ESC: Erasing Space Concept for Knowledge Deletion

Tae-Young Lee[1*] Sundong Park[2*] Minwoo Jeon[2*] Hyoseok Hwang[2†]
Gyeong-Moon Park[1†]

[1]Korea University, Seoul, Republic of Korea
[2]Kyung Hee University, Yongin, Republic of Korea

{tylee0415, gm-park}@korea.ac.kr
{sundong, alsdn2530, hyoseok}@khu.ac.kr

Abstract

As concerns regarding privacy in deep learning continue to grow, individuals are increasingly apprehensive about the potential exploitation of their personal knowledge in trained models. Despite several research efforts to address this, they often fail to consider the real-world demand from users for complete knowledge erasure. Furthermore, our investigation reveals that existing methods have a risk of leaking personal knowledge through embedding features. To address these issues, we introduce a novel concept of ***K****nowledge* ***D****eletion (****KD****), an advanced task that considers both concerns, and provides an appropriate metric, named* ***K****nowledge* ***R****etention score (****KR****), for assessing knowledge retention in feature space. To achieve this, we propose a novel training-free erasing approach named* ***E****rasing* ***S****pace* ***C****oncept (****ESC****), which restricts the important subspace for the forgetting knowledge by eliminating the relevant activations in the feature. In addition, we suggest* ***ESC*** *with* ***T****raining (****ESC-T****), which uses a learnable mask to better balance the trade-off between forgetting and preserving knowledge in KD. Our extensive experiments on various datasets and models demonstrate that our proposed methods achieve the fastest and state-of-the-art performance. Notably, our methods are applicable to diverse forgetting scenarios, such as facial domain setting, demonstrating the generalizability of our methods. The code is available at https://github.com/KU-VGI/ESC.*

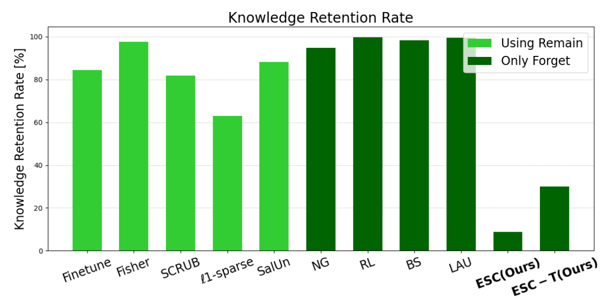

Figure 1. The recovery rate of each unlearning method using All-CNN in CIFAR-10. We conducted linear probing with each unlearned feature extractor and obtain the accuracy on the forgetting training data. Knowledge retention rate is calculated as acc_f^R/acc_f^O, where acc_f^R is the accuracy after linear probing, and acc_f^O is the accuracy of the original model. While existing unlearning methods still retain the forgetting knowledge, our methods effectively eliminate it from the feature representation.

1. Introduction

In response to growing concerns about data privacy, concerted efforts have been made to strengthen data protection regulations, such as General Data Protection Regulation [44]. These regulations pose a great challenge to modern deep learning models, particularly in satisfying the requirement to erase personal information upon an individual's request. Differential Privacy (DP) [8] provides a prominent definition for privacy concerns, and it aims to remove individual-specific information while preserving the utility of the data for a given task. Inspired by this concept, Machine Unlearning (MU) has recently emerged as an approach to addressing this challenge, enabling them to effectively forget selected knowledge while preserving others.

Previous MU approaches have largely followed privacy definitions of DP [1, 12, 23], aiming to approximate a retrained model [10] without considering the unlearned model's state. While this is widely adopted, it primarily focuses on data removal, overlooking practical user concerns in knowledge removal. In real-world scenarios, users initiate removal requests with two expectations: utility-focused

*Equal contribution.
†Corresponding authors.

and privacy-focused. Utility-focused requests seek to disable the model's ability to use the forgetting information, while privacy-focused requests aim to prevent any private information leakage, particularly against malicious attacks. For example, if the unlearned model can still utilize the forgetting knowledge, even if it results from the remaining knowledge, users cannot fully trust the removal process. To better meet real-world demands, we expand the scope of MU to address both utility and privacy effectively.

In addition, we point out that existing MU methods face another issue: knowledge retention in the feature. Our analysis of existing MU methods in Section 3.2 revealed that current MU approaches primarily focus on modifying the classification head while neglecting the feature extractor, which retains significant trained knowledge. This leads to incomplete removal, as shown in Figure 1. This figure indicates that retraining only a new head on top of a frozen unlearned feature extractor leads to significant performance recovery. This suggests a significant amount of feature-level knowledge still remains. This poses challenges for both utility and privacy. From a utility perspective, users can still leverage forgetting knowledge via embedding features, while from a privacy perspective, malicious users can directly use these features to extract private information.

Considering the aforementioned challenges, we first define a novel ***K**nowledge **D**eletion (**KD**)* problem setting. The proposed task aims to overcome the limitations of current MU methods by effectively erasing feature-level knowledge while simultaneously addressing user-specific requests. These requests include both utility-focused and privacy-focused needs, ultimately meeting real-world expectations for unlearning. Moreover, we propose a new benchmark, ***K**nowledge **R**etention score (**KR**)*, to evaluate residual feature-level knowledge (detailed in Section 3.2.3).

To tackle KD, we propose two rapid yet powerful KD methods: ***E**rasing **S**pace **C**oncept (**ESC**) and **ESC** with **T**raining (**ESC-T**)*. ESC effectively erases the corresponding knowledge by deactivating the feature space related to forgetting data *without any training process*. ESC-T, an enhanced version with lightweight optimization, uses a learnable mask to better manage the trade-off between remaining and forgetting knowledge in KD. Through comprehensive experiments on diverse datasets and models, we demonstrate that ESC and ESC-T can successfully remove the forgetting knowledge from the original model, even at the feature-level. Furthermore, our methods can be applied to various realistic KD scenarios, *e.g.*, facial domain, and show their effectiveness. To the best of our knowledge, our methods not only yield remarkable performances in diverse analyses but also achieve the fastest processing for KD.

Our main contributions can be summarized as follows:

- For the first time, we redefine the Machine Unlearning approach from a user-centric perspective and propose a new setting named *Knowledge Deletion (KD)*, which expands the scope of unlearning to feature-level knowledge. To assess this, we introduce a novel benchmark named *Knowledge Retention (KR)* score.
- To facilitate KD, we propose a novel training-free approach, *Erasing Space Concept (ESC)*, which can eliminate the space of concept in the embedding feature space for removing the forgetting knowledge.
- We also propose a training-based ESC, *ESC with Training (ESC-T)*, which conducts more fine-grained removal and alleviates the trade-off between removal and preservation of the knowledge.
- From the extensive experiments and analysis, we demonstrate that our proposed methods effectively erase the forgetting knowledge from the pre-trained networks in various datasets, models, and scenarios, and achieve the fastest and state-of-the-art performance in KD.

