PeriUn: Enhancing Unlearning by Selectively Forgetting Peripheral Samples

HEE BIN YOO[1], **Dong-Sig Han**[2], **Jaein Kim**[1], **Byoung-Tak Zhang**[1*]

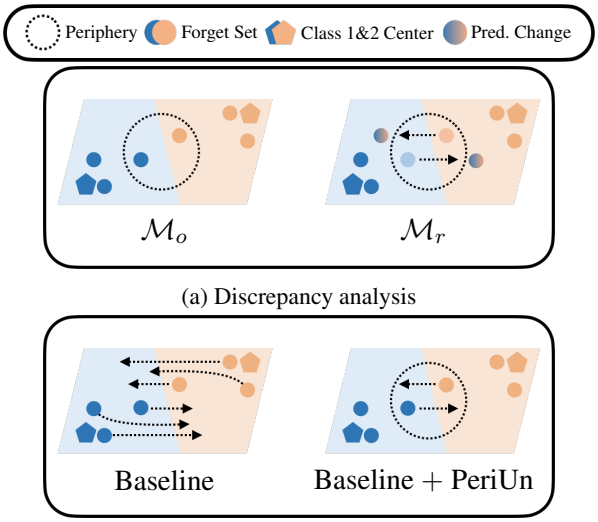
[1]Seoul National University,
[2]Imperial College London
yooheebin@snu.ac.kr, ehd4585@gmail.com, qpwodlsqp@snu.ac.kr, btzhang@snu.ac.kr

Abstract

Once trained, neural networks memorize information in diffusely encoded parameters, making it difficult to forget in support of the right to be forgotten. Unlearning aims to remove the influence of data, with performance measured against a retrained model that excludes the data. However, understanding the behavior of gold-standard retraining remains underexplored. We compare original and retrained models and observe that most prediction changes occur in peripheral samples near decision boundaries. Consequently, we propose PeriUn, a selective strategy that unlearns only peripheral samples to mimic retrained model behavior with minimal disruption, unlike prior works that remove the entire request. Combined with the Random Label based method, PeriUn significantly improves both generalization and privacy metrics. Specifically, on TinyImageNet with VGG16, PeriUn increases the Tug-of-War score by 22 points compared to the strongest. Besides, the MIA gap score surpasses the state-of-the-art method, improving by 8.7 points after applying PeriUn. Further analyses confirm that PeriUn better preserves the feature space and aligns closely with the retrained model.

Introduction

With growing concerns over privacy regulations and the "right to be forgotten" (Mantelero 2013; Information 2017), unlearning techniques designed to remove the influence of specific data from networks have garnered increasing attention (Cao and Yang 2015). As an ideal solution, a retrained-from-scratch model trained only on the retain set, i.e., samples not subject to deletion requests, serves as the oracle, but it is prohibitively expensive to obtain. Therefore, recent studies (Triantafillou et al. 2024; Ebrahimpour-Boroojeny, Sundaram, and Chandrasekaran 2025) have aimed to make unlearning methods approximate the accuracy of retraining. For instance, various studies have proposed to encourage sparsity (Jia et al. 2024) or preserve non-salient parameters (Fan et al. 2024). However, approximation based approaches have inherent limitations, as they may unintentionally remove useful information from the retain set, i.e., catastrophic forgetting, an issue addressed by He et al. (2024); Chen et al. (2024); Peng, Tang, and Yang (2025) through explicit retain set information preservation.



(a) Discrepancy analysis

(b) Proposed method PeriUn

Figure 1: (Top) Discrepancy observed between original model $\mathcal{M}_o$ and retrained model $\mathcal{M}_r$, where peripheral samples predominantly contribute to confidence and prediction changes. (Bottom) Comparison between PeriUn and the standard baseline. PeriUn is a selective unlearning strategy that provides only the periphery samples in the forget set to the unlearning algorithm. This approach mimics the behavior of retraining, thereby mitigating catastrophic forgetting.

Notably, catastrophic forgetting does not occur in the oracle retrained model. This observation motivates our central arguments; understanding and mimicking the behavior of the retrained model can guide more effective unlearning methods that mitigate catastrophic forgetting and improve performance. To explore these arguments, we investigate the following research questions:

> *What are the differences between the original and retrained models, and their causes? Can these differences be leveraged to enhance unlearning?*

To investigate these questions, we analyze changes in terms of confidence and prediction correctness of forget set samples by comparing the original and retrained models. In the original model, these samples tend to be predicted with over-

*Corresponding author

confidence and near-perfect accuracy, whereas both of their confidence and accuracy noticeably decline in the retrained model. To identify the causes responsible of this shift, we conduct additional analyses at the individual sample level. A notable finding is that predictions change from correct to incorrect with reduced confidence, and this occurs primarily in peripheral samples near decision boundaries. This directly addresses our first research question. In addition, our fine-grained analysis extends prior work (Seo, Kim, and Han 2025; Georgiev et al. 2025), which focuses solely on macroscopic changes between the original and retrained models.

Based on this observation, we hypothesize that selectively unlearning on peripheral samples rather than the entire set can effectively prevent catastrophic forgetting and improve performance. The above hypothesis guides our approach to the second research question. We propose Peripheral Unlearning, **PeriUn**, a selective unlearning strategy that targets only the peripheral samples within the forget set (see Figure 1), mimicking retrained model by focusing on samples that mainly cause changes in confidence and prediction.

Using PeriUn, we conducted comparative experiments with various leading unlearning methods across diverse architectures and datasets, evaluating their ability to mimic retraining in terms of performance, privacy, and prediction behavior. In particular, combining PeriUn with the Saliency Unlearn (Fan et al. 2024) led to significant improvements. For instance, on the TinyImageNet dataset with the VGG16-bn model, the ToW score improved by 22.04 points on a 100-point scale compared to the best baseline. Moreover, the MIA gap score surpasses the previous best method, improving by 8.7 points relative to baselines without PeriUn. In addition, we quantitatively and qualitatively analyzed the prediction showing PeriUn enables the unlearned model to better mimic the retrained model compared to the baseline, while reducing catastrophic forgetting. Furthermore, we validated the effectiveness of PeriUn through ablations.

