Toward Efficient Data-Free Unlearning

Chenhao Zhang[1], Shaofei Shen[1], Weitong Chen[2], Miao Xu[1]*

[1]University of Queensland
[2]University of Adelaide
chenhao.zhang@uq.edu.au, shaofei.shen@uq.edu.au, weitong.chen@adelaide.edu.au, miao.xu@uq.edu.au

Abstract

Machine unlearning without access to real data distribution is challenging. The existing method based on data-free distillation achieved unlearning by filtering out synthetic samples containing forgetting information but struggled to distill the retaining-related knowledge efficiently. In this work, we analyze that such a problem is due to over-filtering, which reduces the synthesized retaining-related information. We propose a novel method, Inhibited Synthetic PostFilter (ISPF), to tackle this challenge from two perspectives: First, the Inhibited Synthetic, by reducing the synthesized forgetting information; Second, the PostFilter, by fully utilizing the retaining-related information in synthesized samples. Experimental results demonstrate that the proposed ISPF effectively tackles the challenge and outperforms existing methods.

Code — https://github.com/ChildEden/ISPF
Extended version — https://arxiv.org/abs/2412.13790

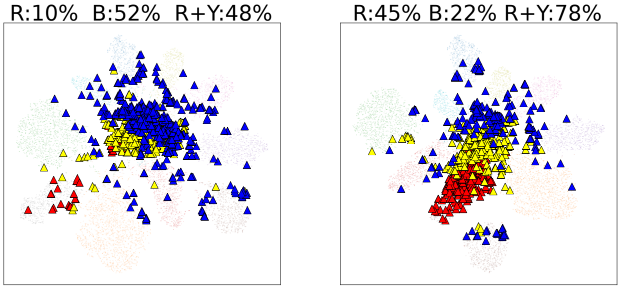
1 Introduction

Machine unlearning (Bourtoule et al. 2021; Nguyen et al. 2022; Xu et al. 2024) is an emerging paradigm that enables machine learning models to selectively forget training data, primarily to enhance user privacy and comply with regulations (Garg, Goldwasser, and Vasudevan 2020; BUKATY 2019), or to correct errors within the dataset (Cao and Yang 2015; Marchant, Rubinstein, and Alfeld 2022). Existing unlearning methods typically require access to the original training data to accurately discern which information needs to be removed (the "forgetting" data) and which must be preserved (the "retaining" data). This access is crucial for precisely identifying the parameters impacted by the forgetting data, ensuring that only these parameters are adjusted (Fan et al. 2023; Foster, Schoepf, and Brintrup 2024) while the functionality of the model is maintained for the retaining data (Thudi et al. 2022; Kurmanji et al. 2023). However, in practice, the original training data may not always be available due to reasons such as compliance with privacy laws that mandate data deletion, or organizational policies aimed at optimizing storage efficiency. Moreover, alternative data sources that share a similar distribution with the training data might also be inaccessible due to legal restrictions. Despite these challenges in data access, requests for unlearning tasks such as forgetting specific concepts or classes of data can still be initiated; this scenario, where unlearning is required without access to the original or similar training data, is termed *data-free unlearning*.

An existing method addressing data-free unlearning is Generative Knowledge Transfer (GKT) (Chundawat et al. 2023), which utilizes Data-free Knowledge Distillation (Micaelli and Storkey 2019) to selectively transfer knowledge from a trained model to an unlearned model. Both data-free unlearning and data-free knowledge distillation (DFKD) train a generator with the principle of synthesizing new samples that simulate the original training distribution, which can be achieved by harnessing the capabilities of the well-trained model, and followed by the knowledge distillation (Chen et al. 2019; Choi et al. 2020; Do et al. 2022). With meaningful synthetic samples in hand, GKT selectively distil only the necessary retaining knowledge by filtering out synthetic samples potentially belonging to the forgetting class. Specifically, it excludes samples where the probability associated with forgetting classes as outputted by the logits exceeds a very low threshold before distillation. This filtering strategy can ensure that samples involved in distillation contain minimal forgetting class information and effectively filter out forgetting knowledge.

Despite selective distillation being addressed, the training principle of DFKD's generator, which with the aim of synthesizing "not yet seen" data for student networks, can present a novel challenge in unlearning scenarios. Intuitively, in the context of unlearning, forgetting class samples will inevitably be the "not yet seen" data for the student network, resulting the generator increasingly producing a significant number of samples from the forgetting class. As shown in Figure 1, when the filtering strategy is employed, the generator can synthesize an increasing number of forgetting class samples. Consequently, the large volume of filtered-out potential forgetting class samples significantly reduces the pool of data available for distillation, leading to inefficiencies. Furthermore, this filtering impedes the distillation process by inadvertently excluding logits outputs from other classes, which are crucial for knowledge transfer and represent a key advantage of knowledge distillation that relies on these outputs instead of hard labels.

*Corresponding Author.



(a) Without filter (b) With filter

Figure 1: Comparative Visualization of Synthetic Samples during the DFKD process on the SVHN Dataset: **Background shadow** illustrates the real data distribution; **Triangles:** Synthetic samples; **Red (R):** Samples of the forgetting class (digit "7"); **Yellow (Y):** Non-forgetting class samples but still filtered out by the filter of GKT; **Blue (B):** Samples deemed suitable for participation in the distillation process. **(a)** When there is no filter, i.e., the distillation of complete knowledge, the synthesized data has minimal bias toward the forgetting class. **(b)** When a filter is used, i.e. when performing unlearning, a biased high volume of the forgetting class sample is synthesized.

