17] Ga Wu, Masoud Hashemi, and Christopher Srinivasa. PUMA: performance unchanged model augmentation for training data removal. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8675–8682. AAAI Press, 2022.

[18] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[19] Tianqi Chen, Shujian Zhang, and Mingyuan Zhou. Score forgetting distillation: A swift, data-free method for machine unlearning in diffusion models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[20] Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 843–851. AAAI Press, 2025.

[21] Dasol Choi and Dongbin Na. Distribution-level feature distancing for machine unlearning: Towards a better trade-off between model utility and forgetting. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 2536–2544. AAAI Press, 2025.

[22] Chao Liu, Zhiyong Zhang, and Dong Wang. Pruning deep neural networks by optimal brain damage. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors, *15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore, September 14-18, 2014*, pages 1092–1095. ISCA, 2014.

[23] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 598–605. Morgan Kaufmann, 1989.

[24] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3513–3526, 2019.

[25] 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 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9301–9309. Computer Vision Foundation / IEEE, 2020.

[26] Binchi Zhang, Yushun Dong, Tianhao Wang, and Jundong Li. Towards certified unlearning for deep neural networks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[27] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 12043–12051. AAAI Press, 2024.

[28] Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Certified machine unlearning via noisy stochastic gradient descent. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[29] Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Langevin unlearning: A new perspective of noisy gradient descent for machine unlearning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[30] Zijie Zhang, Yang Zhou, Xin Zhao, Tianshi Che, and Lingjuan Lyu. Prompt certified machine unlearning with randomized gradient smoothing and quantization. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[31] Ziyao Liu, Huanyi Ye, Yu Jiang, Jiyuan Shen, Jiale Guo, Ivan Tjuawinata, and Kwok-Yan Lam. Privacy-preserving federated unlearning with certified client removal. *IEEE Trans. Inf. Forensics Secur.*, 20:3966–3978, 2025.

[32] Thanh Trung Huynh, Trong Bang Nguyen, Thanh Toan Nguyen, Phi Le Nguyen, Hongzhi Yin, Quoc Viet Hung Nguyen, and Thanh Tam Nguyen. Certified unlearning for federated recommendation. *ACM Trans. Inf. Syst.*, 43(2):48:1–48:29, 2025.

[33] Yushun Dong, Binchi Zhang, Zhenyu Lei, Na Zou, and Jundong Li. IDEA: A flexible framework of certified unlearning for graph neural networks. In Ricardo Baeza-Yates and Francesco Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 621–630. ACM, 2024.

[34] Zhenyu Liao, Romain Couillet, and Michael W. Mahoney. A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[35] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[36] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[37] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans. Knowl. Data Eng.*, 35(11):11019–11038, 2023.

[38] Theresia Ratih Dewi Saputri and Seok-Won Lee. A study of cross-national differences in happiness factors using machine learning approach. *Int. J. Softw. Eng. Knowl. Eng.*, 25(9-10):1699–1702, 2015.

[39] Zongwen Fan, Jin Gou, and Shaoyuan Weng. A novel fuzzy feature generation approach for happiness prediction. *IEEE Trans. Emerg. Top. Comput. Intell.*, 8(2):1595–1608, 2024.

[40] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016.

A DecoRemoval Training Algorithm

In 3.1 and 3.2, we have introduced feature dimensionality reduction methods using kernel based mapping and certified removal mechanism incorporating random loss perturbations. By utilizing the characteristics of random Fourier transform and sample feature weightings are fused through iterative loss fusion, and feature weights and related parameters are updated.So we summarized the overall algorithm flow of DecoRemoval and introduced the data removal mechanism after adding feature dimensionality reduction.

The DecoRemoval Algorithm aims to minimize feature correlation through the use of Random Fourier Features (RFF) and optimize sample weights efficiently. In the first step, the algorithm prepares the training dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and initializes the neural network's feature extraction weights w_{extr} and classification layer weights w_{clf}. The RFF transformation is applied to map the input features into a higher-dimensional space, resulting in transformed feature vectors Z_i. In the second step,the algorithm calculates the feature dependence between pairs of transformed features and optimizes sample weights w_i to minimize this dependence, achieving decorrelation. For certified data removal in the third step, the influence of each sample is removed by applying a Newton update rule to the classifier weights. To prevent leakage of information from the removed samples, a perturbation term is added to the loss function. The final output includes the optimized classifier weights w^*_{clf} and the transformed feature vectors Z_i.