The main contributions of this work are as follows:

- Our work presents a novel analysis of how individual forget-set samples change from the original to retrained models, uncovering patterns for effective unlearning.
- Building on these insights, we introduce a selective forgetting strategy that closely approximates the retrained model and mitigates catastrophic forgetting.
- Extensive experiments on image classification benchmarks demonstrate state-of-the-art ToW and MIA gap performance of PeriUn, validating its effectiveness.

Related Works

Coreset. Existing studies demonstrating the existence of coresets in neural network training inspire our selective forgetting approach. Notably, Toneva et al. (2019); Paul, Ganguli, and Dziugaite (2023) demonstrated that comparable results can be achieved without utilizing the entire training dataset. Similarly, Guo, Zhao, and Bai (2022) constructed coresets based on confidence-based selection, aligning with our approach.In addition, Jain et al. (2022); Zhang et al. (2023) confirmed the presence of coresets in transfer learning scenarios. Collectively, these studies on coresets suggest that, in the context of unlearning, it may be sufficient to selectively remove a few samples to mimic the retrained model, rather than indiscriminately deleting all data.

Memorization. Prior research on memorization supports both the validity and necessity of the Peripheral Unlearning strategy proposed in this work. Carlini, Erlingsson, and Papernot (2019); Jiang et al. (2021) empirically demonstrated that while some samples are correctly predicted even when excluded from training, others make incorrect predictions when held out. Notably, Jiang et al. (2021) observed that such vulnerable samples tend to be visually ambiguous and have lower average confidence. These findings align with Feldman and Zhang (2020); Feldman (2021), which showed that samples susceptible to incorrect prediction under hold-out are *memorized* samples and memorization contributes to model performance. Taken together, the above results suggest that it is more principled to selectively remove *memorized* samples supporting the strategy of PeriUn.

Unlearning. There have been diverse approaches to erase the influence of data to be deleted on the trained model. *Fine-Tuning (FT)* tunes the model using only the retain set, thereby inducing catastrophic forgetting of the forget set (Golatkar, Achille, and Soatto 2020). *Gradient Ascent (GA)* updates the model in the direction that maximizes the loss on the forget set, encouraging removal of the corresponding information (Thudi et al. 2022). *Random Label (RL)* perturbs the learned representation by assigning random labels to the forget set and fine-tuning the model on the entire dataset (Hayase, Yasutomi, and Katoh 2020). *Gradient Ascent–Gradient Descent (GAGD)* applies GA to the forget set and GD to the retain set, aiming to balance removal and preservation (Kurmanji et al. 2023). *Fine-Tuning with L1 Regularization (FT+L1)* encourages parameter sparsity through L1 regularization, improving unlearning effectiveness via pruning effects (Jia et al. 2024). *Saliency Unlearning (SU)* performs RL-style unlearning by selectively updating only salient parameters based on gradient analysis (Fan et al. 2024). Additional methods are in the appendix.

Problem Setup

We consider the problem of machine unlearning in image classification (Zhao et al. 2024; Fan et al. 2024). In particular, we focus on handling random instance removal requests to comply with user data deletion demands (Golatkar et al. 2021; Shibata et al. 2021; He et al. 2021). Given a training dataset $\mathcal{D}_{\text{train}}$, test dataset $\mathcal{D}_t$, and an original model $\mathcal{M}_o$ trained on $\mathcal{D}_{\text{train}}$, a user may request to remove the influence of a specific subset $\mathcal{D}_f \subset \mathcal{D}_{\text{train}}$, referred to as the forget set.

Approximate Unlearning. The ideal way to fulfill this request is to retrain a model from scratch on the remaining data $\mathcal{D}_r = \mathcal{D}_{\text{train}} \setminus \mathcal{D}_f$, resulting in a retrained model $\mathcal{M}_r$. This process, referred to as exact unlearning, produces an oracle model for comparison. However, retraining from scratch incurs computational costs equivalent to training $\mathcal{M}_o$, which is often prohibitively expensive in practice. As a result, approximate unlearning methods aim to transform $\mathcal{M}_o$ into an unlearned model $\mathcal{M}_u$ without full retraining. These methods

	Dataset - Model	Train	Test
1	CIFAR10 - ResNet18	99.46±0.11	92.31±0.42
2	CIFAR100 - ResNet50	99.97±0.01	76.32±0.26
3	TinyImagenet - ResNet18	99.98±0.00	65.19±0.35
4	TinyImagenet - VGG16-bn	99.98±0.00	60.40±0.59

Table 1: Train/test classification accuracy of the original models $\mathcal{M}_o$ used in the unlearning experiments. The networks classify the training data with near-perfect accuracy.

are given access to $\mathcal{M}_o$, the forget set $\mathcal{D}_f$, and the retain set $\mathcal{D}_r$, and seek to approximate the behavior of $\mathcal{M}_r$.

Evaluation Metric. To evaluate the quality of unlearning, we adopt the Tug-of-War (ToW) and MIA gap metric introduced by Zhao et al. (2024). ToW evaluates how well $\mathcal{M}_u$ mimics the predictive behavior of $\mathcal{M}_r$. At the same time, the MIA gap captures the similarity in membership inference attack success rates, reflecting alignment in privacy behavior.

Definition 1 (Tug-of-War)**.** Let a_s^m denote the classification accuracy of model $\mathcal{M}_m$ on dataset $\mathcal{D}_s$. The ToW score between $\mathcal{M}_u$ and $\mathcal{M}_r$ is defined as:

$$\text{ToW}(\mathcal{M}_r, \mathcal{M}_u, \mathcal{D}_f, \mathcal{D}_r, \mathcal{D}_t) = (1 - |a_f^r - a_f^u|) \cdot (1 - |a_r^r - a_r^u|) \cdot (1 - |a_t^r - a_t^u|) \cdot 100. \quad (1)$$

A ToW score closer to 100 indicates a better approximation of the retrained model's behavior across all data splits.