In this work, we first strictly analyze the efficiency problem when using Data-free Knowledge Distillation (DFKD) to perform selective distillation and achieve data-free unlearning. We find that enriching the information about retaining classes involved in the distillation process can effectively improve student's learning of retaining-related knowledge. We propose the Inhibited Synthetic PostFilter (ISPF) method to achieve this goal from two perspectives. The first is to reduce the synthetic potential forgetting class sample, which is named Inhibited Synthesis (IS), by designing a new objective function for training the generator. This objective can effectively suppress the generator's exploration of the forgetting classes' distribution. The second is to involve all synthetic samples in the distillation process and filter out forgetting class knowledge by modifying the teacher's output, which is named PostFilter (PF). This allows the distillation process to leverage as much information as possible about the retaining classes in a synthesis batch. We summarize our contribution as follows:

- We identify and analyze the challenge associated with applying DFKD in unlearning. Our findings suggest that by enriching the information related to retaining classes in the distillation process, we can significantly improve the student model's acquisition of knowledge pertaining to these classes.
- We propose two key technologies for tackling this challenge: 1) Inhibited Synthesis for reducing the synthesis of information about the forgetting class and 2) PostFilter for leveraging as much information as possible about the retaining classes.
- Experimental results show that our proposed method can tackle the challenge and outperform existing methods.

Due to space constraints, the related works section is provided in Appendix A of the extended version.

2 Preliminaries

Given a dataset $\mathcal{D} = \{(x_i, y_i)|x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_i^N$, where the data instance $x \in \mathbb{R}^d$ is sampled from a real distribution $P(x)$, i.e., $x \sim P(x)$. A neural network $h(\theta, x)$ which is parameterized by θ can output a probability vector that indicates the probability of the given x being classified as each class, i.e., $h(\theta, x) \in (0, 1)^K$, where K is the number of classes. When the network is required to unlearn classes $\mathcal{Y}_f$, the classes that need to be retained are in $\mathcal{Y}_r$, where $\mathcal{Y}_f \cup \mathcal{Y}_r = \mathcal{Y}$ and $\mathcal{Y}_f \cap \mathcal{Y}_r = \emptyset$. We can further define the corresponding subsets $\mathcal{D}_f$ and $\mathcal{D}_r$, where $\mathcal{D}_f = \{(x, y)|y \in \mathcal{Y}_f\}$ and $\mathcal{D}_r = \{(x, y)|y \in \mathcal{Y}_r\}$. Most existing unlearning methods, which we denote as $\mathcal{U}$, require data samples from the real distribution, and their goal can be formed as $\theta_{un} = \mathcal{U}(\theta_{or}, x)$, where θ_{or} is the parameter of the original model.

Data-free Unlearinng A data-free unlearning method is required to take only the trained network $h(\theta_{or}, \cdot)$ without sample $x \sim P(x)$, i.e., $\theta_{un} = \mathcal{U}(\theta_{or})$.

Data-free Knowledge Distillation (DFKD) Knowledge Distillation (KD) can transfer knowledge from a well-trained teacher network $h(\theta_T, \cdot)$ to a randomly initialized student network $h(\theta_S, \cdot)$. For representation convenience, we denote them as $T(\cdot)$ and $S(\cdot)$, respectively. The general goal of KD is to minimize the gap between the outputs of these two networks by using the Kullback–Leibler divergence (Kullback and Leibler 1951) to measure the discrepancy. In DFKD, the input samples $\widetilde{x}$ are synthesized by a generator, thus the student's loss function is

$$\mathcal{L}_S(\widetilde{x}) = D_{KL}(T(\widetilde{x}) \parallel S(\widetilde{x})), \tag{1}$$

where $\widetilde{x} = G(z)$ and the $G(z)$ is a generator that can project a given lower-dimensional noise $z \in N(0, I)^{d_z}$ to $\widetilde{x} \in \mathbb{R}^d$. DFKD methods usually train the generator adversarially by maximizing the output distribution discrepancy between teacher and student, i.e.,

$$\mathcal{L}_{adv}(\widetilde{x}) = -D_{KL}(T(\widetilde{x}) \parallel S(\widetilde{x})). \tag{2}$$

PreFilter In addition to using the same objectives as Eq. 2 to train the generator, the existing data-free unlearning method, GKT (Chundawat et al. 2023), applies a filter ahead of the generator which receives all the synthetic samples and filters them out before passing them to the student and we refer to this as PreFilter. Specifically, the synthetic samples that can be involved in the distillation satisfy

$$\forall \widetilde{x}_i, \forall k \in \mathcal{Y}_f, T_k(\widetilde{x}_i) < \delta, \tag{3}$$

where k is the class label, $T_k(\cdot)$ is the k-th value of the output probability vector and δ is a hyperparameter whose exact value for ten-class datasets in the GKT is 0.01.

3 Method

This section is structured as follows: Section 3.1 analyzes the challenge encountered in unlearning scenarios using

DFKD directly. Section 3.2 and Section 3.3 introduce the two key techniques, i.e., Inhibited Synthesis and PostFilter, respectively. The overall algorithm is placed in Appendix B.