Algorithm 1: Factor decorrelation enhanced data removal

```
Inputs: training dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, kernel function $k$, neural network with feature
  extraction layer $w_{extr}$, and linear classification layer $w_{clf}$;
Hyperparameters: number of features $m_Z$, number of samples $n$;
Step 1: Random Fourier Feature Mapping;
for $i = 1, \ldots, n$ do
    Sample $\omega_i \sim \mathcal{N}(0, I)$, $\phi_i \sim$ Uniform$(0, 2\pi)$;
    $Z_i =$ RFF_transform$(X_i, \omega_i, \phi_i)$;
Step 2: Discriminative-Preserving Factor Decorrelation;
for each pair $(Z_i, Z_j)$ do
    Compute feature dependence: $I_{ij} =$ FrobeniusNorm$(\hat{\Sigma}_{ij})$;
Apply sample weights $w_i$ to minimize feature dependence: $w^* =$ Optimize Weights$(\hat{\Sigma}_{AB;w})$;
Step 3: Smoothed Data Removal;
for each sample $(x_n, y_n)$ do
    Compute gradient: $\Delta = \nabla L(w_{clf}; (x_n, y_n))$;
    Compute hessian: $H_{w^*_{clf}} = \nabla^2 L(w^*_{clf}; D')$;
    Update classifier: $w^-_{clf} = w^*_{clf} + H^{-1}_{w^*_{clf}} \Delta$;
Add random linear term to the loss: $L_{\mathbf{b}} = L + \mathbf{b}^\top w_{clf}$;
Return: Optimized classifier parameters $w^*_{clf}$ and transformed feature vectors $Z_i$.
```

B Proof of Robustness Under Loss Perturbation

In this section, we provide a formal proof that adding a linear perturbation term to the training loss does not affect the correctness of the linear authentication removal mechanism.

B.1 Perturbed Loss Function

We consider a perturbed loss function of the following form:

$$L_{\mathbf{p}}(w_{clf}; D) = \sum_{i=1}^{n} L\left(w_{clf}^\top x_i, y_i\right) + \mathbf{b}^\top w_{clf}, \tag{18}$$

where $w_{clf} \in \mathbb{R}^d$ is the linear classifier, and $\mathbf{b} \in \mathbb{R}^d$ is a random vector sampled from a fixed distribution (e.g., Gaussian or uniform). The term $\mathbf{b}^\top w_{clf}$ introduces controlled randomness to the optimization process.

B.2 Gradient and Hessian Analysis

Let $L_{\text{orig}}(w_{clf}) = \sum_{i=1}^{n} L(w_{clf}^{\top} x_i, y_i)$ denote the original loss function. Then, the gradient of the perturbed loss is:

$$\nabla L_{\mathbf{p}}(w_{clf}) = \nabla L_{\text{orig}}(w_{clf}) + \mathbf{b}. \tag{19}$$

The Hessian of the perturbed loss is:

$$\nabla^2 L_{\mathbf{p}}(w_{clf}) = \nabla^2 L_{\text{orig}}(w_{clf}) + \nabla^2(\mathbf{b}^{\top} w_{clf}) = \nabla^2 L_{\text{orig}}(w_{clf}), \tag{20}$$

since the second derivative of a linear term is zero. Thus, the curvature of the loss landscape (captured by the Hessian) remains unchanged by the perturbation.

B.3 Effect on Removal Update

Assume that we wish to remove the final sample (x_n, y_n) from the dataset D, yielding the modified dataset $D' = D \setminus (x_n, y_n)$. The Newton-based removal update is given by:

$$w_{clf}^{-} = w_{clf}^{*} + H^{-1} \nabla L(w_{clf}^{*}; (x_n, y_n)), \tag{21}$$

where w_{clf}^{*} is the minimizer of the loss (perturbed or unperturbed), and H is the Hessian of the loss over D' evaluated at w_{clf}^{*}.

Under the perturbed loss, we denote the minimizer as w_{clf}^{*p}, which satisfies:

$$\nabla L_{\text{orig}}(w_{clf}^{*p}) + \mathbf{b} = 0 \quad \Rightarrow \quad \nabla L_{\text{orig}}(w_{clf}^{*p}) = -\mathbf{b}. \tag{22}$$

However, this constant offset in the gradient does not affect the sample-specific gradient $\nabla L(w_{clf}; (x_n, y_n))$, nor the Hessian H, since both are independent of $\mathbf{b}$. Therefore, the removal update remains:

$$w_{clf}^{-} = w_{clf}^{*p} + H^{-1} \nabla L(w_{clf}^{*p}; (x_n, y_n)), \tag{23}$$

which is structurally identical to the original update formula. As a result, the removal mechanism is preserved under loss perturbation.

B.4 Conclusion

The addition of a linear perturbation term does not interfere with the linear removal update. The gradient shift induced by $\mathbf{b}$ is constant and does not impact the relative influence of any individual data point. The Hessian remains unchanged, and the Newton update retains its validity. This confirms the robustness of our removal strategy under smoothed loss perturbations.