2. Related Work

Definition of MU. MU is defined as a process of effectively removing the knowledge or influence of specific data from a previously trained model. The goal of MU is to replicate a retrain-from-scratch model [10], inspired by the privacy definitions in DP [8]. Despite its widespread use, several studies suggest that comparing unlearned models against the retrain-from-scratch model is not an appropriate evaluation method for MU. Thudi *et al*. [42] theoretically demonstrate that similar model parameters can arise from different datasets. Goel *et al*. [11] also suggests that retrained models can vary depending on the hyperparameters. Therefore, existing evaluation has a risk for robust assessment of unlearned models. Similar to Kurmanji [23], we also consider practical requests, such as *"Remove my knowledge"*, considering both utility and privacy. We formulate this to provide an appropriate response from the user's perspective and extend it to the feature level.

MU Methods. MU methods can be broadly classified into two categories: exact unlearning and approximate unlearning. Exact unlearning methods [1, 14, 25, 27, 42, 45, 47, 48] aim to completely remove the influence of specific class or data from a model, as if it had never been used. However, these processes still require significant computational resources. In contrast, approximate unlearning eliminates the forgetting knowledge or data from the trained model while preserving the others. To better preserve utility, most approaches often use the entire training dataset [6, 9, 12, 14, 23, 26, 40], which is typically suited for instance-wise unlearning. However, in real-world scenarios, retaining sensitive data conflicts with privacy goals and legal constraints [30, 31, 44, 49], limiting their applicability. Recently, several methods [3, 18, 19, 35] have tried to conduct unlearning only with forgetting data, and we only consider this setting in this work. The most closely related work to ours is Kodge *et al*. [20], which uses Singular Value Decomposition for MU with whole datasets. In contrast, we consider the feature-level knowledge deletion from the user's request only with forgetting data for the first time.

3. Method

3.1. Preliminaries

Notations. $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ is a training dataset consisting of the input $x_i \in X$ and its class label $y_i \in Y$, where N denotes the dataset size. We denote $\mathcal{D}_f \subseteq \mathcal{D}$ as the forgetting training data, while $\mathcal{D}_r = \mathcal{D} \backslash \mathcal{D}_f$ as the remaining training data. Next, we denote the classification model as f_θ parameterized by θ. To illustrate our method, we further divide it into two parts: $f_\theta(x) = g_\phi \circ h_\psi(x)$, where h is a feature extractor, g is a classification head, and $\theta = \{\phi, \psi\}$.

Gold Standard of MU. Traditional MU aims to remove the influence of specific data or classes from a trained model based on the privacy definition of DP. For this reason, traditional MU typically considers the retrain-from-scratch model as the gold standard. As a result, to evaluate the efficiency of an unlearned model, most studies use privacy measures, such as accuracy or Membership Inference Attack (MIA) [2, 5, 33, 36–38, 43], by assessing the gap between the unlearned model and the retrained model [6, 9, 14, 23, 26]. While MIA, which determines whether specific data was used in model training, theoretically considers the retrained model as optimal, relying solely on it as the benchmark is not always suitable for other evaluation aspects, as discussed in Section 2. More importantly, to meet user demands like *knowledge removal*, it is essential to extend the data-centric approach to a broader context. To address this, we redefine the MU problem to Knowledge Deletion. In the following section, we introduce the concept of Knowledge Deletion alongside realistic user demands and propose a new optimal goal.

3.2. Problem Formulation

As already mentioned, KD reinterprets traditional MU from the user's perspective while simultaneously emphasizing the removal of knowledge at the feature level. This section provides a detailed explanation of this setting.

3.2.1. Extended Interpretation of MU

We redefine the interpretation of MU to accommodate real-world knowledge removal requests, focusing on the user's perspective. Users interact with the deployed model and evaluate its functionality through its outputs. For instance, if specific knowledge is removed but the model still utilizes it through the remaining knowledge, the user perceives no difference and does not consider the knowledge truly removed. Therefore, from the utility perspective, a good knowledge-deleted model should *minimize the utility of forgetting knowledge* while *maximizing the utility of remaining knowledge*.

The second aspect is privacy. If KD focuses solely on utility, the goal could be achieved by simply modifying nodes of the final layer (*i.e*., head). However, removal requests are not restricted to the output alone; they must also ensure the protection of privacy against arbitrary attacks such as MIA. In KD, MIA is evaluated based on the retrain-from-scratch model, similar to previous works, as the retrained model is theoretically optimal from the perspective of membership.

3.2.2. Knowledge Deletion.

In this section, we highlight a key limitation of existing Machine Unlearning (MU) methods: the lack of unlearning at the feature level. As illustrated in Figure 1, the knowledge retained in features indicates that the forgetting knowledge has not been fully removed, making it highly vulnerable to knowledge leakage through its outputs. Motivated by these findings, we expand the scope of MU to the removal of feature-level knowledge. Features capture the most representative form of learned knowledge in a trained model. For instance, in Self-Supervised Learning (SSL) [4, 17, 28], the quality of the extracted features is used to evaluate how effectively the model has learned from the given data, emphasizing their role as a core representation of learned knowledge. Consequently, KD aims to effectively remove information at the feature level to ensure comprehensive knowledge removal requests in the real world.

3.2.3. Knowledge Retention Score

In KD, our goal is to remove feature-level knowledge from the unlearned model, necessitating an appropriate knowledge measure in feature for evaluating the model after KD. However, there are no metrics for this. Most existing MU studies rely on metrics such as accuracy or MIA. In addition, some studies, like Chen *et al*. [3], analyzed model activations using Grad-CAM [34], or employed distance-based analyses, including activation distance [12, 13], Zero Retrain Forgetting (ZRF) [6], and JS-distance [6], as well as probability-based methods like distribution of the entropy [3, 18]. Despite these efforts, all of these approaches are based on the model's output, like logit, and do not capture the feature-level information. Although there has been an attempt to directly evaluate the model by measuring weight differences, such as layer-wise distance [40], this method does not effectively determine whether the model has truly forgotten, as mentioned in Section 2.

Therefore, inspired by the Self-Supervised Learning (SSL) approach, we propose a new benchmark, named ***K**nowledge **R**etention Score (**KR**)*, to adequately evaluate knowledge retention at the feature level. KR is specifically designed to measure the extent of knowledge retention in

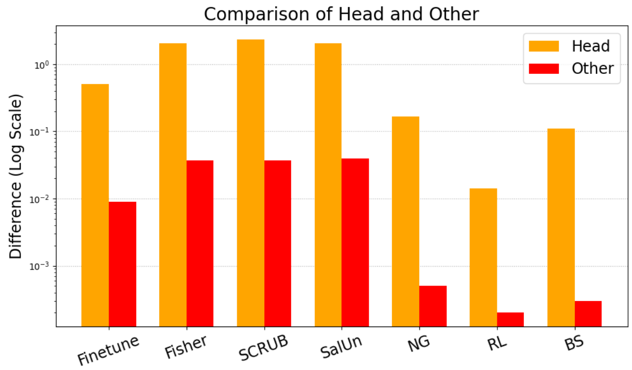

Figure 2. Using CIFAR-10 [22] with All-CNN [39], we measure weight difference between the original model and unlearned model. We calculate the difference as $\gamma \cdot \|\theta^l_{ori} - \theta^l_{ul}\| / \|\theta^l_{ori}$, where θ means weight of model and γ is a scaling factor that reflects how the change in a specific layer contributes to the overall change in the model. The change in the head is dominant over the change in the entire model for all MU methods. This implies that most models focus on the head when performing MU.