3.1 Challenge in Unlearning

We first expand the form of Eq. 2 into the following form

$$\mathcal{L}_{adv}(\widetilde{x}_i) = -\sum_k T_k(\widetilde{x}_i) \cdot [\log(T_k(\widetilde{x}_i)) - \log(S_k(\widetilde{x}_i))], \tag{4}$$

where $\widetilde{x}_i = G(z_i)$. Note that this is an objective function for updating G, thus, the T and S are fixed when computing $\mathcal{L}_{adv}$. The goal of minimizing the Eq. 4 is equal to maximizing the $-\mathcal{L}_{adv}$

$$\begin{aligned} &\min_G \mathbb{E}_{\widetilde{x}=G(z)} [\mathcal{L}_{adv}(\widetilde{x})] \\ =&\max_G \mathbb{E}_{\widetilde{x}=G(z)} [-\mathcal{L}_{adv}(\widetilde{x})] \\ =&\max_G \mathbb{E}_{\widetilde{x}=G(z)} \left[\sum_k T_k(\widetilde{x}) \cdot [\log(T_k(\widetilde{x})) - \log(S_k(\widetilde{x}))]\right] \end{aligned} \tag{5}$$

The expected result for the student is that there is no high enough output probability of forgetting class for any data sample. Therefore, for any forgetting class f, the f-th element of the student's output is very low

$$\forall \widetilde{x} = G(z),\ 0 < S_f(\widetilde{x}) < \epsilon. \tag{6}$$

To reach the goal of maximizing Eq. 5 when $S_f(\widetilde{x})$ is small it is necessary to increase $T_f(\widetilde{x})$ to a large value. This leads to an increasing probability that synthetic samples will determined by the teacher as forgetting class. As the training processes, an increasing number of synthetic data $\widetilde{x}$ will be filtered out due to the $T_f(\widetilde{x})$ exceeding the specified threshold δ of the PreFilter, resulting in a decreasing number of samples being used for distillation. This ultimately leads to a reduction in distillation efficiency.

3.2 Inhibited Synthesis (IS)

As analyzed above, the student's lack of familiarity with the synthetic samples, which contain information regarding the forgetting class, encourages G to explore further into samples that contain a greater quantity of information regarding the forgetting class. This results in a greater number of synthetic samples being filtered out by PreFilter before distillation. Therefore, we want the generator to synthesize fewer samples of the forgetting class and more samples of the retaining class. To address this issue, we need to encourage the generator to explore samples that are not previously encountered by the student, while simultaneously suppressing the generator's exploration of the underlying real distribution of the forgetting class.

To minimize the number of samples generated by G that contain information about the forgetting class, it is necessary to reduce the value of $T_f(\widetilde{x})$. We can achieve this by reducing the gap between T_f and S_f given that $S_f(\widetilde{x})$ is always low for each synthetic sample. Therefore, we propose the inhibited synthesis loss for the generator's learning, i.e.,

$$\begin{aligned} \mathcal{L}_{IS}(\widetilde{x}_i) = &-\sum_{k\in\mathcal{Y}_r} T_k(\widetilde{x}_i) \cdot [\log(T_k(\widetilde{x}_i)) - \log(S_k(\widetilde{x}_i))] \\ &+ \sum_{f\in\mathcal{Y}_f} T_f(\widetilde{x}_i) \cdot [\log(T_f(\widetilde{x}_i)) - \log(S_f(\widetilde{x}_i))]. \end{aligned} \tag{7}$$

Given that samples with $T_f(\widetilde{x})$ exceeding a very small threshold ϵ are excluded from distillation, leaving only $\widetilde{x}$'s with $0 < T_f(\widetilde{x}) < \epsilon$, therefore, the $S_f(\widetilde{x})$ of a student engaged in learning with the objective of Eq. 1 will invariably be smaller than ϵ. Consequently, we can use the Eq. 7 to actively diminish the $T_f(\widetilde{x})$, thereby impeding the generator from producing samples with elevated $T_f(\widetilde{x})$.

3.3 PostFilter (PF)

In the previous discussion, we endeavored to diminish the number of samples excluded from distillation by impeding G's synthesis of samples bearing information about the forgetting class, thereby augmenting the number of samples utilized for distillation. The experiments revealed that even when the inhibited G has synthesized almost no samples that can be classified as the forgetting class by the teacher, some samples are still filtered out (third column in Figure 2). This is because the synthetic samples lack sufficient purity to represent an individual class. Synthetic samples belonging to the retaining class may contain information about the forgetting class, and similarly, synthetic samples belonging to the forgetting class may also contain information about retaining classes. In light of this observation, we wished to enrich the material employed in the distillation process by fully using the synthetic samples.

Specifically, we construct new supervision information for the student by redistributing the logits of the teacher output. This is achieved by setting the forgetting classes' value in the teacher's output logits to the lowest value and distributing the sum of the subtracted logits evenly to the logits of the retaining classes. We denote the logits output (network's output before Softmax) of each sample from the teacher as $\boldsymbol{t}$. We first calculate the total logits value Δ that needs to be redistributed by summing the difference between forgetting class logits and the minimal value in $\boldsymbol{t}$, i.e.,

$$\Delta = \sum_{k\in\mathcal{Y}_f} [\boldsymbol{t}_k - min(\boldsymbol{t})] \tag{8}$$

Then, we construct the supervision information $\hat{\boldsymbol{t}}$ by redistributing the Δ to all retaining classes and setting the forgetting class logits value as the minimum, i.e.,

$$\hat{\boldsymbol{t}}_k = \begin{cases} \boldsymbol{t}_k + \frac{\Delta}{K-|\mathcal{Y}_f|}, & \text{if } k \notin \mathcal{Y}_f, \\ min(\boldsymbol{t}), & \text{otherwise.} \end{cases} \tag{9}$$