C In-distribution Setting Experiment

For the in distribution data scenario, we evaluated the accuracy and F1 score of Certified Removal(CR), DecoRemoval(DR), Certified Unlearning(CU), and SSD on different datasets. This setting enables us to systematically evaluate the robustness of the model to domain differences and its generalization ability when the training distribution remains largely unchanged.

Table 4: Comparison of ACC (%) and F1 scores across different methods under in-distribution setting

Method	**MNIST**		**CIFAR10**		**SST-2**		**ESS**		**CGSS**	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
CR	91.974	0.930	90.878	0.931	96.103	0.979	92.863	0.939	92.798	0.921
SSD	92.429	0.940	92.320	0.964	96.982	0.987	93.754	0.943	93.305	0.930
CU	94.213	0.964	93.271	0.946	96.923	0.982	93.671	0.950	93.193	0.945
DR (Ours)	**95.841**	**0.968**	**95.034**	**0.963**	**97.011**	**0.988**	**93.852**	**0.969**	**93.860**	**0.948**

D Unlearning Performance

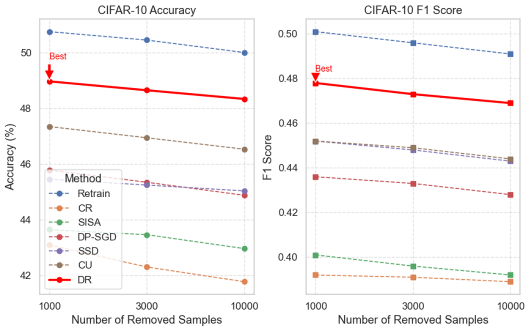

Figure 3: Removal performance of different numbers of removed samples in CIFAR-10

Accuracy and F1 Score.
These results demonstrate the effectiveness of DecoRemoval, our factor decorrelation-based data removal strategy. By adaptively reweighting feature dimensions to suppress redundant correlations, DecoRemoval eliminates the need for full retraining, ensuring fast and efficient data removal. It performs well across various datasets and conditions, providing high forgetting fidelity and accuracy while balancing computational efficiency and rigorous unlearning, making it ideal for privacy-sensitive applications.

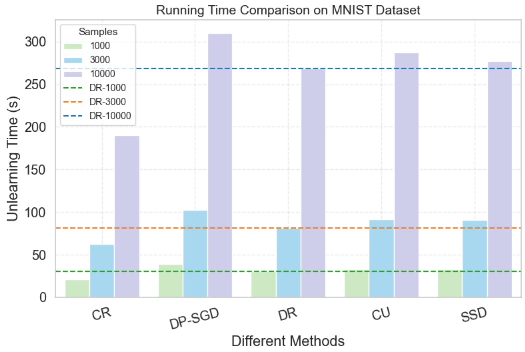

Figure 4: Efficiency of different removed numbers in CIFAR-10 datasets

Efficiency. Overall, our DecoRemoval achieves an effective balance between high forgetting fidelity and practical efficiency, especially when facing out of distribution situations as shown in Figure 4. It can achieve performance close to the level of retraining without incurring the huge cost of comprehensive retraining, and has a significant improvement in the balance between accuracy and efficiency compared to existing data removal mechanisms. This makes it a highly promising solution for scalable and trustworthy machine learning.

E Key Parameter Study

Our DecoRemoval algorithm adjusts the hidden layer dimension and the random Fourier transform dimension of the neural network under out-of-distribution settings, and the tested results are reported in terms of Accuracy and F1 score on the datasets, as shown in Figure 5. Specifically, when the hidden layer dimension is about 80, the accuracy and F1 score of the two happiness datasets ESS and CGSS, as well as the sentiment text dataset SST-2, reach their maximum values, outperforming other traditional models and approaching the results of retraining. For the image dataset MNIST, the performance reaches its best when the hidden layer dimension is about 400.

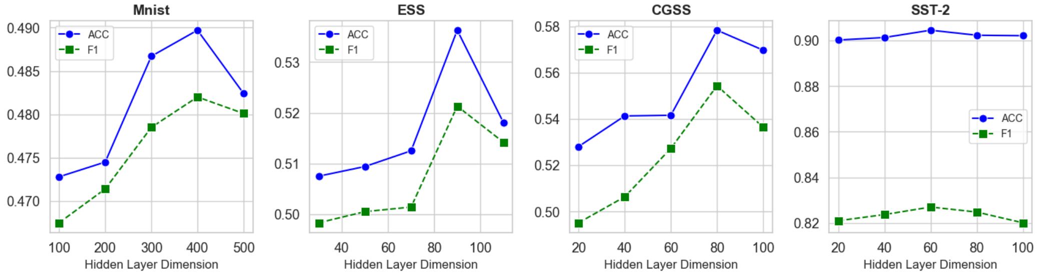

Figure 5: Results of different hidden layer dimensions in adaptive weighted factor decorrelation

The results indicate that the DecoRemoval algorithm can significantly improve its generalization ability in the presence of out-of-distributed data.