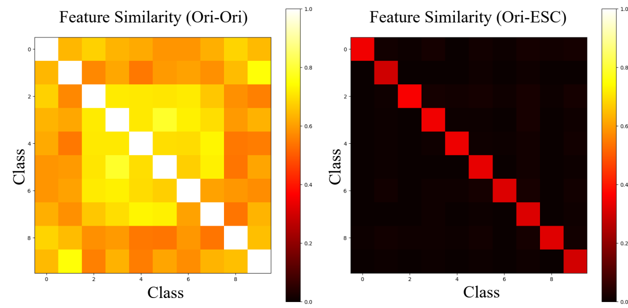

Figure 3. We calculate the cosine similarity between features for each class. (Left) Cosine similarity between the original features. (Right) Cosine similarity between the original features and the ESC features, where the principal component has been removed from the original features.

the feature extractor. KR performs linear probing [21] using feature extractor after KD, and then measures the accuracy for both forgetting and remaining data. We interpret this similarly to utility, as mentioned above.

3.3. Erasing Space Concept

Why existing works cannot unlearn at the feature level? Most existing MU methods develop unlearning methods based on Negative Gradient or Random Label approaches, and perform end-to-end training using these forgetting losses. These approaches rely on a logit-based loss function, which makes it easy for the model to find a shortcut solution by focusing on the classification head. In Figure 2, we experimentally analyze this by measuring the weight difference between the unlearned model and the original model. The results indicate that existing methods are biased toward perturbing at the head. Consequently, most MU methods fail to effectively unlearn the encoder, where most of the learned knowledge resides. Therefore, to achieve a more fundamental removal of knowledge, it is essential to directly erase feature-level knowledge.

Motivation. Our ultimate goal is to make it impossible for both users and malicious actors to extract any forgetting knowledge from the model by directly removing the knowledge in the feature space while preserving the remaining knowledge. To achieve this, we need to decompose the forgetting knowledge from others and remove the model's capability, *i.e.*, activation. Inspired by Gu *et al.* [15], we can expect that we can decompose and deactivate the feature by using only partial principal directions given by Singular Value Decomposition (SVD). In Figure 3, we conducted a toy experiment. We calculate the cosine similarity of features between each class. On the left side, the results show that the original features are highly entangled between classes, e.g., the minimum value of similarity is greater than 0.5. On the right side, the ESC features by our method are clearly disentangled from the features of other classes (details about the ESC features are provided in the ESC section). Furthermore, they are no longer aligned with the original features of the same class, which contain the learned knowledge, with all similarity values falling below 0.35. Building on this insight, we introduce **E**rasing **S**pace **C**oncept named **ESC**, which can remove the forgetting knowledge by deactivating in the feature space.

Erasing Space Concept. We start with feature decomposition on the feature space to extract principal components. We pass the forgetting data through the feature extractor of the original model (h_ψ) to obtain a feature matrix $\mathbf{Z}_f$ (see Figure 4 (a)), which is then decomposed using SVD as follows:

$$\mathbf{Z}_f = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^\top, \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ represent singular directions in the feature dimension and data length dimension, respectively, and $\mathbf{\Sigma}$ is the diagonal matrix containing singular values. From Equation 1, we can get principal directions from the left singular vectors, *i.e.*, $\mathbf{U} = [u_1, u_2, \cdots, u_d]$, where $u_i \in \mathbb{R}^d$ is sorted based on the singular value, σ_i. Based on our toy experiment, with the frozen backbone, our proposed ESC projects features onto the subspace spanned by the pruned principal directions. To obtain the pruned subspace, we eliminate the first $p\%$ of principal directions from $\mathbf{U}$, as shown in Figure 4 (b), where p is a pruning hyperparameter. Higher p values correspond to larger knowledge deletion, and when $p = 0$, it indicates that the model remains identical to the original. We can get the pruned principal directions $\mathbf{U}_P$ as:

$$\mathbf{U}_P = \mathbf{U}[k :], \tag{2}$$

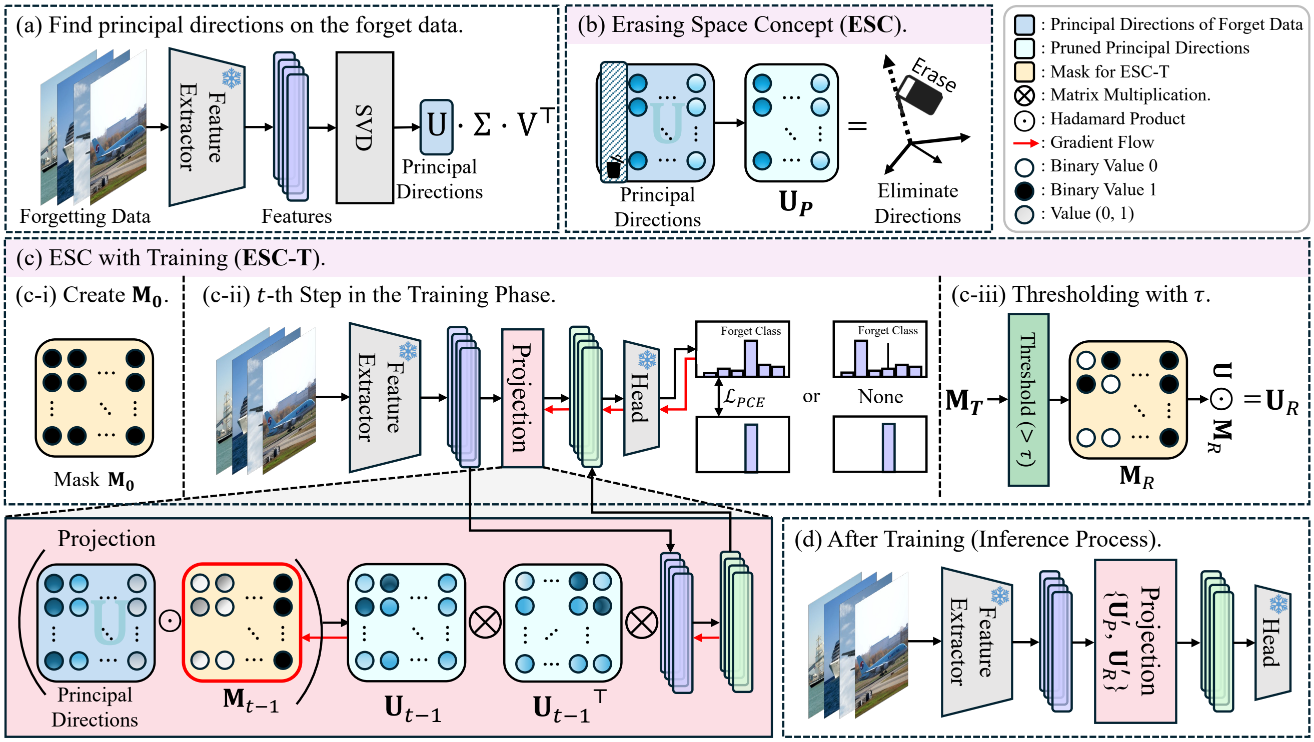

Figure 4. An overview of our methods. (a) We start with extracting the principal directions ($\mathbf{U}$) of embedding features from the forgetting data using SVD. (b) ESC erases crucial directions from $\mathbf{U}$, and we can get the pruned principal directions $\mathbf{U}_P$. (c) ESC-T enhances ESC by incorporating our forgetting loss, $\mathcal{L}_{PCE}$, refining the erasure process to eliminate only important elements rather than entire directions. ESC-T yields the refined principal directions $\mathbf{U}_R$, which improve the trade-off forgetting and preservation. (d) During the inference phase, we project the extracted features onto the subspace formed by $\mathbf{U}_P$ or $\mathbf{U}_R$ from each method.

where k is the number of pruned directions, determined by p as $k = \frac{d}{100} \cdot p$, and it can vary depending on the embedding dimension. Using this selectively pruned principal directions, our unlearned feature extractor and the model by ESC are defined as follows:

$$\begin{aligned} h_{\psi_P}(x) &= \mathbf{U}_P \cdot \mathbf{U}_P^\top \cdot h_\psi(x), \\ f_{ESC}(x) &= g_\phi \circ h_{\psi_P}(x). \end{aligned} \tag{3}$$

From the extensive experiments, we demonstrate that even without any training process, this simple approach successfully deactivates the knowledge of forgetting data in both the output and feature spaces, *i.e.*, *KD* is fulfilled.

3.4. Erasing Space Concept with Training

While ESC effectively deactivates the key features related to the forgetting data, removing entire vectors along the principal directions limits the flexibility of the removal process. Hard thresholding, a naïve approach, can lead to over-forgetting; therefore, it is necessary to restrict the intensity of the removal selectively. To this end, we propose **ESC** with **T**raining (**ESC-T**), which enhances ESC through an additional lightweight training process that selectively blocks only the key elements in each principal direction related to the forgetting data.

In ESC-T, we do not use the pruned principal directions in ESC, which eliminates the entire vectors. As shown in Figure 4 (c-i), we start ESC-T with creating a learnable refining mask $\mathbf{M}_0$, initialized with $\mathbf{1}$, to obtain the optimized principal directions $\mathbf{U}_R$.

We then optimize the $\mathbf{M}_0$ by using our proposed forgetting loss (Figure 4 (c-ii)). Following the work of [12, 19, 41], we design a *Penalized Cross-Entropy Loss* for training the mask, which penalizes the mask when the model makes correct predictions. This loss is defined as follows:

$$\mathcal{L}_{PCE} = \begin{cases} -\mathcal{L}_{CE}(f_\theta(x), y), & \text{if } \hat{y} = y, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\hat{y}$ is the predicted class from the model f_θ. We update only the mask to minimize $\mathcal{L}_{PCE}$ until either all predictions of the unlearned model are incorrect for the given forgetting data, or a specified number of steps has been reached. Additionally, at each training step t, we constrain the values of all elements in the mask $\mathbf{M}_t$ between 0 and 1 to maintain stable feature space. When the learning process is done, we apply a threshold τ to modify the mask $\mathbf{M}_T$ to be binary, *i.e.*, containing only 0s and 1s, resulting in the final binary

Method	D_r	CIFAR-10 $D_f(\downarrow)$	CIFAR-10 $D_r(\uparrow)$	CIFAR-10 $D_{ft}(\downarrow)$	CIFAR-10 $D_{rt}(\uparrow)$	CIFAR-10 $HM(\uparrow)$	CIFAR-10 $HM_{\mathrm{t}}(\uparrow)$	CIFAR-10 MIA	CIFAR-10$-KR$ $D_f(\downarrow)$	CIFAR-10$-KR$ $D_r(\uparrow)$	CIFAR-10$-KR$ $D_{ft}(\downarrow)$	CIFAR-10$-KR$ $D_{rt}(\uparrow)$	CIFAR-10$-KR$ $HM(\uparrow)$	CIFAR-10$-KR$ $HM_{\mathrm{t}}(\uparrow)$
Original	-	98.42	98.29	85.90	86.57	3.11	24.25	57.68 ± 0.67	98.40	97.81	86.40	85.96	3.15	23.48
Retrain		0.00	96.93	0.00	86.02	98.44	92.48	50.06 ± 0.43	41.28	96.18	44.40	84.38	72.92	67.03
Finetune [46]		6.89 ± 1.71	98.90 ± 0.14	5.53 ± 1.27	85.94 ± 0.17	95.91 ± 0.97	90.00 ± 0.62	<u>53.13 ± 0.67</u>	82.99 ± 2.03	98.60 ± 0.05	69.57 ± 3.33	85.35 ± 0.14	28.97 ± 2.98	44.80 ± 3.68
Fisher [12]		0.74 ± 0.65	96.16 ± 0.25	0.63 ± 0.51	81.22 ± 0.79	97.69 ± 0.18	89.38 ± 0.28	54.60 ± 0.95	96.00 ± 0.05	95.76 ± 1.16	83.33 ± 0.42	81.81 ± 0.10	7.68 ± 0.10	27.69 ± 0.57
SCRUB [23]	✓	10.39 ± 3.10	97.15 ± 0.45	4.50 ± 1.99	83.33 ± 0.38	93.23 ± 1.89	89.00 ± 1.09	55.50 ± 1.46	80.41 ± 4.91	96.62 ± 0.33	65.53 ± 1.99	81.85 ± 0.29	32.57 ± 6.96	48.51 ± 2.03
ℓ_1-sparse [26]		00.00 ± 0.00	89.95 ± 0.15	00.00 ± 0.00	84.81 ± 2.00	94.71 ± 0.30	91.78 ± 3.92	55.76 ± 1.25	62.03 ± 2.09	88.57 ± 0.49	61.12 ± 0.97	85.29 ± 0.61	53.15 ± 2.07	53.42 ± 0.80
SalUn [9]		00.00 ± 0.00	98.86 ± 0.34	00.00 ± 0.00	86.80 ± 0.70	**99.43 ± 0.69**	**92.94 ± 1.38**	56.42 ± 0.46	86.83 ± 3.06	98.12 ± 0.28	83.81 ± 0.33	91.22 ± 2.09	23.22 ± 4.72	27.49 ± 0.38
NG [41]		2.46 ± 0.30	90.27 ± 0.16	2.17 ± 0.15	75.83 ± 0.05	93.76 ± 0.06	85.44 ± 0.03	53.80 ± 1.68	93.21 ± 1.37	99.14 ± 0.05	69.23 ± 1.14	83.02 ± 0.22	12.69 ± 2.42	44.89 ± 1.24
RL [12]		13.25 ± 0.44	98.21 ± 0.04	7.93 ± 0.46	80.43 ± 0.03	92.12 ± 0.25	85.85 ± 0.19	57.10 ± 1.73	98.05 ± 0.16	99.70 ± 0.03	72.97 ± 0.72	82.52 ± 0.42	3.83 ± 0.31	40.72 ± 0.86
BS [3]		10.98 ± 0.64	95.89 ± 1.60	13.43 ± 2.71	84.41 ± 1.12	92.32 ± 0.42	85.43 ± 0.73	57.26 ± 0.38	96.71 ± 1.02	96.93 ± 0.44	83.20 ± 1.84	85.40 ± 0.31	6.34 ± 1.88	28.03 ± 2.53
LAU [19]	✗	0.15 ± 0.01	92.84 ± 0.42	0.07 ± 0.05	82.28 ± 0.43	96.22 ± 0.22	90.25 ± 0.28	54.92 ± 0.49	97.86 ± 0.44	97.39 ± 0.39	85.57 ± 0.62	85.71 ± 0.30	4.19 ± 0.85	24.70 ± 0.91
ESC(Ours)		9.46 ± 0.06	96.52 ± 0.01	10.73 ± 0.12	85.59 ± 0.01	93.43 ± 0.03	87.39 ± 0.06	**53.02 ± 0.53**	8.53 ± 0.58	97.45 ± 0.24	8.23 ± 0.55	86.56 ± 0.22	**94.36 ± 0.27**	**89.08 ± 0.16**
ESC-T(Ours)		0.00 ± 0.00	97.23 ± 0.19	0.00 ± 0.00	86.08 ± 0.27	<u>98.60 ± 0.10</u>	<u>92.51 ± 0.16</u>	56.72 ± 0.24	29.47 ± 2.55	97.25 ± 0.47	25.07 ± 2.78	86.12 ± 0.40	<u>81.74 ± 1.56</u>	<u>80.12 ± 1.44</u>