Experimental Setup. We evaluate unlearning performance on four dataset–architecture pairs. Specifically, we use CIFAR10, CIFAR100 (Krizhevsky 2009), and TinyImageNet (Le and Yang 2015) with ResNet18, ResNet50 (He et al. 2015), and VGG16-bn (Simonyan and Zisserman 2015) following Zhao et al. (2024). Each row in Table 1 corresponds to *Setup 1–4* and reports the accuracy of $\mathcal{M}_o$. For statistical significance, we use three different fixed random initializations. Then, we generate and fix three random forget sets of size 3000. We utilize an NVIDIA RTX 3090, an Intel Xeon 5218, and 128 GB of memory with Ubuntu 18.04.

Preliminary Analysis of the Retrained Model

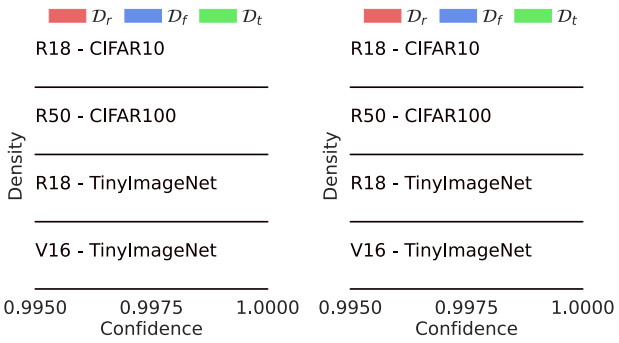
To address the first question, we analyze output differences between $\mathcal{M}_o$ and $\mathcal{M}_r$, especially on $\mathcal{D}_f$, and examine how these results can be leveraged to improve unlearning.

Observation 1. We demonstrate that the output prediction of data subsets $(\mathcal{D}_r, \mathcal{D}_f, \mathcal{D}_t)$ drawn from the same underlying distribution, split by i.i.d. sampling, can be separated based on whether the data were used during training or not.

Observation 2. Building on the above observation and considering that $\mathcal{D}_f$ is used for training in $\mathcal{M}_o$ but not in $\mathcal{M}_r$, we analyze what individual samples in $\mathcal{D}_f$ change to exhibit distinct characteristics as seen and unseen in train.

Observation 1: Seen vs. Unseen

We observe that the confidence distributions are distinguishable concerning whether $\mathcal{D}_r$, $\mathcal{D}_f$, or $\mathcal{D}_t$ was used during training, and samples within each data subset share common internal characteristics. Note that we define confidence as the maximum softmax probability across all classes. We measure the confidence density of $\mathcal{M}_o$ and $\mathcal{M}_r$ trained under four different setups and visualize the results of the distributions. In detail, we subsample each set to match the size of $|\mathcal{D}_f|$ for a fair comparison and plot the histogram's KDE.



(a) $\mathcal{M}_o$ (b) $\mathcal{M}_r$

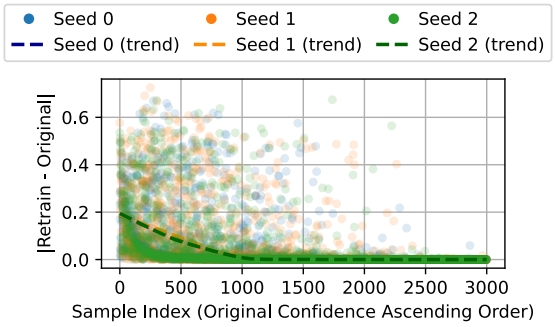
Figure 2: Observation 1: Seen vs. Unseen confidence analysis. Seen data refers to $(\mathcal{D}_r, \mathcal{D}_f)$ under $\mathcal{M}_o$, as well as $\mathcal{D}_r$ under $\mathcal{M}_r$, which are used in training, while all others are considered unseen. We observe that the model is overconfident in seen data, while exhibiting low confidence in unseen. Note that in (b), $\mathcal{D}_f$ and $\mathcal{D}_t$ overlap.

Our experimental results in Figure 2 show that samples in the seen subsets tend to exhibit overconfidence, while those in the unseen set generally show underconfidence. This observation is consistent with findings from prior studies (Shokri et al. 2017; Datta et al. 2025). We also found consistent differences in accuracy between seen and unseen data, regardless of data subsets or random seed. As shown in Table 1 and Table 2, the average accuracy on seen datasets is always near 100% within one percentage point. In comparison, accuracy differences among unseen datasets remain within one percentage point. These results confirm that the datasets $\mathcal{D}_r$, $\mathcal{D}_f$, and $\mathcal{D}_t$, which are separated via i.i.d. sampling, can be divided into two groups based on their training exposure, i.e., seen vs. unseen, and that each group shares nearly identical accuracy and confidence distribution patterns. Additional results are in the appendix *Confidence Distribution Results*.

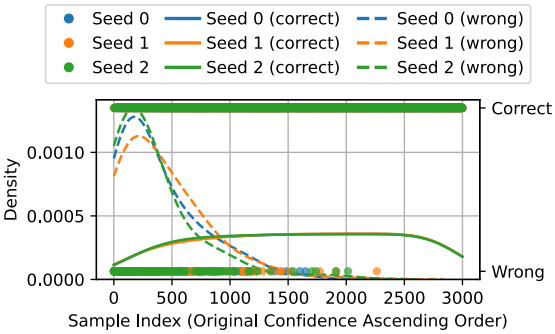
Observation 2: Discrepancy Analysis on $\mathcal{D}_f$

The forget dataset $\mathcal{D}_f$ is seen by $\mathcal{M}_o$ but unseen by $\mathcal{M}_r$, leading $\mathcal{M}_o$ to show overconfident, near-perfect accuracy, while $\mathcal{M}_r$ has lower confidence and accuracy on $\mathcal{D}_f$. To understand what drives this difference at the level of individual samples, we perform an analysis to identify samples that contribute to the shift from seen to unseen behavior.