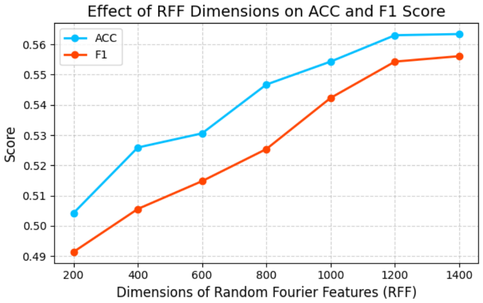

Figure 6: Experimental results of ESS in different RFF dimensions

RFF dimensions

During training with varying RFF dimensions, we observed that both model accuracy and F1 score consistently improved as the dimensionality increased, peaking at 1000 dimensions as shown in Figure 6. Beyond this point, performance gains plateaued, showing minimal change with further increases. These findings suggest that the DecoRemoval algorithm effectively enhances generalization under non-out-of-distribution conditions, particularly when equipped with a sufficiently expressive RFF representation.

F Backbones

Table 5: Comparison of ESS Deletion Efficiency of Different Backbones in Deep Predictive Models (the closer to Retrain, the better)

Backbones	**Retrain**		**Certified Removal**		**FD-DR(ours)**	
	Time(s) ↓	**ACC(%)** ↑	**Time(s)** ↓	**ACC(%)** ↑	**Time(s)** ↓	**ACC(%)** ↑
MLP	1539.1	55.4	7.4	48.6	11.5	**54.8**
LSTM	3583.1	53.9	20.4	48.2	28.6	**52.5**
Transformer	1956.2	54.1	16.4	48.5	15.4	**53.1**

G Limitations

Limited Exploration Beyond Feature-Level Decorrelation. This work primarily focuses on mitigating out-of-distribution (OOD) challenges through feature-level factor decorrelation. While effective, it leaves open how this approach interacts with other common OOD handling techniques such as data augmentation, adversarial training, or ensemble learning. A promising direction for future work is to explore how these methods can be systematically integrated with data removal strategies to enhance both generalization and unlearning robustness.

Trade-off Between Certified Removal and Accuracy. Although certified removal offers strong theoretical guarantees and is efficient for linear layers, it may be suboptimal in scenarios where high predictive accuracy is critical, such as medical diagnosis or financial forecasting. In such cases, privacy strength could be moderately relaxed in favor of more expressive unlearning methods, such as influence function-based unlearning or fine-tuning-based approximate unlearning, to strike a better balance between utility and privacy.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The theoretical design effectively reduces feature redundancy and data leakage risk through factor decorrelation and loss perturbation, while experimental results demonstrate consistent performance improvements across multiple datasets and distribution shifts, indicating strong generalizability to various models and real-world data removal scenarios.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper clearly discusses its limitations in Appendix.G. It acknowledges that the current work mainly focuses on feature-level factor decorrelation to address OOD challenges, leaving the integration with other OOD techniques like data augmentation or adversarial training unexplored. Additionally, it notes that while certified removal provides strong theoretical guarantees and efficiency for linear layers, it may not be optimal for applications requiring very high accuracy, suggesting a trade-off between privacy strength and model utility in such scenarios.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For each theoretical result, we provide complete and correct assumptions and proofs. In the main text, we provide core formulas and flowcharts, and provide a complete proof of the reliability of the formulas in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper provides detailed descriptions of the all five datasets, model architectures, training procedures, hyperparameters, and evaluation metrics. Additionally, the experimental setup and parameter settings are clearly documented, enabling reproducibility of the main results that support the paper's claims.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: After the paper review is approved, we will upload the core code, dataset, and other related work to an anonymous URL: https://anonymous.4open.science/r/DecoRemoval-770220/ .

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant training and testing details, including data splits, hyperparameter settings, optimizer type, and selection criteria, are thoroughly described in the Experimental Setting section, ensuring clarity and reproducibility of the reported results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: This article provides a detailed explanation of the types of computing hardware used, including the types and quantity requirements of GPUs. The approximate training time for each experiment provides sufficient details to reproduce the computing environment and resource requirements.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research fully complies with the NeurIPS Code of Ethics, ensuring responsible data usage, privacy protection, and adherence to ethical standards throughout the study.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses positive impacts such as enhanced data privacy protection and improved robustness in machine learning models, as well as potential negative impacts including limitations in privacy-utility trade-offs and challenges in applying the method across all domains.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this work, including datasets and code repositories, have correctly cited the corresponding original references. All code is using the latest publicly available version. The licenses for these assets are recognized and respected according to the terms set by the original creators.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.