Method	D_r	CIFAR-100 $D_f(\downarrow)$	CIFAR-100 $D_r(\uparrow)$	CIFAR-100 $D_{ft}(\downarrow)$	CIFAR-100 $D_{rt}(\uparrow)$	CIFAR-100 $HM(\uparrow)$	CIFAR-100 $HM_{\mathrm{t}}(\uparrow)$	CIFAR-100 MIA	CIFAR-100$-KR$ $D_f(\downarrow)$	CIFAR-100$-KR$ $D_r(\uparrow)$	CIFAR-100$-KR$ $D_{ft}(\downarrow)$	CIFAR-100$-KR$ $D_{rt}(\uparrow)$	CIFAR-100$-KR$ $HM(\uparrow)$	CIFAR-100$-KR$ $HM_{\mathrm{t}}(\uparrow)$
Original		96.66	95.71	78.50	79.70	6.45	33.86	49.23 ± 1.42	83.06	83.27	69.40	70.54	28.15	42.68
Retrain		0.00	88.78	00.00	78.87	94.06	88.19	63.17 ± 2.87	54.66	82.03	87.80	73.10	58.40	53.51
Finetune [46]		0.00 ± 0.00	89.98 ± 0.42	0.00 ± 0.00	74.18 ± 0.25	94.72 ± 0.87	85.18 ± 0.50	69.12 ± 0.97	40.68 ± 1.60	78.21 ± 0.27	38.25 ± 0.20	72.88 ± 11.16	67.47 ± 1.07	66.85 ± 4.38
Fisher [12]		0.00 ± 0.00	67.46 ± 0.39	00.00 ± 0.00	54.59 ± 0.59	80.57 ± 0.28	70.63 ± 0.49	55.28 ± 1.40	81.46 ± 2.57	84.05 ± 1.63	60.96 ± 1.60	70.24 ± 1.35	30.38 ± 3.57	50.18 ± 1.67
SCRUB [23]	✓	19.15 ± 0.05	86.98 ± 1.69	14.14 ± 0.93	75.82 ± 0.62	83.80 ± 0.79	80.53 ± 0.58	56.33 ± 0.67	74.85 ± 0.46	79.25 ± 0.22	55.27 ± 0.21	65.57 ± 0.46	38.19 ± 0.51	53.18 ± 0.19
ℓ_1-sparse [26]		0.00 ± 0.00	57.34 ± 0.14	0.00 ± 0.00	50.59 ± 0.41	72.89 ± 0.28	67.79 ± 0.81	56.42 ± 0.56	60.56 ± 0.03	65.78 ± 0.32	54.63 ± 0.09	58.53 ± 0.43	49.31 ± 0.63	51.11 ± 0.86
SalUn [9]		0.76 ± 0.00	96.64 ± 0.01	2.33 ± 0.02	67.73 ± 0.88	**97.93 ± 0.00**	79.99 ± 0.61	56.12 ± 0.72	86.68 ± 1.95	96.61 ± 0.32	66.86 ± 1.77	71.86 ± 0.89	23.41 ± 2.98	45.36 ± 1.80
NG [41]		13.24 ± 0.61	84.50 ± 0.32	10.87 ± 0.42	71.64 ± 0.08	85.62 ± 0.64	79.43 ± 0.15	54.97 ± 0.59	52.39 ± 0.33	70.68 ± 0.15	44.24 ± 0.10	62.19 ± 0.21	56.90 ± 0.19	58.80 ± 0.12
RL [12]		27.45 ± 0.53	84.74 ± 0.43	19.33 ± 0.35	70.38 ± 0.21	78.17 ± 0.24	75.17 ± 0.10	53.23 ± 1.46	46.48 ± 0.31	70.76 ± 0.22	38.88 ± 0.50	61.48 ± 0.27	60.95 ± 0.14	61.30 ± 0.38
BS [3]		9.93 ± 0.79	61.16 ± 1.14	7.23 ± 0.74	50.53 ± 1.03	72.85 ± 0.56	65.42 ± 0.69	54.30 ± 1.97	55.98 ± 1.56	72.63 ± 0.61	46.63 ± 1.60	62.42 ± 0.54	54.81 ± 1.32	57.54 ± 0.99
LAU [19]	✗	0.00 ± 0.00	44.14 ± 6.55	0.00 ± 0.00	36.84 ± 5.10	61.25 ± 6.47	53.84 ± 5.56	**60.53 ± 0.85**	81.81 ± 0.58	83.14 ± 0.12	68.43 ± 1.80	70.68 ± 0.62	29.85 ± 0.78	43.64 ± 1.71
ESC(Ours)		0.80 ± 0.14	89.46 ± 0.19	0.50 ± 0.1	74.21 ± 0.17	94.08 ± 0.17	85.01 ± 0.15	58.37 ± 0.38	0.27 ± 0.01	76.55 ± 0.45	0.59 ± 0.02	65.74 ± 0.36	**86.62 ± 0.28**	<u>79.14 ± 0.21</u>
ESC-T(Ours)		0.00 ± 0.00	91.20 ± 0.55	0.09 ± 0.01	76.73 ± 0.84	<u>95.40 ± 0.30</u>	**86.80 ± 0.53**	<u>58.93 ± 1.27</u>	8.59 ± 0.59	80.46 ± 0.49	8.05 ± 0.09	71.86 ± 0.89	<u>85.59 ± 0.24</u>	**80.67 ± 0.79**