The student will then use the redistributed teacher logits as supervision information to calculate losses

$$\mathcal{L}_S(\widetilde{x}) = D_{KL}(Softmax(\hat{\boldsymbol{t}}) \parallel S(\widetilde{x})). \tag{10}$$

A simpler implementation can be setting the value of $T(\widetilde{x})$ corresponding to the forgetting class as 0 and renormalizing other values to obtain the distillation target. We experimentally compared this simple implementation with our proposed PF (Appendix I), and the results show that PF outperforms this simple implementation in terms of both unlearned model's performance and unlearning guarantee.

4 Experiment

We evaluate the effectiveness of the ISPF which is composed of two proposed techniques, the Inhibited Synthesis (IS) and PostFilter (PF), on three widely used benchmark datasets, i.e., SVHN (Netzer et al. 2011), CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009). Two neural network architectures, i.e., AllCNN (Springenberg et al. 2015) and ResNet18 (He et al. 2016), are used in our experiments. For SVHN and CIFAR-10 datasets, we apply both network architectures. For CIFAR-100, we apply only ResNet18. Implementation details are in the Appendix C.

4.1 Experimental Setup

Datasets We perform unlearning on each class from both SVHN and CIFAR-10. CIFAR-100 has 100 classes which can be divided into 20 super-classes. We perform single-class unlearning on three randomly selected classes of the CIFAR-100, i.e., $y_f \in \{18, 33, 79\}$. We also use the CIFAR-100 for multi-classes unlearning experiments where more than one class needs to be unlearned. We use all 100 classes to train the original model for generality and choose one from each of 10 different super-classes to make up the 10 classes to unlearn. Specifically, we select class labels $\mathcal{Y}_f = \{0, 1, 2, 3, 4, 5, 6, 8, 9, 12\}$, and a more detailed explanation about why selecting these classes is presented in the section of **CIFAR-100 Results**.

Baselines We refer to the model retrained from scratch as the gold result and compare the proposed methods with the existing data-free unlearning method that requires no access to data from real distribution, the **GKT** (Chundawat et al. 2023). The naive method of blocking synthetic samples that can be determined as forgetting class by the original model has also been included in baselines as **BlockF**.

Fundamental DFKD Method We select the DFQ (Choi et al. 2020) as the main fundamental DFKD method because it is a representative DFKD method that obeys the adversarial inversion-and-distillation paradigm and has been widely used as a baseline in other DFKD works. In the following experimental sections, DFKD refers to DFQ unless otherwise stated. We also conducted experiments using ZSKT (Micaelli and Storkey 2019), which was originally used by the GKT, as the fundamental DFKD (Appendix H).

Evaluation Metrics To evaluate the effectiveness of our proposed methods, we refer to existing works (Chundawat et al. 2023; Chen et al. 2023; Tarun et al. 2023; Chen et al. 2024; Fan et al. 2023) and select the following four metrics.

Classification Accuracies: We test unlearned models on the forgetting test data $\mathcal{D}_f^{test}$ and the retaining test data $\mathcal{D}_r^{test}$ for obtaining accuracies A_f and A_r, respectively. Both A_f and A_r are the closer to that of the *Retrain* model the better.

Anamnesis Index (AIN): As introduced by Chundawat et al. (2023), the AIN evaluates how much forgetting information remains in an unlearned model by measuring the amount of time (training steps) it takes to relearn a comparable A_f as the original model. This is a ratio of an unlearned model's relearning time to a retrained model's relearning time. Therefore, the closer the ratio is to 1, the better.

Membership Inference Attack (MIA): We implement two types of MIA attack. 1) Following (Fan et al. 2023; Chen et al. 2024), we define the first MIA metric MIA$_I$ as the rate of unlearning samples that are identified as not being in the training set of the unlearned model. A higher MIA$_I$ indicates a more effective unlearning. 2) We also refer to experiments in Boundary Unlearning (Chen et al. 2023) and implement a simple general MIA based on (Shokri et al. 2017). In this approach, multiple shadow models are trained to gather signals for the attacker's training. The attacker aims to determine whether a given signal originates from a sample within the in-training dataset. During the attacker's testing, a test set that includes both in-training and out-of-training data is used and we define the MIA$_{II}$ as the F1 score when it attacks each unlearned model. Generally, the attacker can achieve the best performance on the signals obtained from the original model, however, a lower MIA$_{II}$ doesn't mean it's a better unlearning guarantee, as a randomly initialized model provides signals that can confuse the attacker. Therefore MIA$_{II}$ should be close to the F1 when attacking the retrained model.