Confidence Difference Analysis. We measured the confidence of each sample in the forget set $\mathcal{D}_f$ using both $\mathcal{M}_o$ and $\mathcal{M}_r$, then compared their differences. As in Figure 3a, samples were sorted in ascending order based on their confidence from $\mathcal{M}_o$, and Locally Weighted Scatterplot Smoothing was applied to visualize the overall trend. High-confidence samples in $\mathcal{M}_o$ showed little change and,

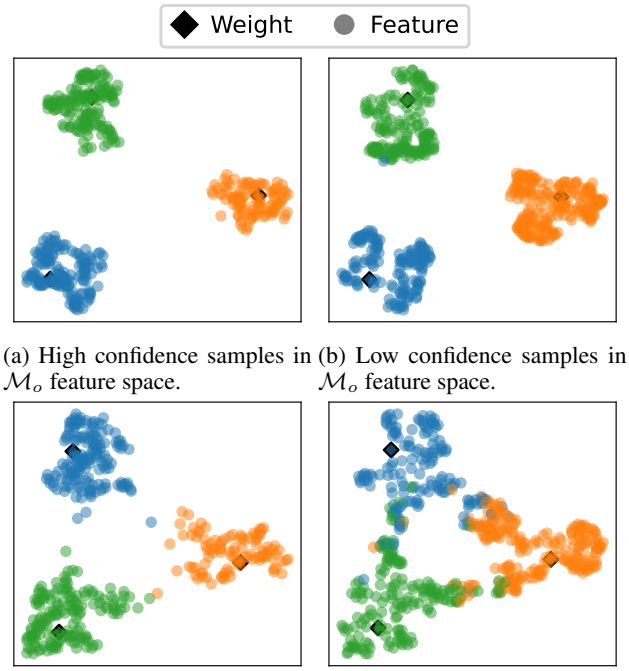


(a) Confidence difference between $\mathcal{M}_o$ and $\mathcal{M}_r$ on $\mathcal{D}_f$



(b) Correctness density of $\mathcal{M}_r$ on $\mathcal{D}_f$.

Figure 3: Discrepancy analysis on $\mathcal{D}_f$ in Setup 1. Low-confidence samples exhibit greater confidence shifts and are the most misclassified samples in $\mathcal{M}_r$.



(a) High confidence samples in $\mathcal{M}_o$ feature space.

(b) Low confidence samples in $\mathcal{M}_o$ feature space.

(c) High confidence samples in $\mathcal{M}_r$ feature space.

(d) Low confidence samples $\mathcal{M}_r$ feature space.

Figure 4: T-SNE of $\mathcal{D}_f$ features from $\mathcal{M}_o$ and $\mathcal{M}_r$ in Setup 1. Samples are split by the median $\mathcal{M}_o$ confidence.

thus, remained confident in $\mathcal{M}_r$, while lower-confidence samples exhibited larger changes. This suggests that less confident samples are more susceptible to retraining. This result can be interpreted from the perspective of the Neural Collapse (NC) (Papyan, Han, and Donoho 2020), where features align with the corresponding classifier weight, inducing overconfidence. From this view, *periphery* features that are farther from the class center, i.e., those with lower confidence, tend to show prediction changes in retraining.

Correctness Change Analysis. We evaluated the correctness of each sample in $\mathcal{D}_f$ under $\mathcal{M}_r$, then sorted the samples in ascending order based on confidence from $\mathcal{M}_o$. We applied Kernel Density Estimation to visualize the distributions of correctly and incorrectly predicted samples. Note that, due to the interpolation regime (Gamba et al. 2023; Block, Mokhtari, and Shakkottai 2025), $\mathcal{D}_f$ achieved near-perfect accuracy in $\mathcal{M}_o$. Therefore, we interpret the incorrect samples in $\mathcal{D}_f$ as those whose labels changed from correct to incorrect. As shown in Figure 3b, samples with low confidence in $\mathcal{M}_o$ tended to be misclassified by $\mathcal{M}_r$, whereas correctly predicted samples were distributed evenly.

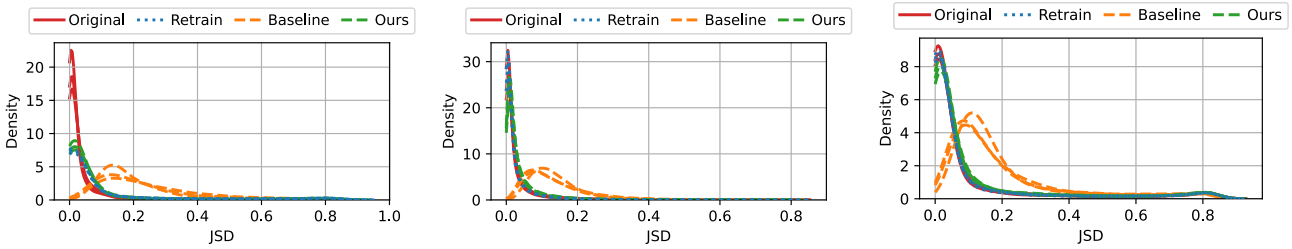
Feature Change Analysis. For qualitative analysis, we extracted features from the penultimate layer for $\mathcal{M}_o$ and $\mathcal{M}_r$. We performed T-SNE on samples split by the lower and upper 50% confidence from $\mathcal{M}_o$. The results in Figure 4 show that samples with high confidence form tight clusters around the class weight, while low-confidence samples are more peripheral. Moreover, in $\mathcal{M}_r$, as low-confidence samples become more intermixed near class boundaries than high-confidence samples. These results provide visual evidence that low-confidence *periphery* samples undergo greater changes in features and predictions at retraining.

Brief Discussion. We identified low-confident *peripheral* samples as the primary drivers of change during retraining and thus interpretable as " forgotten." This trend was consistent across setups. See Appendix *Discrepancy Analysis*.