Method	D_r	Tiny-ImageNet $D_f(\downarrow)$	Tiny-ImageNet $D_r(\uparrow)$	Tiny-ImageNet $D_{ft}(\downarrow)$	Tiny-ImageNet $D_{rt}(\uparrow)$	Tiny-ImageNet $HM(\uparrow)$	Tiny-ImageNet $HM_{\mathrm{t}}(\uparrow)$	Tiny-ImageNet MIA	Tiny-ImageNet$-KR$ $D_f(\downarrow)$	Tiny-ImageNet$-KR$ $D_r(\uparrow)$	Tiny-ImageNet$-KR$ $D_{ft}(\downarrow)$	Tiny-ImageNet$-KR$ $D_{rt}(\uparrow)$	Tiny-ImageNet$-KR$ $HM(\uparrow)$	Tiny-ImageNet$-KR$ $HM_{\mathrm{t}}(\uparrow)$
Original		96.72	96.47	90.30	90.24	6.34	17.52	55.80 ± 0.62	95.55	94.80	90.50	88.57	8.50	17.16
Retrain		0.00	96.93	0.00	86.02	98.44	92.48	51.33 ± 0.59	66.56	92.71	63.30	82.79	49.15	50.86
Finetune [46]	✓	20.58 ± 0.61	96.95 ± 0.06	18.77 ± 0.61	84.55 ± 0.58	87.31 ± 0.11	82.86 ± 1.15	59.40 ± 2.27	64.61 ± 0.83	93.40 ± 0.89	62.73 ± 0.38	81.84 ± 0.57	51.33 ± 0.89	51.21 ± 0.43
NG [41]		0.00 ± 0.00	0.56 ± 0.00	00.00 ± 0.00	0.56 ± 0.00	1.11 ± 0.00	1.11 ± 0.00	48.57 ± 1.86	1.25 ± 0.83	0.96 ± 0.05	1.37 ± 0.70	0.89 ± 0.11	1.90 ± 0.10	1.76 ± 0.21
RL [12]		5.13 ± 0.84	5.76 ± 0.52	5.57 ± 0.76	5.71 ± 0.61	10.86 ± 0.93	10.76 ± 1.08	<u>52.90 ± 1.14</u>	8.44 ± 1.55	7.49 ± 1.40	8.23 ± 0.45	7.23 ± 1.73	13.85 ± 2.43	13.40 ± 3.02
BS [3]		11.25 ± 0.41	14.52 ± 1.21	10.03 ± 0.23	14.54 ± 1.09	24.96 ± 1.78	25.03 ± 1.61	60.07 ± 0.47	40.85 ± 2.63	39.67 ± 1.90	39.63 ± 2.83	39.37 ± 1.36	47.49 ± 1.30	47.66 ± 1.63
LAU [19]	✗	0.00 ± 0.00	95.62 ± 0.01	0.00 ± 0.00	89.76 ± 0.03	97.76 ± 0.00	94.60 ± 0.02	49.03 ± 1.84	95.52 ± 0.20	94.77 ± 0.05	89.33 ± 0.85	88.67 ± 0.04	8.56 ± 0.36	19.04 ± 1.36
ESC(Ours)		0.10 ± 0.02	96.36 ± 0.05	0.03 ± 0.06	90.57 ± 0.05	<u>98.10 ± 0.02</u>	<u>95.03 ± 0.05</u>	**50.20 ± 3.06**	0.27 ± 0.11	94.64 ± 0.10	0.20 ± 0.17	88.88 ± 0.02	**97.19 ± 0.06**	**94.02 ± 0.04**
ESC-T(Ours)		0.00 ± 0.00	96.49 ± 0.23	0.00 ± 0.00	90.57 ± 0.12	**98.21 ± 0.12**	**95.05 ± 0.07**	56.73 ± 1.55	1.36 ± 0.07	94.53 ± 0.28	0.87 ± 0.04	88.81 ± 0.16	<u>96.54 ± 0.17</u>	<u>93.69 ± 0.08</u>

Table 1. Accuracy, MIA, and KR performance in the KD setting are evaluated using CIFAR-10 with AllCNN, CIFAR-100 with ResNet-18, and Tiny-ImageNet with ViT. D_r indicates which methods use the remaining data during the unlearning process. The table presents the mean and standard deviation (mean ± std) across three trials. The best value is highlighted in bold, while the second-best is underlined. Additionally, the best result among methods that do not use D_r is colored in blue.

mask $\mathbf{M}_R$ (Figure 4 (c-iii)). This last step can completely remove the influence of important elements for forgetting knowledge while preserving unimportant elements. Based on $\mathbf{M}_R$, we modify Equation 2 as follows:

$$\mathbf{U}_R = \mathbf{U} \odot \mathbf{M}_R, \tag{5}$$

where $\mathbf{U}_R$ is the trained principal directions and $\odot$ is the Hadamard product. We then finally get ESC-T as follows:

$$\begin{aligned} h_{\psi_R}(x) &= \mathbf{U}_R \cdot \mathbf{U}_R^\top \cdot h_\psi(x), \\ f_{ESC-T}(x) &= g_\phi \circ h_{\psi_R}(x). \end{aligned} \tag{6}$$

We provide a detailed description of Algorithm 1 for ESC-T in the supplementary A. Through this simple yet efficient learning step, ESC-T enhances the removal intensity without degrading the model's performance. In Figure 4 (d), we describe the inference process of our methods.

4. Experiments

4.1. Experimental Setup

Datasets, Models, and Setting. We conducted unlearning on the CIFAR-10 [22], CIFAR-100 [22], Tiny-ImageNet [24], and Lacuna-10 [12, 23] datasets. In the case of KD, we removed 10% of the total number of classes, while for random data forgetting, we unlearned 10% of the data. Experiments were conducted on various models, including All-CNN [39], ResNet-18 [16] pre-trained on ImageNet-1K [32], and Vision Transformer (ViT) [7] pre-trained on ImageNet-21K [29]. We conducted all experiments using a single NVIDIA A5000 GPU. Further details on datasets and setups are available in the supplementary B.