4.2 Comparison With Baselines

Unlearned Model Performance Table 1 reports unlearning performance averaged across all classes, and we also provide detailed results when each class is set as the forgetting class in Appendix D. Table 1 illustrates that the unlearned model obtained by BlockF still exhibits generalization ability to the forgetting class. This suggests that solely blocking samples, which are identified as the forgetting class by the original model, from participating in the distillation process can not effectively way block the forgetting class information. This is because the synthetic samples cannot represent an individual class in a sufficiently distinct manner, and the student can still learn the forgetting class through the samples that are determined as other classes. The GKT uses PreFilter to filter samples based on the teacher's confidence in the forgetting class. This approach ensures that only samples with minimal confidence in the forgetting class are included in the distillation process, effectively filtering out knowledge related to the forgetting class in comparison to BlockF. This can be seen by comparing the A_f performance of GKT and BlockF. However, the use of the PreFilter will cause the increase in synthesizing forgetting class samples and also the exclusion of samples from other classes (third column in Figure 2), which in turn leads to a reduction in the knowledge related to the retaining class participating in the distillation and affects the learning efficiency of retaining knowledge. For example, the A_r of GKT is significantly inferior to that of DFKD which strives to facilitate the com-

Arch	Method	SVHN					CIFAR-10				
		A_f	A_r	MIA$_I$	MIA$_{II}$	AIN	A_f	A_r	MIA$_I$	MIA$_{II}$	AIN
AllCNN	Original	93.86±0.27	93.86±0.27	4.96	39.67	-	90.91±0.13	90.91±0.13	2.99	45.86	-
	DFKD	91.37±0.23	91.37±0.23	0.0	39.56	0.07	84.85±0.15	84.85±0.15	1.2	44.91	0.11
	Retrain	0.0±0.0	94.14±0.13	100.0	33.28	1.0	0.0±0.0	91.39±0.04	100.0	37.34	1.0
	BlockF	90.33±0.71	92.35±0.07	1.91	38.75	0.06	65.67±1.45	85.08±0.5	10.4	44.65	0.10
	GKT	0.0±0.0	55.23±2.42	75.0	19.86	1.30	0.0±0.0	57.62±1.13	47.42	21.51	1.44
	IS (ours)	0.0±0.0	90.88±0.21	**100.0**	**33.51**	**0.99**	0.0±0.0	84.93±0.02	**100.0**	**37.35**	0.86
	PF (ours)	0.0±0.0	91.15±0.85	**100.0**	34.24	0.91	0.0±0.0	76.76±0.06	**100.0**	26.36	1.28
	ISPF (ours)	0.0±0.0	**92.68±0.13**	**100.0**	34.54	0.96	0.0±0.0	**86.02±0.03**	**100.0**	36.14	**0.89**
ResNet18	Original	94.34±0.11	94.34±0.11	2.55	44.59	-	92.02±0.51	92.02±0.51	2.94	42.17	-
	DFKD	91.72±0.38	91.72±0.38	0.83	38.34	0.17	83.25±0.59	83.25±0.59	0.0	39.67	0.04
	Retrain	0.0±0.0	94.43±0.03	100.0	36.51	1.0	0.0±0.0	92.4±0.23	100.0	30.81	1.0
	BlockF	74.97±1.55	91.87±0.03	51.35	36.51	0.12	48.5±0.84	81.49±0.44	6.47	36.21	0.04
	GKT	0.0±0.0	88.75±1.43	**100.0**	29.07	0.48	0.0±0.0	58.11±0.76	45.0	16.25	**0.51**
	IS (ours)	0.0±0.0	90.75±0.55	**100.0**	**34.27**	0.46	0.0±0.0	80.54±0.25	**100.0**	28.11	0.22
	PF (ours)	0.0±0.0	91.61±0.41	**100.0**	32.28	**0.52**	0.0±0.0	81.21±0.72	**100.0**	22.73	0.45
	ISPF (ours)	0.0±0.0	**91.92±0.23**	**100.0**	33.67	0.45	0.0±0.0	**83.33±0.81**	**100.0**	**29.36**	0.28

Table 1: Unlearning performance averaged across all classes. The AIN is reported as a ratio and all other metrics are reported as percentages (%). The **bold** record indicates the best result and the underlined record indicates the second best result.

plete transfer of knowledge from the teacher to the student.

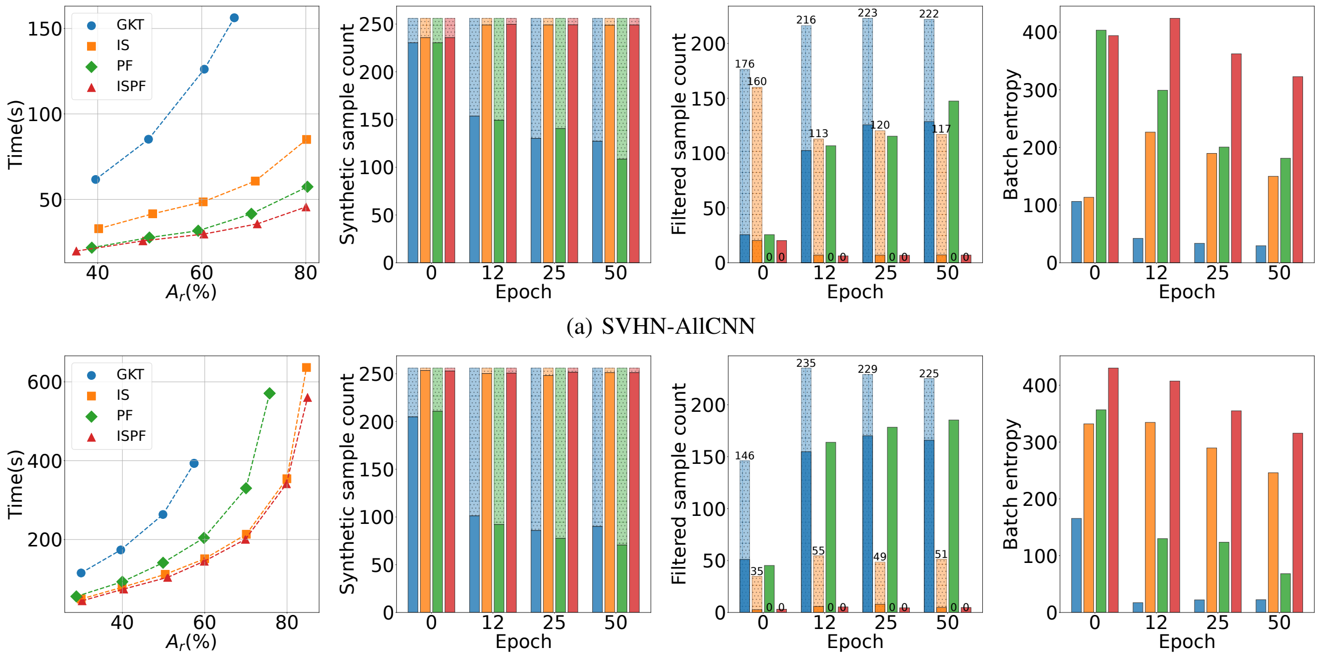
In contrast, our proposed ISPF outperformances baselines on both A_r and A_f. This is because ISPF effectively inhibits the synthesis of forgetting class samples through IS, while using PF to fully utilize the retaining-related knowledge in the synthetic samples. We will further explain why the IS and PF work in the **Ablation** section.