Method

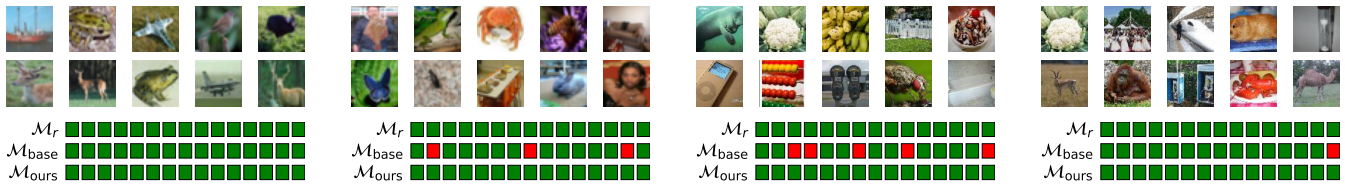
Building on our analysis that *periphery* samples contribute significantly to changes caused by retraining, we propose a simple principle to improve approximate unlearning: *unlearn only uncertain*. We introduce **Peri**pheral **Un**learning, **PeriUn** (see Figure 1b) that selects the subset of the forget set $\mathcal{D}_f$ consisting of low-confidence samples near the decision boundary. We hypothesize that our principle prevents catastrophic forgetting during the unlearning by safely ignoring confident samples of forget set $\mathcal{D}_f$, which remain unchanged in retraining, and better approximates $\mathcal{M}_r$.

PeriUn is described in Algorithm 1. Given an original model $\mathcal{M}_o$, a forget set $\mathcal{D}_f = (X_f, Y_f)$, and a selection ratio $\alpha \in (0, 1)$, PeriUn constructs a new forget set by selecting the least confident samples in $\mathcal{D}_f$. Specifically, for each input $x_i \in X_f$, the model outputs a probability vector $p_i \leftarrow \mathcal{M}_o(x_i)$, and the confidence score is computed



(a) Forget Set (b) Retain Set (c) Test Set

Figure 5: Jenson-Shannon distance distribution between one-hot label vectors and softmax outputs in CIFAR10 / ResNet18 (Setup 1). Our PeriUn alleviates catastrophic forgetting and better approximates the retrain model compared to the SU baseline. We visualize the distribution using the first unlearn seed and all three init seeds, overlaid in the same color.



(a) CIFAR10 / ResNet18 (b) CIFAR100 / ResNet50 (c) TinyImageNet / ResNet18 (d) TinyImageNet / VGG16-bn

Figure 6: Qualitative comparison between PeriUn and Baseline (SU). (Above) Top 10 most confident samples in $\mathcal{M}_f$ selected, mostly visually normal. Images are ordered from left to right, top to bottom, in descending order of confidence. (Below) The top 15 highest-confidence samples are highlighted in red for those that were originally correct but later changed to incorrect (i.e., deletion) by each model. Retrain and Periun tend to preserve high-confidence samples, while Baseline deletes all requested samples, including high-confidence ones that do not need to be removed, which suggests catastrophic forgetting.

as $c_i \leftarrow \max_j p_{ij}$. We then compute the threshold τ as the α-quantile of these confidence scores. Samples with confidence $c_i \leq \tau$, i.e., confidence lower than or equal to this threshold τ, are selected to form the new forget set $\mathcal{D}_f^{\text{new}}$.

Algorithm 1: Selecting low confident samples in $\mathcal{D}_f$

Input 1: Original model $\mathcal{M}_o$
Input 2: Forget set $\mathcal{D}_f = (X_f, Y_f)$
Parameter: Number of classes K, Selection ratio α
Output: New forget set $\mathcal{D}_f^{new}$

1: Compute probabilities

$$p \leftarrow \mathcal{M}_o(X_f) \in \mathbb{R}^{|\mathcal{D}_f| \times K}.$$

2: Let confidence $c \in \mathbb{R}^{|\mathcal{D}_f|}$ where $c_i \leftarrow \max_j p_{ij}$
3: Determine threshold $\tau \leftarrow \text{Quantile}_\alpha(c)$.
4: Select indices $I = \{i \mid c_i \leq \tau\}$.
5: Extract truncated forget set $\mathcal{D}_f^{new} = \mathcal{D}_f[I]$.
6: **return** $\mathcal{D}_f^{new}$

Experiments

Our experiments aim to assess whether PeriUn effectively mimics retraining behavior and improves unlearning performance, thereby addressing our second research question. To this end, we pursue the following objectives.

- Analyze whether PeriUn produces softmax predictions that more closely resemble those of the retrained model.
- Evaluate whether PeriUn enhances existing methods in terms of generalization and privacy performance.
- Investigate the impact of the selection ratio α.
- Assess the compatibility of PeriUn on various baselines.

Hyperparameter. We share hyperparameter search procedures and values in Appendix Hyperparameter Information.
Understanding the Effect of PeriUn. We claim that PeriUn better approximates the retrained model than baselines, while preventing catastrophic forgetting that would overly alter the output. To verify this, we use RL and SU, which directly manipulate features and integrate well with **PeriUn**, and conduct quantitative and qualitative analyses.

To quantitatively analyze the prediction behavior, we measure the Jensen-Shannon distance (JSD) distributions between one-hot labels and the softmax outputs from $\mathcal{D}_f$, $\mathcal{D}_r$, and $\mathcal{D}_t$ to compare $\mathcal{M}_o$, $\mathcal{M}_r$, baseline, and baseline+PeriUn. Figure 5 presents the results for CIFAR10 and ResNet18 (Setup 1) on SU. The results indicate that baseline+PeriUn yields output distributions consistently closer to $\mathcal{M}_r$ than the baseline. Moreover, the baseline exhibits higher JSD after unlearning, reflecting lower confidence in the correct class and weaker convergence toward Neural Collapse, which suggests more severe catastrophic forgetting. These suggest that, unlike baseline methods that indiscriminately modify all forget set samples and cause redun-

