A. Related Works

Machine unlearning approaches are designed to expunge information pertaining to a particular subset of training data from the model weights, while maintaining the model performance on the rest of the data. The concept of machine unlearning was first introduced in [3] as an efficient forgetting algorithm tailored for statistical query learning. Bourtoule *et al*. [2] proposed a framework that shards data into multiple models, enabling precise unlearning of specific data segments. This method ensures complete forgetting but incurs significant storage costs due to the need to maintain multiple models or gradients. In the context of model interpretability, Koh *et al*. [20] provided a Hessian-based method for estimating the influence of a training point on the model predictions. Guo *et al*. [14] introduced ϵ-certified removal, which applied differential privacy [7] to certify the data removal process, and proposed a method for removing information from model weights in convex problems using Newton's method. Neel *et al*. [27] proposed a gradient descent-based method for data deletion in convex settings, providing theoretical guarantees for multiple forgetting requests. Although these approaches have been proven effective, they are not fully suitable for deep neural network due to its non-convex nature.

Recently, there have been numerous attempts to address machine unlearning in deep neural networks. Golatkar *et al*. [10,11] took an information-theoretic approach to eliminate data-specific information from weights, leveraging the Neural Tangent Kernel (NTK) theory [18]. Fisher forgetting [11] utilized the Fisher information matrix to identify the optimal noise level required to effectively eliminate the influence of samples designated for unlearning. Liu *et al*. [24] presented that increasing model sparsity can boost effective unlearning, and proposed a unlearning framework that utilizes pruning methods [25] on top of existing unlearning approaches. Chundawat *et al*. [6] used a teacher-student distillation framework, where the student model selectively receives knowledge from both effective and ineffective teachers, facilitating targeted forgetting. Similarly, Kurmanji *et al*. [22] employed a teacher-student network but simplify the approach by using only a single teacher. Our self-distillation loss shares some similarities with these distillation-based approaches, but it offers clear advantages by targeting an equilibrium, resulting in more stable training. Unlike previous unlearning works, our primary focus is on latent feature representation space, aimed at effectively mitigating the information leakage problem associated with machine unlearning.

On the other hand, several works have proposed modifications to the original model training to make the resulting model more amenable to unlearning. Thudi *et al*. [33] introduced a regularizer to reduce the verification error, which approximates the distance between the unlearned model and a retrained model, aiming to facilitate easier unlearning in the future. Zhang *et al*. [38] presented a training process that quantizes gradients and applies randomized smoothing, which is designed to make unlearning unnecessary in the future and comes with certifications under some conditions. However, these approaches assumes that the deletion request does not cause significant changes in data distribution, which is not applicable to practical scenarios such as class unlearning.

B. Implementation details

NMI To calculate the normalized mutual information (NMI), we initially perform k-means clustering on dataset $\mathcal{D}$ based on feature representations, with k equal to the number of classes, Y. Let $\mathbf{K} \in \{1, ..., Y\}^{|\mathcal{D}|}$ represent the cluster assignments for $\mathcal{D}$, and $\mathbf{X} \in \{0, 1\}^{|\mathcal{D}|}$ indicate whether each sample belongs to the forget set. NMI is then computed using the formula $\frac{I(\mathbf{K},\mathbf{X})}{\min(H(\mathbf{K}),H(\mathbf{X}))}$, where $I(\cdot, \cdot)$ is the mutual information and $H(\cdot)$ denotes the entropy.

F1 score To measure the F1 score, we utilize the same k-means clustering. We calculate the recall and precision for each cluster regarding $\mathcal{D}_f$. Precision is defined as the proportion of cluster examples that belong to $\mathcal{D}_f$, while recall is the proportion of $\mathcal{D}_f$ assigned to the cluster. The F1 score is computed by the harmonic mean of recall and precision. We report the final F1 score for the cluster that yields the highest value, indicating the most relevant cluster to $\mathcal{D}_f$.

Membership inference attack (MIA) success rate Following prior work [24], we employ a confidence-based MIA predictor. Given the unlearned model, θ_u, and the datasets, $\mathcal{D}_f$, $\mathcal{D}_r$, and $\mathcal{D}_{\text{test}}$, we first calculate the confidence, denoted as $q(\cdot)$, for each example in the datasets. Then, we train a logistic regression model, $h(\cdot)$, using $\mathcal{D}_r$ and $\mathcal{D}_{\text{test}}$, which aims to predict $h(q(x)) = 1$ for $x \in \mathcal{D}_r$ and $h(q(x)) = 0$ for $x \in \mathcal{D}_{\text{test}}$. We measure the MIA success rate by averaging $h(q(x))$ for all $x \in \mathcal{D}_{\text{f}}$, where the lower values indicate successful unlearning.