Unlearning Guarantee We note that in some settings, a single metric may not effectively differentiate the discrepancy in unlearning guarantee between methods. We therefore include three widely used metrics for evaluating the unlearning guarantee of unlearned models. Our proposed methods consistently outperform the comparative method on two or even three metrics across diverse settings. In Appendix G, we present additional perspectives, including the model's predictive distribution and representation, to demonstrate that our proposed method exhibits performances more closely aligned with *Retrain*.

1) MIA$_I$: According to (Fan et al. 2023; Chen et al. 2024), the MIA$_I$ metric is defined as the rate of unlearning samples that are identified as not being in the training set of the unlearned model. As shown in Table 1, the MIA$_I$ of the retrain model is 100%, indicating that the attacker believes that none of the forgetting samples are included in the retrain model's training data. This result can also be observed in the Figure 2 in (Chen et al. 2024). Conversely, models with all classes' knowledge exhibit comparable and relatively low performance on MIA$_I$. For instance, the MIA$_I$ of the original model and the model obtained through DFKD, both fall below 5%. In comparison methods, the BlockF exhibits an A_f that is not low, and MIA$_I$ is comparable to the original model and the one obtained by complete distillation. GKT's MIA$_I$ is considerably higher than BlockF's but remains significantly lower than Retrain's. Our proposed ISPF's MIA$_I$ is consistent with Retrain's, indicating that ISPF has a superior unlearning guarantee in terms of MIA$_I$.

2) MIA$_{II}$: From Table 1, models with comprehensive knowledge of all classes exhibit the highest MIA$_{II}$. For instance, both the Original and DFKD demonstrate MIA$_{II}$ values of approximately 40% or greater, which are markedly higher than those observed in the Retrain. In the comparison methods, BlockF's MIA$_{II}$ is typically higher than Retrain's and nearly equivalent to Original's, indicating that there is still a considerable amount of knowledge related to forgetting data in the unlearned model obtained by BlockF. GKT has the lowest MIA$_{II}$, which is because GKT is unable to effectively maintain the knowledge of retaining classes, resulting in the unlearned model's overall performance being low, and more akin to a model that has not been adequately trained. As previously stated, the signal generated by an inadequately trained model can also confuse the attacker, resulting in a low success rate of the attack. Therefore, the MIA$_{II}$ metric should be as close to Retrain as possible. The results demonstrate that our proposed method's MIA$_{II}$ is the closest to Retrain.

3) AIN: Models with complete knowledge have AIN scores close to or even less than 0.1, as observed in those of DFKD and BlockF. This means that the models obtained by these two methods require only a few training steps to restore A_f performance to a level comparable to that of the original model and also indicates that these models still contain forgetting related knowledge. When the AllCNN network is employed, our proposed methods yield AIN scores of approximately 1, indicating that the unlearned models obtained through our proposed methods contain an equivalent amount of forgetting-related knowledge as the retrained model, which is essentially negligible. While the AIN value of our proposed method is smaller when using the ResNet18, it remains considerably higher than that of



(a) SVHN-AllCNN

(b) CIFAR10-AllCNN

Figure 2: The colors in all figures are used to distinguish the different methods. The top row shows the results under the SVHN-AllCNN setting, and the bottom row shows the results under CIFAR10-AllCNN. The **first column** shows the results for A_r vs. wall time. In the **second column**, the light bar filled with dots shows the number of synthetic samples classified as forgetting classes by the original model, and the dark bar shows the number of retaining class samples. In the **third column**, the light bar filled with dots shows the number of samples filtered out before distillation and the number indicates the exact number of filtered-out samples, and the darker bar shows the number of synthetic forgetting class samples, which is the same as the lighter bar in the second column. Corresponding results on ResNet18 are in the Appendix E.

DFKD and BlockF, still indicating that the unlearned models acquired through ISPF incur a substantially higher cost to restore A_f to a comparable level to original model.