Baselines. We compared our methods with various approximate unlearning methods. **Original** is a complete model trained using $\mathcal{D}$ from scratch or pre-trained weights.

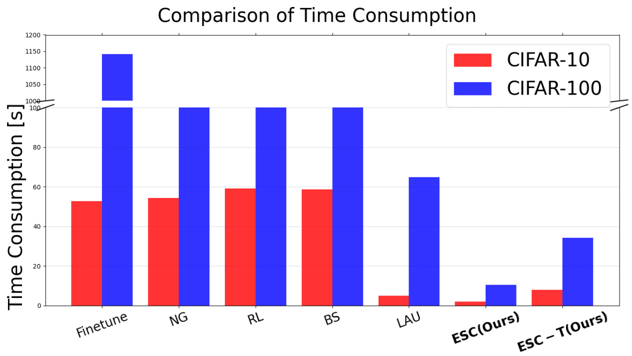

Figure 5. Comparison of time consumption. Our approaches are the most efficient method compared with others. Additional results reported in Supplementary E.

Retrain refers to a model trained from scratch using only $\mathcal{D}_r$. However, for ResNet-18 and ViT, we initialize from a pre-trained model, so it is not strictly the same with retrain-from-scratch. We focus on real-world scenarios and primarily compared our approach with existing MU methods, including **NG** [41], **RL** [12], **BS** [3], and **LAU** [19], all of which use only the forgetting data. For more extensive experiments, we also included **Finetune** [46], **Fisher** [12], **SCRUB** [23], ℓ_1**-sparse** [26], and **SalUn** [9], although all of these methods require the use of the entire remaining data.

Evaluation Metrics. In our experiments, we divided the datasets into remaining training data $\mathcal{D}_r$, forgetting training data $\mathcal{D}_f$, remaining testing data $\mathcal{D}_{rt}$, and forgetting testing data $\mathcal{D}_{ft}$. We calculated accuracy for each dataset segment both in post-unlearning and the proposed KR. Furthermore, similar to [19], we utilized the Harmonic Mean (HM) for comprehensive evaluation. To assess the privacy guarantee, we also constructed MIA, widely used in MU.

4.2. Experimental Results

We present comprehensive experimental results of KD across multiple datasets and models, compared with various MU methods, even though some of these methods essentially required the entire datasets, as shown in Table 1. The results indicate that our methods achieved the best utility-focused performance in all cases compared to methods that did not use D_r. Even when compared to the method using D_r, our approach showed comparable results. The results also demonstrated that ESC-T achieved a better balance between forgetting and preservation compared to ESC by selectively removing elements in the principal directions through a simple training process. In privacy-focused MIA, our approach also demonstrated performance similar to that of other MU methods overall. We additionally reported the ZRF [6] metric, an additional privacy-related evaluation, in the supplementary E.

In the KR setting, all baseline methods, even those using D_r, suffered from substantial knowledge retention in the features. ℓ_1-sparse showed comparable results among the MU methods across all experimental settings, as it directly prunes model parameters for unlearning. However, as demonstrated in our analysis in Figure 3, the original model features are highly entangled across different classes, making pruning insufficient to effectively eliminate the forgetting knowledge. In contrast, our methods significantly outperformed other baselines, and we effectively eliminated the forgetting knowledge in the feature space while preserving the remaining knowledge.

In Figure 5, we present the time consumption during the KD. ESC achieved the smallest time consumption because it required only a single forward process. ESC-T was also efficient, as it involved training only the small parameters of the learnable mask with a high learning rate. Specifically, ESC is 9.17 times faster than the previously fastest method, LAU, across all settings. We highlight again that our methods only use forgetting data, so that it is a more practical solution in the real world.

p (%)	CIFAR-100		CIFAR-100$-KR$	
	$HM(\uparrow)$	$HM_t(\uparrow)$	$HM(\uparrow)$	$HM_t(\uparrow)$
1.7	**99.01**	**96.45**	**98.30**	**95.70**
5	98.58	96.00	97.30	94.69
10	98.30	95.53	95.19	92.61
50	95.75	92.99	76.66	74.05

Table 2. Impact of pruning hyperparameter p in ESC. The results show that ESC is robust to pruning hyperparameter p.

Method	$\mathcal{L}_{PCE}$	Clip	τ	CIFAR-100		CIFAR-100$-KR$	
				$HM(\uparrow)$	$HM_t(\uparrow)$	$HM(\uparrow)$	$HM_t(\uparrow)$
ESC				99.01	96.45	98.30	95.70
	✓			99.05	96.41	92.03	89.59
	✓	✓		99.04	96.46	94.70	91.59
	✓		✓	90.33	89.60	97.48	94.22
ESC-T	✓	✓	✓	**99.05**	**96.47**	**97.56**	**95.11**

Table 3. Ablation table to figure out the effectiveness of our training method in ESC-T. The results clearly show the effectiveness of each components in ESC-T.

4.3. Ablation Study

p **Ablation.** We conducted an ablation study on the pruning hyperparameter p in ESC. As p approaches 0%, more knowledge about the forgetting data is eliminated, which can affect the remaining accuracy. In contrast, as the p approaches 100%, it eliminates more knowledge about the forgetting data and can influence the remaining accuracy. As shown in Table 2, ESC performed robustly across different values of p, and additional results are provided in the supplementary material.

Ablation Study for ESC-T. To assess the effectiveness of each component of ESC-T, we present the ablation study

τ	CIFAR-100		CIFAR-100$-KR$	
	$HM(\uparrow)$	$HM_t(\uparrow)$	$HM(\uparrow)$	$HM_t(\uparrow)$
0.6	98.60	95.95	<u>97.38</u>	**95.13**
0.7	<u>99.05</u>	<u>96.43</u>	**97.49**	**95.13**
0.8	**99.06**	**96.49**	96.86	94.09
0.9	98.26	95.44	58.27	58.79

Table 4. Changes in unlearning performance with ViT based on hyper-parameter τ of ESC-T.

Method	D_r	CIFAR-10 Random Data Forgetting (10%)					
		$D_f(\downarrow)$	$D_r(\uparrow)$	$D_t(\downarrow)$	MIA	Avg. Gap	RTE(s)
Retrain		95.58	100.00	95.23	74.64	-	2681.72
Finetune [46]	✗	99.85	100.00	95.08	87.73	4.38	132.02
RL [12]	✗	94.52	99.97	93.66	27.99	12.33	2.57
SCRUB [23]	✗	99.99	100.00	95.41	86.41	4.09	160.39
BadT [6]	✗	100.00	100.00	95.27	60.33	4.69	182.78
SalUn [9]	✗	100.00	99.99	95.18	63.61	3.88	150.51
NG [41]	✓	96.57	96.59	89.67	79.26	3.65	26.79
ESC(Ours)	✓	100.00	100.00	95.07	73.43	**1.45**	2.24
ESC-T(Ours)	✓	99.86	97.78	92.73	76.74	<u>2.77</u>	29.47

Table 5. Accuracy, MIA, and Run Time Efficiency (RTE) performance using ResNet-18 on CIFAR-10. The table presents the average of across three trials. The best value is in bold, the second-best is underlined, and the best without D_r is in blue.

results in Table 3. Mask learning using $\mathcal{L}_{PCE}$ enhanced the KD performance; however, this improvement alone was insufficient. Introducing clipping to restrict mask values within the range of 0 to 1 led to an improvement in KR performance. Applying a threshold effectively prevented recovery but resulted in a degradation of accuracy. Notably, when all components were used together, ESC-T, the overall performance was the best. We additionally conducted analysis about the clipping method in the supplementary D.