Setup	1 - CIFAR10 / ResNet18				2 - CIFAR100 / ResNet50			
	Forget	Retain	Test	ToW	Forget	Retain	Test	ToW
Retrain	92.55±0.43	99.55±0.11	92.13±0.31		76.15±0.98	99.97±0.00	76.01±0.67	
FT	95.93±0.51	98.51±0.37	90.62±0.49	94.18±0.84	74.79±1.93	87.67±1.41	64.86±1.06	76.37±3.05
FT+L1	95.06±0.54	97.90±0.25	90.16±0.41	94.00±0.63	99.98±0.02	99.97±0.00	76.18±0.29	75.59±0.99
GA	99.47±0.17	99.46±0.10	92.21±0.29	92.60±0.68	99.92±0.05	99.97±0.00	76.19±0.34	75.69±0.96
GAGD	94.62±0.58	99.40±0.07	91.52±0.27	97.12±0.79	79.32±2.60	99.04±0.17	70.73±0.44	90.73±1.67
RL	80.64±2.01	98.21±0.38	89.70±0.66	84.82±2.87	48.70±1.20	91.88±1.24	64.95±0.89	59.33±2.40
SU	94.36±2.12	99.42±0.17	91.46±0.56	96.77±1.63	73.20±3.13	99.95±0.01	73.90±0.40	94.58±2.49
SU+ours	93.92±0.70	99.60±0.10	91.60±0.31	**97.92±0.83**	76.22±0.74	99.94±0.01	74.62±0.29	**97.78±0.89**
Setup	3 - TinyImagenet / ResNet18				4 - TinyImagenet / VGG16-bn			
	Forget	Retain	Test	ToW	Forget	Retain	Test	ToW
Retrain	64.21±1.14	99.98±0.00	64.49±0.34		59.90±1.27	99.98±0.00	60.28±0.39	
FT	81.09±1.52	96.57±0.75	58.00±0.96	75.06±0.88	71.67±2.21	82.66±2.30	49.04±1.57	64.70±1.38
FT+L1	68.79±0.90	90.21±1.59	55.79±0.79	78.61±1.41	68.20±1.08	85.96±1.52	50.98±0.69	71.49±1.03
GA	99.88±0.07	99.98±0.01	64.48±0.24	64.14±1.12	97.58±3.54	98.58±3.14	57.83±2.59	59.79±1.21
GAGD	60.22±2.14	99.44±1.07	60.26±0.65	91.45±1.94	57.74±16.81	87.94±24.81	45.75±11.96	73.48±25.23
RL	31.99±1.32	87.06±1.11	52.86±0.80	52.18±1.93	29.63±1.28	86.19±1.69	49.25±1.12	53.51±2.52
SU	55.41±1.23	99.97±0.01	62.51±0.43	89.39±1.53	35.64±1.96	99.97±0.00	58.00±0.32	74.01±1.64
SU+ours	64.08±0.86	99.97±0.01	63.27±0.27	**97.55±0.68**	62.94±0.80	99.98±0.00	59.34±0.30	**96.05±1.26**

Table 2: Comparison with existing methods. Forget, retain, and test accuracies, along with ToW scores, are reported across four setups. Applying our method to SU improves performance in all cases.

dant damage, our method selectively unlearns, more closely mimicking retraining without catastrophic forgetting.

For qualitative analysis, we confirmed in Figure 6 that PeriUn imitates the correctness behavior of the retrained model, which preserves high-confidence ones. Visual inspection showed the top 10 highest-confidence $\mathcal{D}_f$ samples of $\mathcal{M}_o$ are regular and recognizable images likely classifiable and indeed remain correct without inclusion in retraining. Furthermore, an analysis of the top 15 high-confidence samples that changed from correct to incorrect predictions reveals that both the retrained model and PeriUn successfully preserved them, unlike the baseline. This indicates that the baseline removes unnecessary samples to be erased according to retraining, whereas PeriUn effectively replicates the behavior of retrained model. Additional results for this section are in the appendix under *Comparison with Baseline*.

Comparison with the State-of-the-art Baselines. As shown in Table 2, we observed consistent improvements in ToW performance when applying PeriUn to SU, the state-of-the-art unlearning approach based on Random Label. Our approach, SU+PeriUn, outperforms the best ToW scores of all existing baselines across all experimental setups. Moreover, compared to the best-performing baseline in each case, our method achieves average ToW improvements of 0.8 points on CIFAR-10 with ResNet18, 3.2 points on CIFAR-100 with ResNet50, 6.1 points on TinyImageNet with ResNet18, and 22.04 points on TinyImageNet with VGG16-bn. These results indicate that PeriUn achieves an ToW score of 96 across all setups, demonstrating its ability to mimic the performance of the retrained model closely.

Effect of Selection Ratio. We conduct an ablation study on the selection ratio α. Since high-confidence samples often remain correctly predicted after retraining, excluding them may preserve key information and improve similarity to $\mathcal{M}_r$. Conversely, excluding too many samples can hinder unlearning. We thus expect an inverted U-shaped trend with respect to the selection ratio α, where performance improves up to a certain point and then declines. This supports the notion that our periphery-based selection is an adjustable algorithm that corresponds to our hypothesis.

Setup		1 - C10 / Res18	2 - C100 / Res50
α	0.1	**97.92±0.83 (+1.14)**	84.14±0.59 (-10.44)
	0.3	97.56±1.69 (+0.79)	**97.78±0.89 (+3.20)**
	0.5	97.06±1.31 (+0.29)	97.12±1.48 (+2.54)
	0.7	96.95±1.74 (+0.18)	94.09±3.22 (-0.48)
	1.0	96.77±1.63 (Base)	94.58±2.49 (Base)
Setup		3 - TinyIn / Res18	4 - TinyIn / VGG16
α	0.1	80.59±0.90 (-8.80)	76.62±1.09 (+2.62)
	0.3	89.12±1.01 (-0.27)	89.00±1.40 (+15.00)
	0.5	**97.55±0.68 (+8.16)**	**96.05±1.26 (+22.04)**
	0.7	96.41±1.20 (+7.01)	89.17±1.30 (+15.17)
	1.0	89.39±1.53 (Base)	74.01±1.64 (Base)

Table 3: ToW measure across different selection ratios. As the ratio decreases from 1.0, performance initially increases and then decreases, exhibiting an inverted U-shaped trend.

To test this, under the SU+PeriUn setting, we varied the selection ratio $\alpha \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ and measured ToW as in Table 3. Across all setups, we observed an inverted U-shape trend in performance with respect to the selection ratio. It validates that excluding high-confidence samples up to a certain level can contribute positively to performance. This provides empirical support for the design rationale of PeriUn. While a slight performance drop was observed at $\alpha = 0.7$ in the CIFAR100 / ResNet50 (Setup 2), it was not statistically significant (paired t-test, $p = 0.35$).