Linear probing Given the target model θ, linear probing protocol involves training a new linear classifier on top of its frozen feature extractor. For evaluating LP($\mathcal{D}_r$), the linear classifier is trained with $\mathcal{D}_r$, and we report the performance on the $\mathcal{D}_r$ to

Table A. Unlearning results on the CIFAR-10 dataset averaging over five different configurations.

Method	DA	LP($\mathcal{D}_f$)	LP($\mathcal{D}_r$)	F1	NMI	Acc($\mathcal{D}_f$)	Acc($\mathcal{D}_r$)	MIA
Original	0.34	92.9	92.5	0.99	0.96	92.9	92.0	0.91
Retrained	0.79	65.4	92.1	0.54	0.31	0.0	92.2	0.37
FT	0.51	88.3	92.8	0.80	0.65	40.2	92.8	0.21
FT (classifier only)	0.34	92.9	92.5	0.99	0.96	0.0	92.8	0.00
NegGrad	0.55	66.8	90.2	0.51	0.23	3.7	85.2	0.63
Fisher	0.37	88.5	90.2	0.97	0.89	8.4	88.6	0.01
SCRUB	0.41	74.7	92.0	0.76	0.59	50.2	91.8	0.46
EU-k	0.73	68.1	90.7	0.73	0.46	0.0	90.9	0.19
CF-k	0.60	81.3	**92.1**	0.66	0.43	13.7	92.1	0.15
MUDA (Ours)	**0.82**	**66.4**	92.3	**0.54**	**0.32**	0.0	92.3	0.29

Table B. Unlearning results on the CIFAR-100 dataset averaging over five different configurations.

Method	DA	LP($\mathcal{D}_f$)	LP($\mathcal{D}_r$)	F1	NMI	Acc($\mathcal{D}_f$)	Acc($\mathcal{D}_r$)	MIA
Original	0.50	75.0	72.1	0.73	0.64	76.8	72.3	0.91
Retrained	0.74	50.8	71.2	0.33	0.19	0.0	71.4	0.18
FT	0.58	70.2	70.7	0.71	0.61	44.0	71.2	0.08
FT (classifier only)	0.54	74.2	70.5	0.99	0.97	0.0	66.3	0.01
NegGrad	0.55	52.0	**71.3**	0.73	0.60	20.0	71.2	0.15
Fisher	0.59	67.0	62.7	0.87	0.78	0.0	62.3	0.06
SCRUB	0.60	60.0	70.1	0.68	0.57	32.0	70.3	0.32
EU-k	0.70	35.2	37.1	0.26	0.14	0.0	28.0	0.45
CF-k	0.54	72.6	69.4	0.93	0.90	70.8	69.5	0.81
MUDA (Ours)	**0.73**	**37.0**	**71.1**	**0.34**	**0.21**	0.0	71.1	0.15

focus on the discriminability of the retain samples. To measure LP($\mathcal{D}_f$), we train a linear classifier using $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$, and report the performance on $\mathcal{D}_f$, which is for evaluating the identifiability of $\mathcal{D}_f$.

Hyperparameters We tune the learning rate for all compared approaches within $\{0.1, 0.01, 10^{-3}, 10^{-4}\}$, except for the NegGrad, for which we use $\{10^{-4}, 10^{-5}\}$. For EU-k and CF-k, we follow the same k with prior work [9], updating the conv4 and fc layers of ResNet while keeping the other layers frozen. For SCRUB, we follow the original paper's code implementation with $\alpha = 0.001$ and $\gamma = 0.99$. For Fisher forgetting, we use the code implementation provided in [11]. We set 200 training iterations for our framework.

C. Additional experimental results

C.1. Results on existing evaluation metrics

To provide a comprehensive view, we evaluate the unlearning algorithms with existing measurements, including Acc($\mathcal{D}_f$), Acc($\mathcal{D}_r$), and the MIA score. Table A, B, and C present the overall experimental results.

D. Discussion

D.1. Random sample unlearning

While most existing works have been evaluated under a random sample unlearning scenario, we did not explicitly address this setting. We argue that if the forget set is randomly drawn from the training set, these random forget samples do not provide meaningful additional information beyond the remaining samples, implying no information needs to be removed.