Efficiency To demonstrate the efficiency of ISPF, we report the time taken by each method to achieve a similar A_r in the first column of Figure 2. Since PostFilter and PreFilter process different numbers of samples during distillation, the time required for each training step varies. Therefore, we use wall time (actual elapsed time) for a more fair comparison of their efficiency. The results show that ISPF requires significantly less training time than GKT to reach the same A_r, demonstrating substantial efficiency gains for both PostFilter and PreFilter techniques within ISPF.

4.3 Ablation

In this subsection, we conduct ablation experiments on two key techniques in the ISPF to elucidate their contributions respectively. Specifically, when examining the IS, we utilize the PreFilter to filter forgetting-related knowledge; when examining the PF, we just exclude the IS from ISPF.

How does the Inhibited Synthesis (IS) work? As previously discussed in the Section 3.1, in the unlearning scenario, in the absence of constraints during the generator's training process, the generator will progressively synthesize more samples that contain information about forgetting classes. As illustrated in the second column in Figure 2, in methods that do not incorporate the IS, such as GKT and PF, an increasing number of samples comprising information about forgetting classes are synthesized as training progresses, whereas a decreasing number of samples contain information about retaining classes. Furthermore, the third column of Figure 2 demonstrates that the number of samples filtered out by the PreFilter is considerably larger than the number of samples belonging to the forgetting classes. This suggests that a significant proportion of samples from the retaining classes are also filtered out due to their resemblance to the forgetting classes, which further reduces the number of samples involved in distillation and reduces the student's learning efficiency. By comparing the results of GKT and IS, IS not only markedly reduces the synthesis of samples for forgetting classes, but also significantly reduces the number of samples that are filtered out by the PreFilter. This can also be seen from the representation visualization in Figure 3. This suggests that IS is an effective method of suppressing the synthesis of samples that contain forgetting class information, thereby markedly enhancing student's learning efficiency in retaining knowledge.

Method	Single-Class		Multi-Classes	
	A_f	A_r	A_f	A_r
Original	70.3±0.3	70.31±0.33	70.3±0.3	70.31±0.33
DFKD	61.8±0.1	61.81±0.16	61.8±0.1	61.8±0.16
Retrain	0.0±0.0	70.36±0.11	0.0±0.0	70.13±0.04
GKT	0.0±0.0	59.65±0.37	0.0±0.0	49.86±3.01
IS (ours)	0.0±0.0	60.72±0.36	0.0±0.0	58.32±0.51
PF (ours)	0.0±0.0	61.67±0.36	0.0±0.0	57.07±2.19
ISPF (ours)	0.0±0.0	**62.14±0.25**	0.0±0.0	**62.58±0.62**

Table 2: CIFAR-100 results.

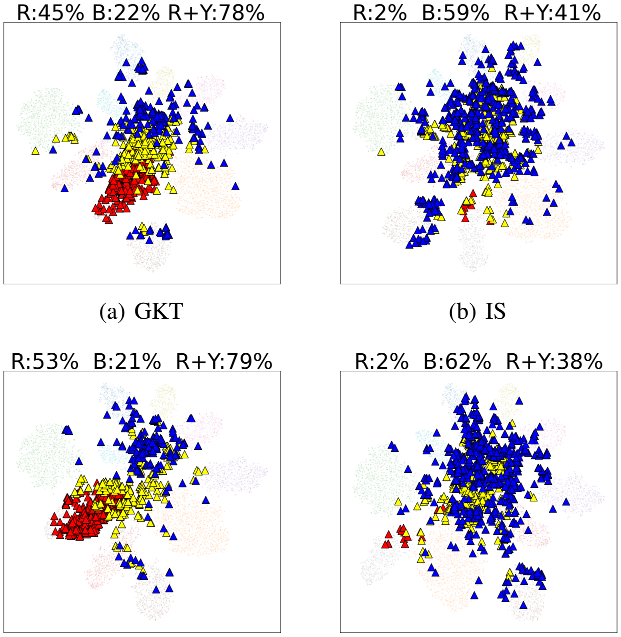
How does the PostFilter (PF) work? A comparison of the results of GKT and PF reveals that, as neither employs IS to restrict the learning of the generator, both appear to synthesize an increasing number of forgetting class samples as the training progresses. However, in contrast to GKT, the PF does not remove any sample before distillation. Instead, it utilizes as much information as possible about the retaining classes in all samples. To this end, we quantify the information entropy of the retaining classes in the output of the teacher, in each training step. Specifically, we calculated the information entropy in each training step as

$$H_B = -\sum_{i}^{N} \sum_{k \notin \mathcal{Y}_f} T_k(\widetilde{x}_i) \cdot \log(T_k(\widetilde{x}_i)), \tag{11}$$

where the N is the number of samples that are involved in distillation. We report the average H_B across all training steps in the fourth column in Figure 2. As shown in the figure, PF can provide a considerably greater quantity of information to the student in each training step when compared to the GKT. This suggests that, although the PF also encounters the challenge of an increasing number of synthetic forgetting class samples, its utilization of the retaining information within the samples to its fullest potential also markedly enhances the learning efficiency of the student.

4.4 Additional Analysis

Visualization We visualize the representation of synthetic samples during the experiments on SVHN, using the unlearning of "7" as an example, and the results are plotted in Figure 3. The symbols in Figure 3 are consistent with those in Figure 1. Compared to GKT, our IS, while still using PreFilter, significantly suppresses the synthesis of the forgetting class samples and increases the number of samples involved in the distillation. While the number of synthesized forgetting class samples is not reduced in PF, it makes full use of retaining-related information from all synthetic samples. When PF is combined with IS, which is the ISPF, the number of synthesized samples containing forgetting information is further reduced and all samples are fully utilized. We also visualize the synthetic images in Appendix J.