Setup	1 - C10 / Res18		2 - C100 / Res50	
	MIA	Gap	MIA	Gap
Retrain	14.75±0.61		46.07±1.23	
FT	9.73±1.02	5.02	30.99±1.81	15.09
FT+L1	10.88±0.80	<u>3.87</u>	1.53±0.21	44.54
GAGD	9.50±0.83	5.25	35.45±3.76	**10.62**
GA	3.74±0.74	11.01	1.76±0.19	44.31
RL	47.62±4.42	32.87	73.69±1.29	27.61
SU	23.28±6.07	8.53	91.79±1.09	45.71
SU+ours	12.02±1.23	**2.73**	33.84±0.40	<u>12.24</u>
Setup	3 - TinyIn / Res18		4 - TinyIn / VGG16	
	MIA	Gap	MIA	Gap
Retrain	63.24±1.10		61.37±1.14	
FT	31.31±1.56	31.93	29.04±1.58	32.33
FT+L1	37.74±0.99	25.50	33.73±0.99	27.64
GAGD	56.44±3.67	**6.79**	48.56±11.95	<u>12.81</u>
GA	4.71±0.47	58.53	6.84±4.26	54.53
RL	76.84±1.50	13.60	78.03±1.28	16.66
SU	96.42±0.45	33.18	81.73±0.96	20.36
SU+ours	50.90±0.30	<u>12.34</u>	49.75±0.14	**11.62**

Table 4: Confidence-based MIA gap. Our SU+PeriUn achieves comparable or superior performance to existing methods and significantly improves the MIA gap of SU.

Method Evaluation Based on MIA Gap. To evaluate whether our method not only improves ToW but also better approximates the privacy behavior of the retrained model, we measure MIA accuracy for each method and compute the gap to the retrained model, quantifying how closely each method mimics privacy behavior. Following Zhao et al. (2024); Jia et al. (2024), we primarily focus on confidence-based MIA. Additionally, we evaluate MIA performance using two further metrics, output probability and entropy, to provide a more comprehensive analysis. Results for these additional metrics are reported in the appendix *MIA Results*. As shown in Table 4, SU+PeriUn significantly reduces the confidence-based MIA gap compared to SU, suggesting improved alignment with the privacy characteristics of the retrain model. This highlights that SU in itself is vulnerable in privacy despite its strong ToW performance. In contrast, our method enhances both utility and privacy, demonstrating better alignment with the retrain model. Notably, SU+PeriUn achieves the best MIA gap in CIFAR10-ResNet18 and TinyImagenet-ResNet18 (Setups 1 and 3), and ranks second-best in the remaining cases. These results indicate that our approach provides a favorable trade-off between utility and privacy guarantees in unlearning.

Applying PeriUn to Other Baselines. We applied our method to RL, GA, and GAGD models, all of which utilize the forget set $\mathcal{D}_f$. In Table 5, RL exhibited a significant performance improvement similar to SalUn, with a notable ToW gain of 36.32 points on Setup 2. GAGD showed performance improvements in most settings. We attribute the effectiveness of PeriUn on RL and SU, where samples in $\mathcal{D}_f$ are intentionally reassigned to incorrect classes. By discarding non-peripheral samples, PeriUn reduces over-deletion and helps RL and SU better mimic the behavior of a retraining. In contrast, GA and GAGD apply negative gradients that perturb the entire feature space. This disruptive nature likely undermines the benefits of PeriUn, as selective removal of peripheral samples becomes less meaningful.

Setup	1 - C10 / Res18	2 - C100 / Res50
GAGD	97.12±0.79	90.73±1.67
GAGD+ours	96.64±0.75 (-0.48)	**93.00±1.55 (+2.27)**
GA	92.60±0.68	75.69±0.96
GA+ours	92.77±0.65 (+0.17)	75.73±0.89 (+0.04)
RL	84.82±2.87	59.33±2.40
RL+ours	**97.76±0.43 (+12.94)**	**95.65±1.46 (+36.32)**
Setup	3 - TinyIn / Res18	4 - TinyIn / VGG16
GAGD	91.45±1.94	73.48±25.23
GAGD+ours	**93.07±1.71 (+1.62)**	**76.78±8.83 (+3.30)**
GA	64.14±1.12	59.79±1.21
GA+ours	64.11±1.10 (-0.03)	59.91±1.35 (+0.12)
RL	52.18±1.93	53.51±2.52
RL+ours	**86.90±1.88 (+34.72)**	**79.99±1.91 (+26.48)**

Table 5: Our method was applied to GAGD, GA, and RL. Notable improvements are observed in RL, the base of SU.

Discussion

Our work introduces PeriUn, a method motivated by retrained model analysis, memorization, and coreset. Through this approach, we address two key research questions. First, we observe that the original and retrained models differ on the forget set, and attribute these differences primarily to peripheral samples. Second, we propose PeriUn, which improves unlearning while reducing unintended damage.

Moreover, our observation that samples altered to incorrect predictions tend to be low-confidence peripheral samples offers a practical connection to memorization. Prior work has shown a correlation between average confidence across training and memorization (Jiang et al. 2021), and leveraged average confidence as a proxy (Zhao et al. 2024). In contrast, we find that the relative ranking of post-training confidence alone is sufficient to identify such vulnerable samples, without requiring per-epoch tracking. This significantly reduces computational overhead and offers a direct bridge between memorization and unlearning.

Lastly, we show that unlearning a carefully selected subset is sufficient to approximate retraining and even improves performance across all metrics. The performance of PeriUn validates our hypothesis that a coreset-like subset exists in unlearning and that removing it selectively is beneficial, a notion we explore here for the first time.

Conclusion

PeriUn presents a novel yet simple unlearning approach that mimics retraining based on empirical analyses. This work advances unlearning by introducing a selective strategy, which balances privacy and performance more effectively than indiscriminate forgetting. Our findings highlight a promising direction that leverages feature space properties for effective and selective forgetting.

Acknowledgments

We are deeply grateful to Inwon Lee for her invaluable support and encouragement. We are also grateful to Won-Seok Choi for insightful scholarly discussions.