To clarify this, assume that both $\mathcal{D}_{\text{old}}$ and $\mathcal{D}_{\text{new}}$ follow the same distribution as $\mathcal{D}$. Given an old model trained on $\mathcal{D}_{\text{old}}$, we consider a incremental learning scenario involving $\mathcal{D}_{\text{new}}$. Since $\mathcal{D}_{\text{new}}$ follows the same distribution as $\mathcal{D}_{\text{old}}$, it behaves similar

Table C. Unlearning results on the Tiny-ImageNet dataset.

Method	DA	LP($\mathcal{D}_f$)	LP($\mathcal{D}_r$)	F1	NMI	Acc($\mathcal{D}_f$)	Acc($\mathcal{D}_r$)	MIA
Original	0.59	47.6	58.0	0.96	0.92	51.2	59.1	0.89
Retrained	0.73	24.0	58.0	0.19	0.09	0.0	59.1	0.14
FT	0.60	45.6	56.2	0.66	0.55	44.8	57.3	0.78
FT (classifier only)	0.61	47.6	**58.0**	0.66	0.55	0.0	41.2	0.18
NegGrad	**0.74**	29.6	55.1	0.22	0.12	0.0	51.8	0.30
Fisher	0.66	34.8	47.0	0.39	0.25	0.0	43.1	0.20
SCRUB	0.64	35.6	55.8	0.52	0.40	38.4	56.2	0.50
EU-k	0.77	12.8	19.1	0.11	0.04	0.0	11.1	0.42
CF-k	0.63	40.8	52.7	0.48	0.33	31.2	51.6	0.47
MUDA (Ours)	0.70	**26.0**	57.4	**0.21**	**0.10**	0.0	58.1	0.03

as $\mathcal{D}_{\text{old}}$ and the decision boundary would not change significantly during incremental learning. As there are no substantial changes caused by $\mathcal{D}_{\text{new}}$, unlearning $\mathcal{D}_{\text{new}}$ should rarely impact the model parameters.

Furthermore, if a user requests a random subset of samples to be forgotten, it is unclear whether the request refer to the specific selected samples or the entire (sub)class corresponding to those samples. Therefore, we limit the unlearning scenario to cases where the forget set contains meaningful semantics, such as a class, subclass, or group.

Note that the experimental results on defending backdoor attack in Section 5.3 implicitly address random sample unlearning, where the forget set is a random subset of training set. The difference is that in each application, the forget samples share a common semantic, *e.g.*, containing a black patch or label noise.

D.2. Exploiting forget set accuracy and MIA

The forget set accuracy and MIA can be easily exploited with trivial fine-tuning or post-processing techniques, which render them unreliable for adequately evaluating the unlearned model. Below we provide examples of such trivial methods that can easily exploit/circumvent each metric.

Forget set accuracy Achieving 0% accuracy on the forget set can be accomplished by simply setting the bias value of the corresponding class in the classifier to $-\infty$, ensuring that no samples are predicted for that class. This implies that merely matching the accuracy on the forget set does not necessarily indicate successful unlearning.

MIA success rate Since MIA leverages model outputs, such as confidence scores or entropy [32,36], to infer the presence of a sample in the training dataset, it depends heavily on the model overfitting to the dataset. The underlying assumption is that a model will produce more confident predictions for samples it has seen during training compared to unseen data. Hence, if models undergo uncertainty calibration via fine-tuning or post-processing techniques, the MIA may significantly overestimate the effectiveness of unlearning. To empirically support our claim, we fix the feature extractor of θ_o and only fine-tune its final linear classifier using a calibration loss [29]. We observe that simple post-processing calibration with minimal training significantly lowers the MIA score from 0.91 to 0.02 on CIFAR-10, despite no particular efforts to *unlearn*.

D.3. Limitations

Our paper, like previous studies, shares a common limitation: the lack of a theoretical guarantee regarding unlearning. Nonetheless, our framework introduces a unique approach by focusing on feature representation, which supplements previous research efforts by offering a novel and thorough analysis of machine unlearning. Additionally, our dimensional alignment loss requires some amount of retain samples, but we have shown that only a minimal number of these samples are necessary to attain effective unlearning performance.

D.4. Broader impact

By enabling the effective removal of data from machine learning models without requiring complete retraining, machine unlearning helps organizations comply with privacy laws such as GDPR and the CCPA, which mandate the right to be forgotten. This is crucial in situations where users withdraw their consent for data use or when data must be deleted for legal reasons.

Moreover, machine unlearning reduces the risks associated with data breaches, as it ensures sensitive information can be dynamically and reliably erased from models, thus limiting potential misuse. Additionally, this research can lead to more sustainable AI practices by reducing the computational and environmental costs associated with retraining models from scratch. This leads to more ethical AI systems by promoting transparency, user trust, and the responsible use of data.