Sensitivity Analysis of Threshold $\mathcal{T}$. In Table 4, we present the robustness of our approach to changes in the threshold parameter τ. These results demonstrated that our proposed method is not sensitive to variations in τ. In our experiments, we used a value of τ between 0.7 and 0.8, which was empirically found.

4.4. Additional Experiments

We additionally conducted a comprehensive analysis to evaluate generalizability. Due to space limitations, we only report results for random data forgetting and partial visualizations. In the supplementary E, we provide further scenarios, such as incremental KD, along with additional combinations of datasets and models.

Random Data Forgetting. To demonstrate the generalizability of our methods, we also conducted a random data forgetting scenario, aimed at removing the influence of specific data. As shown in Table 5, our methods demonstrated superior performance compared to other approaches (see the average performance gap). Notably, our methods achieved this using only the forgetting data, without compromising the overall performance of the unlearned model.

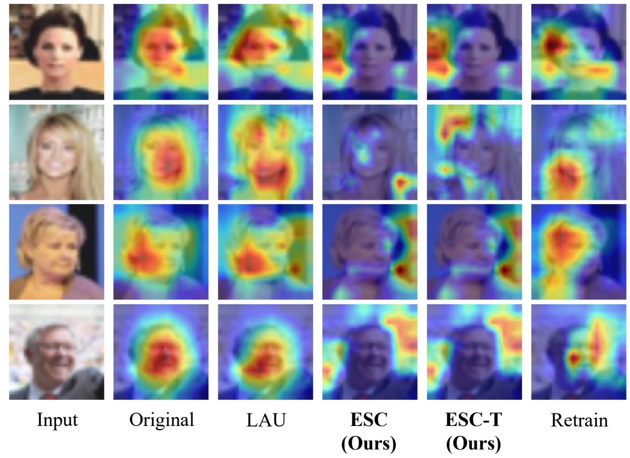

Figure 6. The activation map of each method using Grad-CAM [34]. These maps were visualized for the All-CNN model on the lacuna-10 dataset.

Visualization. To gain a clearer understanding of KD, we visualized the activation maps using Grad-CAM [34] on the facial domain dataset, Lacuna-10 [12, 23]. We trained an All-CNN model to classify identity and compared it with LAU, which is the most comparable method that also uses only the forgetting data. As shown in Figure 6, the original model strongly focused on the target object in the image, particularly emphasizing facial areas to capture identity-related features. Notably, with ESC and ESC-T, we observed a distinct shift in activation from the subjects to the background, significantly more so compared to other baseline methods. We also presented the results for other classical data, like car, along with additional t-SNE results, in the supplementary F.

5. Conclusion

In this paper, we advanced existing MU techniques based on real-world knowledge removal requests, thereby redefining the interpretation of MU. More importantly, we identified a critical issue of knowledge retention in features, and ultimately proposed *Knowledge Deletion*. In response to this, we introduced the *Knowledge Retention Score*, an evaluation metric that measures the intensity of retained knowledge at the feature level after KD. This metric highlighted the limitations of existing methods and laid the foundation for advancing the KD field. To tackle this, we proposed ESC and ESC-T, which effectively remove the knowledge specified for forgetting in KD. Our methods, despite their simplicity, demonstrate significant performance improvements over existing works, achieving superior results in KD with minimal unlearning time. We hope our work inspires new insights and advances in the MU research community.

6. Acknowledgement

This work was supported by Korea Planning & Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea government (MOTIE) (RS-2024-00444344), and in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University)).

References

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021. 1, 2

[2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022. 3

[3] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7766–7775, 2023. 2, 3, 6, 7

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[5] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021. 3

[6] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7210–7217, 2023. 2, 3, 7, 8

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6

[8] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. 1, 2

[9] 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*, 2024. 2, 3, 6, 7, 8

[10] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019. 1, 2

[11] Shashwat Goel, Ameya Prabhu, and Ponnurangam Kumaraguru. Evaluating inexact unlearning requires revisiting forgetting. *CoRR abs/2201.06640*, 2022. 2

[12] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020. 1, 2, 3, 5, 6, 7, 8

[13] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 792–801, 2021. 3

[14] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11516–11524, 2021. 2, 3

[15] Qiao Gu, Dongsub Shim, and Florian Shkurti. Preserving linear separability in continual learning by backward feature projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24286–24295, 2023. 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3

[18] Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4819–4828, 2024. 2, 3

[19] Hyunjune Kim, Sangyong Lee, and Simon S Woo. Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation. In *The 38th Annual AAAI Conference on Artificial Intelligence*. AAAI, 2024. 2, 5, 6, 7

[20] Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. Deep unlearning: Fast and efficient gradient-free class forgetting. *Transactions on Machine Learning Research*, 2024. 3

[21] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019. 4

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4, 6

[23] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2023. 1, 2, 3, 6, 7, 8

[24] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[25] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10. IEEE, 2021. 2

[26] Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2023. 2, 3, 6, 7

[27] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 1749–1758. IEEE, 2022. 2

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. 2021. 3

[29] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021. 6

[30] Eric Rosenbach. Confronting the cyber threat. *Council on Foreign Relations*, 2018. 2

[31] Sara Rosenbaum. Data governance and stewardship: Designing data stewardship entities and advancing data access. *Health Services Research*, 53(S2):996–1005, 2018. 2

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 6

[33] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019. 3

[34] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3, 8

[35] Juwon Seo, Sung-Hoon Lee, Tae-Young Lee, Seungjun Moon, and Gyeong-Moon Park. Generative unlearning for any identity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2024. 2

[36] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015. 3

[37] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[38] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601, 2017. 3

[39] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track*, 2015. 4, 6

[40] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 2, 3

[41] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022. 5, 6, 7, 8

[42] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022. 2

[43] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 14(6):2073–2089, 2019. 3

[44] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springe3r Publishing Company, Incorporated, 2017. 1, 2

[45] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*, pages 622–632, 2022. 2

[46] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. In *Proc. of the 30th Network and Distributed System Security (NDSS)*, 2023. 6, 7, 8

[47] Yinjun Wu, Edgar Dobriban, and Susan Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning*, pages 10355–10366. PMLR, 2020. 2

[48] Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4006–4013, 2022. 2

[49] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, pages 14574–14584, 2020. 2