(a) GKT

(b) IS

(c) PF

(d) ISPF

Figure 3: Visualization on SVHN.

CIFAR-100 Results As shown in Table 2, ISPF still outperforms the existing method, particularly in the setting of multi-classes unlearning. For multi-classes unlearning, we select class labels $\mathcal{Y}_f = \{0, 1, 2, 3, 4, 5, 6, 8, 9, 12\}$. Note that this is a more difficult setup. If the chosen forgetting classes are sharing one superclass, the task can be reduced to forgetting one superclass and fewer retaining classes will be filtered out by the PreFilter. However, if the selected forgetting classes are spread across superclasses, samples from other retaining classes in each superclass will be filtered out due to their similarity to the forgetting classes, resulting in more samples being filtered out in a batch. We also chose 10 classes from two superclasses and the results (Appendix F) show that the number of samples involved in the distillation in GKT is significantly more than that when 10 classes are distributed over 10 superclasses, and the final result of $A_r = 56.04$ (vs. $A_r = 49.86$ in Table 2) is also better. This result demonstrates that the ISPF exhibits superior performance in a more challenging setting, even when compared to the existing method that operates under an easier setting.

5 Conclusion

We theoretically analyzed and experimentally demonstrated the inefficiency in retaining knowledge during data-free unlearning when using Data-free Knowledge Distillation (DFKD). Our findings show that enriching the information related to retaining classes during distillation significantly enhances the student model's learning of retaining-related knowledge. We propose the Inhibited Synthetic PostFilter (ISPF) to achieve this from two perspectives: 1) reducing the synthesis of forgetting class information and 2) fully leveraging the retaining-related information in the synthesized samples. Experimental results confirm that ISPF effectively overcomes this challenge and outperforms existing methods.

Acknowledgements

The work is supported by the Australian Research Council Discovery Projects DE230101116 and DP240103070.

References

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine Unlearning. In *SP 2021*.

BUKATY, P. 2019. *The California Consumer Privacy Act (CCPA): An implementation guide*. IT Governance Publishing. ISBN 9781787781320.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *SP 2015*.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-Free Learning of Student Networks. In *ICCV 2019*.

Chen, H.; Zhu, T.; Yu, X.; and Zhou, W. 2024. Machine Unlearning via Null Space Calibration. In *IJCAI 2024*.

Chen, M.; Gao, W.; Liu, G.; Peng, K.; and Wang, C. 2023. Boundary Unlearning: Rapid Forgetting of Deep Networks via Shifting the Decision Boundary. In *CVPR 2023*.

Choi, Y.; Choi, J. P.; El-Khamy, M.; and Lee, J. 2020. Data-Free Network Quantization With Adversarial Knowledge Distillation. In *CVPR 2020*.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. S. 2023. Zero-Shot Machine Unlearning. *IEEE Trans. Inf. Forensics Secur.*, 18: 2345–2354.

Do, K.; Le, H.; Nguyen, D.; Nguyen, D.; Harikumar, H.; Tran, T.; Rana, S.; and Venkatesh, S. 2022. Momentum Adversarial Distillation: Handling Large Distribution Shifts in Data-Free Knowledge Distillation. In *NeurIPS 2022*.

Fan, C.; Liu, J.; Zhang, Y.; Wei, D.; Wong, E.; and Liu, S. 2023. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. *CoRR*.

Foster, J.; Schoepf, S.; and Brintrup, A. 2024. Fast Machine Unlearning without Retraining through Selective Synaptic Dampening. In *AAAI 2024*.

Garg, S.; Goldwasser, S.; and Vasudevan, P. N. 2020. Formalizing Data Deletion in the Context of the Right to Be Forgotten. In Canteaut, A.; and Ishai, Y., eds., *EUROCRYPT 2020*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR 2016*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22: 79–86.

Kurmanji, M.; Triantafillou, P.; Hayes, J.; and Triantafillou, E. 2023. Towards Unbounded Machine Unlearning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *NeurIPS 2023*.

Marchant, N. G.; Rubinstein, B. I. P.; and Alfeld, S. 2022. Hard to Forget: Poisoning Attacks on Certified Machine Unlearning. In *AAAI 2022*.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot Knowledge Transfer via Adversarial Belief Matching. In *NeurIPS 2019*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*.

Nguyen, T. T.; Huynh, T. T.; Nguyen, P. L.; Liew, A. W.; Yin, H.; and Nguyen, Q. V. H. 2022. A Survey of Machine Unlearning. *CoRR*, abs/2209.02299.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks Against Machine Learning Models. In *SP 2017*.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. A. 2015. Striving for Simplicity: The All Convolutional Net. In Bengio, Y.; and LeCun, Y., eds., *ICLR 2015*.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2023. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*.

Thudi, A.; Deza, G.; Chandrasekaran, V.; and Papernot, N. 2022. Unrolling SGD: Understanding Factors Influencing Machine Unlearning. In *EuroS&P 2022*.

Xu, J.; Wu, Z.; Wang, C.; and Jia, X. 2024. Machine Unlearning: Solutions and Challenges. *IEEE Trans. Emerg. Top. Comput. Intell.*