This work was partly supported by the IITP (RS-2021-II212068-AIHub/10%, RS-2021-II211343-GSAI/10%, RS-2022-II220951-LBA/15%, RS-2022-II220953-PICA/15%), NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/15%), KEIT (RS-2024-00423940/10%), and Gwangju Metropolitan City (Artificial intelligence industrial convergence cluster development project/10%) grant funded by the Korean government.

References

Block, J. L.; Mokhtari, A.; and Shakkottai, S. 2025. Machine Unlearning under Overparameterization. arXiv:2505.22601.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480.

Carlini, N.; Erlingsson, U.; and Papernot, N. 2019. Prototypical Examples in Deep Learning: Metrics, Characteristics, and Utility.

Chen, H.; Zhu, T.; Yu, X.; and Zhou, W. 2024. Machine Unlearning via Null Space Calibration. arXiv:2404.13588.

Datta, E.; Hennig, J.; Domschot, E.; Mattes, C.; and Smith, M. R. 2025. Topology of Out-of-Distribution Examples in Deep Neural Networks. arXiv:2501.12522.

Ebrahimpour-Boroojeny, A.; Sundaram, H.; and Chandrasekaran, V. 2025. AMUN: Adversarial Machine UNlearning. arXiv:2503.00917.

Fan, C.; Liu, J.; Zhang, Y.; Wong, E.; Wei, D.; and Liu, S. 2024. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. arXiv:2310.12508.

Feldman, V. 2021. Does Learning Require Memorization? A Short Tale about a Long Tail. arXiv:1906.05271.

Feldman, V.; and Zhang, C. 2020. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. arXiv:2008.03703.

Gamba, M.; Englesson, E.; Björkman, M.; and Azizpour, H. 2023. Deep Double Descent via Smooth Interpolation. arXiv:2209.10080.

Georgiev, K.; Rinberg, R.; Park, S. M.; Garg, S.; Ilyas, A.; Madry, A.; and Neel, S. 2025. Machine Unlearning via Simulated Oracle Matching. In *The Thirteenth International Conference on Learning Representations*.

Golatkar, A.; Achille, A.; Ravichandran, A.; Polito, M.; and Soatto, S. 2021. Mixed-Privacy Forgetting in Deep Networks. arXiv:2012.13431.

Golatkar, A.; Achille, A.; and Soatto, S. 2020. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. arXiv:1911.04933.

Guo, C.; Zhao, B.; and Bai, Y. 2022. DeepCore: A Comprehensive Library for Coreset Selection in Deep Learning. arXiv:2204.08499.

Hayase, T.; Yasutomi, S.; and Katoh, T. 2020. Selective Forgetting of Deep Networks at a Finer Level than Samples. arXiv:2012.11849.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

He, Y.; Meng, G.; Chen, K.; He, J.; and Hu, X. 2021. DeepObliviate: A Powerful Charm for Erasing Data Residual Memory in Deep Neural Networks. arXiv:2105.06209.

He, Z.; Li, T.; Cheng, X.; Huang, Z.; and Huang, X. 2024. Towards Natural Machine Unlearning. arXiv:2405.15495.

Information, C. L. 2017. Ab-375 privacy: personal information: businesses. Accessed: March 2, 2020. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

Jain, S.; Salman, H.; Khaddaj, A.; Wong, E.; Park, S. M.; and Madry, A. 2022. A Data-Based Perspective on Transfer Learning. arXiv:2207.05739.

Jia, J.; Liu, J.; Ram, P.; Yao, Y.; Liu, G.; Liu, Y.; Sharma, P.; and Liu, S. 2024. Model Sparsity Can Simplify Machine Unlearning. arXiv:2304.04934.

Jiang, Z.; Zhang, C.; Talwar, K.; and Mozer, M. C. 2021. Characterizing Structural Regularities of Labeled Data in Overparameterized Models. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 5034–5044. PMLR.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.

Kurmanji, M.; Triantafillou, P.; Hayes, J.; and Triantafillou, E. 2023. Towards Unbounded Machine Unlearning. arXiv:2302.09880.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Mantelero, A. 2013. The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3): 229–235.

Papyan, V.; Han, X.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663.

Paul, M.; Ganguli, S.; and Dziugaite, G. K. 2023. Deep Learning on a Data Diet: Finding Important Examples Early in Training. arXiv:2107.07075.

Peng, Z.; Tang, Y.; and Yang, Y. 2025. Adversarial Mixup Unlearning. arXiv:2502.10288.

Seo, S.; Kim, D.; and Han, B. 2025. Revisiting Machine Unlearning with Dimensional Alignment. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3206–3215.

Shibata, T.; Irie, G.; Ikami, D.; and Mitsuzumi, Y. 2021. Learning with Selective Forgetting. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 989–996. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks against Machine Learning Models. arXiv:1610.05820.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Thudi, A.; Deza, G.; Chandrasekaran, V.; and Papernot, N. 2022. Unrolling SGD: Understanding Factors Influencing Machine Unlearning. arXiv:2109.13398.

Toneva, M.; Sordoni, A.; des Combes, R. T.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2019. An Empirical Study of Example Forgetting during Deep Neural Network Learning. arXiv:1812.05159.

Triantafillou, E.; Kairouz, P.; Pedregosa, F.; Hayes, J.; Kurmanji, M.; Zhao, K.; Dumoulin, V.; Junior, J. J.; Mitliagkas, I.; Wan, J.; Hosoya, L. S.; Escalera, S.; Dziugaite, G. K.; Triantafillou, P.; and Guyon, I. 2024. Are we making progress in unlearning? Findings from the first NeurIPS unlearning competition. arXiv:2406.09073.

Zhang, Y.; Zhang, Y.; Chen, A.; Jia, J.; Liu, J.; Liu, G.; Hong, M.; Chang, S.; and Liu, S. 2023. Selectivity Drives Productivity: Efficient Dataset Pruning for Enhanced Transfer Learning. arXiv:2310.08782.

Zhao, K.; Kurmanji, M.; Bărbulescu, G.-O.; Triantafillou, E.; and Triantafillou, P. 2024. What makes unlearning hard and what to do about it. arXiv:2406.01257.